

# 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>12</b>
1.1 Aim of the package	12
1.2 History	13
1.2.1 edmac	13
1.2.2 ledmac	14
1.2.3 eledmac	15

---

\*This file (reledmac.dtx) has version number v2.34.6, last revised 2020/10/10.

<sup>†</sup>maieul at maieul dot net

1.2.4 <code>reledmac</code> . . . . .	15
1.3 Bibliography . . . . .	15
1.4 Acknowledgements . . . . .	15
<b>2 How the package works — the problem of the number of <math>\LaTeX</math> runs</b>	<b>15</b>
<b>3 Compatibility warning</b>	<b>16</b>
<b>4 Options</b>	<b>16</b>
4.1 Specific features . . . . .	16
4.2 Optimizing package performance . . . . .	17
<b>5 Text lines and paragraphs numbering</b>	<b>18</b>
5.1 Text lines numbering . . . . .	18
5.2 Paragraphs . . . . .	18
5.2.1 Basics . . . . .	18
5.2.2 Automatically producing <code>\pstart ... \pend</code> . . . . .	19
5.2.3 Content before specific <code>\pstart</code> and after specific <code>\pend</code> . . . . .	19
5.2.4 Content before every <code>\pstart</code> and after every <code>\pend</code> . . . . .	20
5.2.5 Numbering paragraphs ( <code>\pstart</code> ) . . . . .	20
5.2.6 Languages written in Right to Left . . . . .	21
5.2.7 Memory limits . . . . .	21
5.3 Lineation commands . . . . .	21
5.3.1 Disabling lineation . . . . .	21
5.3.2 Setting lineation start and step . . . . .	22
5.3.3 Setting lineation reset . . . . .	22
5.3.4 Setting line number margin . . . . .	23
5.3.5 Other settings . . . . .	23
5.4 Changing the line numbers . . . . .	23
5.4.1 Sublineation . . . . .	23
5.4.2 Locking lineation . . . . .	24
5.4.3 Setting and changing line number . . . . .	24
5.4.4 Line number style . . . . .	24
5.4.5 Skipping and hiding number . . . . .	25
5.5 Adding annotations to line numbers . . . . .	25
5.5.1 Modifying annotation associated with a specific note . . . . .	26
5.5.2 Changing the position of the annotation . . . . .	26
5.5.3 Changing the macro in which annotations are wrapped . . . . .	27
5.5.4 Not printing the second identical annotations . . . . .	28
5.5.5 Separator between annotations . . . . .	28
5.5.6 Annotations in the apparatus . . . . .	28
5.6 Executing code at each line . . . . .	28
5.7 Executing code at a specific line . . . . .	28

<b>6 Apparatus commands</b>	<b>29</b>
6.1 Terminology	29
6.2 Critical notes	29
6.2.1 The lemma	29
6.2.2 Footnotes	30
6.2.3 Endnotes	30
6.2.4 Paragraph in critical apparatus	32
6.2.5 Change lemma and line number	32
6.2.6 Changing the names of commands for critical apparatus	33
6.3 Disambiguation of identical words in the apparatus	33
6.3.1 Basic use	33
6.3.2 Case setting	34
6.3.3 Notes about input encoding with UTF-8 processor	34
6.3.4 Notes about right-to-left texts	34
6.3.5 Use with <code>\lemma</code> command	35
6.3.6 Sameword for a group of words	36
6.3.7 Customizing	37
6.3.8 Problems with some macros	37
6.3.9 Automatic sameword annotation	38
6.4 Apparatus of manuscripts	38
6.4.1 Marking sections of text	39
6.4.2 Layout of the apparatus of manuscripts	39
6.4.3 Settings	40
6.5 Familiar notes	40
6.5.1 Basic use	40
6.5.2 Customizing mark	40
6.5.3 Separator for multiple footnotes	41
6.6 Printing the footnote mark without printing the footnote text	41
6.7 Changing series	41
6.7.1 Create a new series	41
6.7.2 Delete series	41
6.7.3 Series order	41
6.8 Position of critical and familiar footnotes	42
<b>7 Apparatus customization</b>	<b>43</b>
7.1 Notes arrangement in a series	43
7.2 Control line number printing	44
7.2.1 Print line number only at first time	44
7.2.2 Print line number annotation only the first time	44
7.2.3 Print page number only at first time	45
7.2.4 Arbitrary text before line number	46
7.2.5 Separator for line range	46
7.2.6 Abbreviate line range	46
7.2.7 Disable line number	47
7.2.8 Printing pstart number	47
7.2.9 Printing stanza number	47

7.2.10 Separator between line and subline numbers . . . . .	47
7.2.11 Separator between page and line numbers . . . . .	48
7.2.12 Space around number . . . . .	48
7.2.13 Space around line symbol . . . . .	48
7.2.14 Space in place of number . . . . .	48
7.2.15 Boxing line number and line symbol . . . . .	48
7.3 Arbitrary code around line number . . . . .	49
7.4 Separator between the lemma and the note . . . . .	50
7.4.1 For footnotes . . . . .	50
7.4.2 For endnotes . . . . .	50
7.5 Font style . . . . .	51
7.5.1 For line number . . . . .	51
7.5.2 For the lemma . . . . .	51
7.5.3 For all notes . . . . .	51
7.6 Wrapping notes . . . . .	51
7.6.1 Wrapping lemmas . . . . .	51
7.6.2 Wrapping contents . . . . .	52
7.7 Indent of notes content . . . . .	52
7.8 Arbitrary code at the beginning of notes . . . . .	52
7.9 Arbitrary code before inserting note . . . . .	52
7.10 Options for footnotes in columns . . . . .	53
7.10.1 Alignment . . . . .	53
7.10.2 Size of the columns . . . . .	53
7.11 Options for paragraphed footnotes and notes grouped by line . . . . .	53
7.11.1 Mark separation of notes . . . . .	53
7.11.2 Ragged text . . . . .	54
7.12 Options for block of notes . . . . .	54
7.12.1 Grouping notes by line . . . . .	54
7.12.2 Text before notes . . . . .	54
7.12.3 Code before notes . . . . .	55
7.12.4 Spacing . . . . .	55
7.12.5 Rule . . . . .	55
7.12.6 Maximum height . . . . .	55
7.12.7 Width . . . . .	56
7.13 Footnotes and the <code>reledpar</code> columns . . . . .	56
7.14 Line number annotation . . . . .	56
7.15 Endnotes in one paragraph . . . . .	56
<b>8 Fonts</b>	<b>57</b>
<b>9 Verse</b>	<b>57</b>
9.1 Basic . . . . .	57
9.2 Define stanza indents . . . . .	57
9.3 Repeating stanza indents . . . . .	58
9.4 Notes about the scope of the indent settings . . . . .	59
9.5 Manual stanza indent . . . . .	59

9.6 Stanza breaking . . . . .	59
9.7 Hanging symbol . . . . .	60
9.8 Long verse and page break . . . . .	60
9.9 Content before/after verses . . . . .	60
9.10 Numbering stanza . . . . .	61
9.11 Antilabe . . . . .	61
9.12 Various tools . . . . .	62
9.13 Notes on empty lines . . . . .	62
<b>10 Grouping</b>	<b>62</b>
<b>11 Cross referencing</b>	<b>63</b>
11.1 Basic use . . . . .	63
11.2 Cross-referencing to a critical note . . . . .	64
11.3 Cross-referencing which return a number in any case . . . . .	64
11.3.1 Cross-referencing in order to define line number of a critical note . . . . .	64
11.4 Not automatic cross-referencing . . . . .	65
11.5 Normal $\text{\LaTeX}$ cross-referencing . . . . .	65
11.6 References to start and end lines . . . . .	65
11.6.1 Reference to main text lines . . . . .	65
11.6.2 References to lines that are commented on in the apparatus . . . . .	66
11.6.3 Settings . . . . .	66
11.6.4 Combining multiple references . . . . .	68
11.7 Compatibility with <code>xr</code> package . . . . .	68
<b>12 Sidenotes</b>	<b>68</b>
12.1 Basics . . . . .	68
12.2 Setting . . . . .	69
12.2.1 Width . . . . .	69
12.2.2 Vertical position . . . . .	69
12.2.3 Distance to the main text . . . . .	69
12.2.4 Font . . . . .	69
12.2.5 Separator between notes . . . . .	69
<b>13 Indexing</b>	<b>70</b>
13.1 Basics . . . . .	70
13.2 Use with <code>imakeidx</code> or <code>indextools</code> . . . . .	70
13.3 Referring to critical notes . . . . .	70
13.4 Separator between page and line numbers . . . . .	71
13.5 Using <code>xindy</code> . . . . .	71
13.6 Advanced setting . . . . .	72
<b>14 Glossary</b>	<b>72</b>
14.1 Preamble setting . . . . .	72
14.2 Commands . . . . .	72
<b>15 Tabular material</b>	<b>73</b>

<b>16 Sectioning commands</b>	<b>76</b>
16.1 Sectioning commands without line numbers or critical notes . . . . .	76
16.2 Sectioning commands with line numbering and critical notes . . . . .	76
16.3 Optimization . . . . .	77
<b>17 Quotation environments</b>	<b>77</b>
<b>18 Page breaks</b>	<b>77</b>
18.1 Control page breaking . . . . .	77
18.2 Prevent page break in a long verses . . . . .	78
<b>19 Miscellaneous</b>	<b>78</b>
19.1 Known and suspected limitations . . . . .	79
19.1.1 Non-standard geometry . . . . .	79
19.1.2 floatrow package compatibility . . . . .	79
19.1.3 ‘No room for a new’ . . . . .	79
19.1.4 Marginal notes . . . . .	79
19.1.5 Paragraph shape . . . . .	80
19.1.6 Paragraphed footnotes . . . . .	80
19.1.7 Use with other packages . . . . .	80
19.2 Parallel typesetting . . . . .	81
<b>I Implementation overview</b>	<b>82</b>
<b>II Preliminaries</b>	<b>82</b>
II.1 Links with original edmac . . . . .	82
II.2 Package declaration . . . . .	82
II.3 Package options . . . . .	83
II.3.1 Options of reledpar . . . . .	83
II.3.2 Options of reledmac . . . . .	83
II.4 Loading packages . . . . .	85
II.5 Compatibility with Lua $\TeX$ . . . . .	86
II.6 Boolean flags . . . . .	86
II.7 Messages . . . . .	87
II.8 Gobbling . . . . .	94
II.9 Wrapping content . . . . .	94
II.10 Miscellaneous commands . . . . .	95
II.11 Prepare reledpar . . . . .	95
II.12 Booleans provided by other optional packages which are required in any case . . . . .	96
<b>III Sectioning commands</b>	<b>97</b>
<b>IV List macros</b>	<b>102</b>

<b>V Line counting</b>	<b>103</b>
V.1 Choosing the system of lineation . . . . .	103
V.2 Line number margin . . . . .	105
V.3 Line number initialization and increment . . . . .	106
V.4 Line number locking . . . . .	107
V.5 Line number style . . . . .	108
V.6 Line number printing . . . . .	109
V.7 Line number counters and lists . . . . .	110
V.8 Line number locking counter . . . . .	111
V.9 Line number associated to lemma . . . . .	111
V.10 Reading the line-list file . . . . .	115
V.11 Commands within the line-list file . . . . .	117
V.12 Writing to the line-list file . . . . .	130
<b>VI Marking text for notes</b>	<b>138</b>
VI.1 <code>\edtext</code> itself . . . . .	139
VI.2 Substitute lemma . . . . .	147
VI.3 Substitute line numbers . . . . .	147
VI.4 Lemma disambiguation . . . . .	149
<b>VII Paragraph decomposition and reassembly</b>	<b>155</b>
VII.1 Boxes, counters, <code>\pstart</code> and <code>\pend</code> . . . . .	156
VII.2 Processing one line . . . . .	162
VII.2.1 General process . . . . .	162
VII.2.2 Process for “normal” line . . . . .	163
VII.2.3 Process for line containing <code>\eledsection</code> command . . . . .	165
VII.2.4 Hooks . . . . .	166
VII.2.5 Sidenotes and marginal line number initialization . . . . .	167
<b>VIII Line and page number computation</b>	<b>168</b>
VIII.1 Continuous line numbering between parallel typesetting and normal typesetting . . . . .	171
<b>IX Line number annotation</b>	<b>172</b>
<b>X Line number printing</b>	<b>174</b>
<b>XI Pstart number printing inside</b>	<b>179</b>
<b>XII Restoring footnotes and penalties</b>	<b>180</b>
XII.1 Add insertions to the vertical list . . . . .	180
XII.2 Penalties . . . . .	182
XII.3 Printing leftover notes . . . . .	183
XII.4 Text before notes . . . . .	184

<b>XIII Critical footnotes</b>	<b>186</b>
XIII.1 Fonts . . . . .	186
XIII.2 Individual note options . . . . .	186
XIII.3 Notes language . . . . .	187
XIII.4 General survey of the way we manage notes . . . . .	188
XIII.5 General setup . . . . .	188
XIII.6 Footnotes arrangement . . . . .	189
XIII.6.1 User level macro . . . . .	189
XIII.6.2 Normal footnote . . . . .	189
XIII.6.3 Paragraphed footnotes . . . . .	195
XIII.6.4 Columnar footnotes . . . . .	202
XIII.7 Footnote paragraph indent . . . . .	209
XIII.8 Footnote hanging indent . . . . .	209
XIII.9 Critical notes presentation . . . . .	210
XIII.9.1 Font tools . . . . .	210
XIII.9.2 Pstart number in footnote . . . . .	211
XIII.9.3 Lemma printing . . . . .	211
XIII.9.4 Line number printing . . . . .	212
XIII.9.5 Footnote grouped by line . . . . .	224
<b>XIV Familiar footnotes</b>	<b>225</b>
XIV.1 Adjacent footnotes . . . . .	225
XIV.2 Regular footnotes for numbered texts . . . . .	227
XIV.3 Footnote formats . . . . .	229
XIV.4 Footnote arrangement . . . . .	229
XIV.4.1 User level macro . . . . .	229
XIV.4.2 Normal footnotes . . . . .	230
XIV.4.3 Two columns footnotes . . . . .	236
XIV.4.4 Three columns footnotes . . . . .	238
XIV.4.5 Paragraphed footnotes . . . . .	240
XIV.5 Wrapping footnote marks in hyperlink . . . . .	245
<b>XV Code common to both critical and familiar footnote in normal arrangement</b>	<b>246</b>
<b>XVI Footnotes' width for two columns</b>	<b>246</b>
<b>XVII Footnotes' order</b>	<b>248</b>
<b>XVIII Footnotes' rule</b>	<b>248</b>
<b>XIX Specific skip for first series of footnotes</b>	<b>248</b>
XIX.1 Overview . . . . .	248
XIX.2 User level command . . . . .	249
XIX.3 Internal commands . . . . .	249



<b>XX Endnotes</b>	<b>251</b>
XX.1 Internal commands	251
XX.2 User level commands	255
XX.2.1 Inserting contents to endnotes	255
XX.2.2 Printing endnotes	256
<b>XXI Generate series of notes</b>	<b>264</b>
XXI.1 Test if series is still existing	265
XXI.2 Init specific to <code>reledpar</code>	265
XXI.3 For critical footnotes	265
XXI.3.1 Options	265
XXI.3.2 Create inserts, needed to add notes in foot	267
XXI.3.3 Create commands for critical apparatus, <code>\Afootnote</code> , <code>\Bfootnote</code> etc.	267
XXI.3.4 Set standard display	270
XXI.4 For familiar footnotes	270
XXI.4.1 Options	270
XXI.4.2 Create tools for familiar footnotes ( <code>\footnoteX</code> )	271
XXI.5 The endnotes	274
XXI.5.1 The auxiliary file	274
XXI.5.2 The main macro	274
XXI.5.3 Tools	275
XXI.5.4 Internal commands	275
XXI.5.5 The options	275
XXI.6 Init standards series (A,B,C,D,E)	277
<b>XXII Setting series display</b>	<b>277</b>
XXII.1 Change series order	277
XXII.2 Test series order	278
XXII.2.1 Get the first series	278
XXII.3 Series setting	278
XXII.3.1 General way of working	278
XXII.3.2 Tools to set options	279
XXII.3.3 Tools to generate options commands	281
XXII.3.4 Options for critical notes	282
XXII.3.5 Options for familiar notes	284
XXII.3.6 Options for endnotes	285
XXII.4 Hooks for a particular footnote	287
XXII.5 Alias	289
<b>XXIII Output routine</b>	<b>289</b>
XXIII.1 Extra footnotes output	289
XXIII.2 Patching standard output's commands	292
<b>XXIV Page numbering in parallel typesetting</b>	<b>295</b>

<b>XXV Cross referencing</b>	<b>297</b>
XXV.1 Compatibility with xref . . . . .	313
<b>XXVI Sidenotes</b>	<b>314</b>
<b>XXVII Minipages and such</b>	<b>322</b>
<b>XXVIII Indexing</b>	<b>327</b>
XXVIII.1 Looking on package order . . . . .	327
XXVIII.2 Auxiliary macros for \edindex . . . . .	328
XXVIII.3 Code specific to \edindexin critical footnotes . . . . .	329
XXVIII.4 Analysis of command in indexed text . . . . .	330
XXVIII.5 Code for the formatted index . . . . .	330
XXVIII.6 Main code . . . . .	331
XXVIII.7 Hyperlink . . . . .	333
XXVIII.8 ‘innote’ and ‘notenumber’ option of indextols package . . . . .	336
<b>XXIX Glossaries</b>	<b>336</b>
<b>XXX Verse</b>	<b>339</b>
XXX.1 Hanging symbol management . . . . .	339
XXX.2 Using & character . . . . .	340
XXX.3 Code category setting . . . . .	340
XXX.4 Stanza count and indent . . . . .	341
XXX.5 Numbering stanza . . . . .	342
XXX.6 Stanza number in note . . . . .	343
XXX.7 Main work . . . . .	344
XXX.8 Restore catcode and penalties . . . . .	347
XXX.9 Dealing with antilabe . . . . .	347
<b>XXXI Apparatus of Manuscripts</b>	<b>348</b>
XXXI.1 User level macro . . . . .	348
XXXI.2 Setting macro . . . . .	350
XXXI.3 Counters and lists . . . . .	350
XXXI.4 Auxiliary file macros . . . . .	351
XXXI.5 Action macro . . . . .	352
XXXI.6 Inserting footnote . . . . .	357
XXXI.7 Other . . . . .	357
<b>XXXII Arrays and tables</b>	<b>358</b>
XXXII.1 Preamble: macro as environment . . . . .	358
XXXII.2 Tabular environments . . . . .	362
XXXII.2.1 Disabling and restoring commands . . . . .	362
XXXII.2.2 Counters, boxes and lengths . . . . .	366
XXXII.2.3 Tabular typesetting . . . . .	369
XXXII.2.4 Environments . . . . .	380

<b>XXXIII Quotation's commands</b>	<b>381</b>
<b>XXXIV Section's title commands</b>	<b>382</b>
XXXIV.1 Commands to disable some feature . . . . .	382
XXXIV.2 General overview . . . . .	382
XXXIV.3 <code>\beforeeledchapter</code> command . . . . .	383
XXXIV.4 Auxiliary commands . . . . .	383
XXXIV.5 Patching standard commands . . . . .	384
XXXIV.6 Main code of <code>\eledxxx</code> commands . . . . .	389
XXXIV.7 Macros written in the auxiliary file . . . . .	392
<b>XXXV Page breaking or no page breaking depending of specific lines</b>	<b>394</b>
<b>XXXVI Long verse: prevents being separated by a page break</b>	<b>396</b>
<b>XXXVII Tools for <code>hyperref</code> package</b>	<b>397</b>
<b>XXXVIII Compatibility with <code>eledmac</code></b>	<b>397</b>
<b>Appendix A Things to do when changing versions</b>	<b>400</b>
A.1 Migrating from <code>edmac</code> to <code>ledmac</code> . . . . .	400
A.2 Migration from <code>ledmac</code> to <code>eledmac</code> . . . . .	401
A.3 Migration to <code>eledmac</code> 1.5.1 . . . . .	402
A.4 Migration to <code>eledmac</code> 1.12.0 . . . . .	402
A.5 Migration to <code>eledmac</code> 17.1 . . . . .	403
A.6 Migration to <code>eledmac</code> 1.21.0 . . . . .	403
A.6.1 <code>\Xledsetnormalparstuff</code> and <code>\ledsetnormalparstuffX</code> . . .	403
A.6.2 Endnotes . . . . .	403
A.7 Migration to <code>eledmac</code> 1.22.0 . . . . .	403
A.8 Migration to <code>eledmac</code> 1.23.0 . . . . .	403
A.9 Migration from <code>eledmac</code> to <code>reledmac</code> . . . . .	404
A.9.1 Risk of 'no room for a new' . . . . .	404
A.9.2 Multiple indices with <code>memoir</code> . . . . .	404
A.9.3 Deprecated commands and options . . . . .	404
A.9.4 <code>\renewcommand</code> replaced by <code>command</code> . . . . .	405
A.9.5 Commands the names of which have been changed . . . . .	405
A.9.6 Endnotes . . . . .	407
A.9.7 Z Series . . . . .	407
A.9.8 Internal commands . . . . .	407
A.10 Migration to <code>reledmac</code> 2.1.0 . . . . .	407
A.11 Migration to <code>reledmac</code> 2.1.3 . . . . .	407
A.12 Migration to <code>reledmac</code> 2.3.0 . . . . .	407
A.13 Migration to <code>reledmac</code> 2.4.0 . . . . .	408
A.14 Migration to <code>reledmac</code> 2.5.0 . . . . .	408
A.15 Migration to <code>reledmac</code> 2.7.0 . . . . .	408
A.16 Migration to <code>reledmac</code> 2.7.2 . . . . .	408
A.17 Migration to <code>reledmac</code> 2.8.0 . . . . .	408

A.18 Migration to reledmac 2.13.1 . . . . .	408
A.19 Migration to reledmac 2.18.0 . . . . .	409
A.20 Migration to reledmac 2.21.0 . . . . .	409
A.21 Migration to reledmac 2.24.0 . . . . .	409
A.22 Migration to reledmac 2.26.0 . . . . .	409
A.23 Migration to reledmac 2.27.1 . . . . .	409
A.24 Migration to reledmac 2.30.0 . . . . .	409
A.25 Migration to reledmac 2.31.1 . . . . .	410
A.26 Migration to reledmac 2.32.7 . . . . .	410
A.27 Migration to reledmac 2.33.0 . . . . .	410
<b>Appendix B Auxiliary softwares</b>	<b>410</b>
B.1 <i>samewords</i> . . . . .	410
B.2 critical-keys for <i>Emacs</i> . . . . .	410
B.3 critical-marks for <i>Emacs</i> . . . . .	410
B.4 Import from TEI . . . . .	410
B.5 Import from TEI . . . . .	410
<b>References</b>	<b>412</b>
<b>Index</b>	<b>412</b>
<b>Change History</b>	<b>465</b>

## 1 Introduction

### 1.1 Aim of the package

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

- automatic stepped line numbering, by page, section or paragraph;
- sub-lineation within the main series of line numbers;
- variant readings automatically keyed to line numbers;
- caters 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 `reledmac` 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 *Rithmachie* 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 *Rithmachie* 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. Gädke 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` ( $\LaTeX$  `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 TeX macro.

### 1.2.3 `eledmac`

Important changes were put in version 1.0, to make `ledmac` more easily extensible (see 7 p. 43). 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 A.2 p. 401.

### 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 A.9 p. 404.

## 1.3 Bibliography

A collaborative list of works edited with (r)(e)ledmac is available at [https://www.zotero.org/groups/209265/critical\\_editions\\_typeset\\_with\\_edmac\\_ledmac\\_eledmac\\_and\\_reledmac/](https://www.zotero.org/groups/209265/critical_editions_typeset_with_edmac_ledmac_eledmac_and_reledmac/). Please add your own edition made with (r)(e)ledmac.

If you write a book or an article about (r)(e)ledmac, please add it on the collaborative bibliography on <https://www.zotero.org/groups/1024519/>.

## 1.4 Acknowledgements

Maïeul Rouquette may thank his predecessor, and all the `reledmac` community. Especially, he may thanks all the people who checks English spell and grammar for the handbook.

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

The `reledmac` package is a three-pass package like  $\LaTeX$  itself. Although your textual apparatus and line numbers will be printed on the first run, it takes two more compilations by  $\LaTeX$  to be sure that everything is correctly placed, and one more if you typeset right-to-left text with  $\XeLaTeX$ . 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  $\LaTeX$  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  $\LaTeX$ '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 7.12.6 p. 55.

If you use the  $\text{Lua}\TeX$  engine, you need  $\text{Lua}\TeX$  1.1.0 or later. 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.  $\TeX$  is not able to create folder. You should create it yourself.
2. The option does not change the default  $\LaTeX$  auxiliary files (`.aux`, `.toc`, ...).

**antilabe** enable `\antilabe` feature (see 9.11 p. 61);

**eledmac-compat** help to migrate from `eledmac` to `reledmac` (see A.9.5 p. 405);



**nopenalties** must be called in some cases when using paragraphed endnotes (see ?? 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. 77);

**noresetlinenumannotation** does not reset the annotations to line number at each line (see 5.5 p. 25);

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

**swcaseinsensitive** make `\sameword` command case insensitive;

**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 (13.5 p. 71).

## 4.2 Optimizing package performance

**nocritical** disables tools for critical footnotes (`\Afootnote`, `\Bfootnote` etc.). If you do not need critical footnotes, this option lets `reledmac` run faster. It will also preserve room for other packages.

**noeledsec** disables tools for `\eledsection` and related commands (16.2 p. 76).

**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 `reledmac` 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. 62). 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 `\autopar` command needs to be called inside a `\beginnumbering... \endnumbering` structure, as follows:

```

\beginnumbering
\autopar

A paragraph of numbered text.

Another paragraph of numbered
text.

\endnumbering

```

`\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 an optional argument in brackets. Its content will be printed before the beginning of `\pstart` / after the end of `\pend` instead of the argument of `\AtEveryPstart` / `\AtEveryPend`.

---

<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 avoid that `\noindent`.

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

The second optional argument of `\pstart` / `\pend` replaces 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. 57) or `reledpar` (see 19.2 p. 81).

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 avoid 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 at the sides of the text. 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`.

**This feature must be used with caution.**

- It should not be use if you have critical footnotes after `\numberlinefalse`.

- It could be used in the following cases:
  - You want only familiar footnotes, not critical footnotes;
  - You want only parallel typesetting (using `reledpar`) but no critical footnotes.
  - You disable, for a portion of text, line numbering
- It must not be used if:
  - You do not want to have the line number in the margins, but do want to use critical footnotes. In this case, set `\firstlinenum` to a big value, such as 100,000 (5.3.2 p. 22).
  - You want to freeze the line numbering for some line, but still keep critical footnotes. In this case, use `\startlock` (5.4.2 p. 24).

A successor to `reledmac` may disable this feature.

### 5.3.2 Setting lineation start and step

`\firstlinenum`  
`\linenumincrement`

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`  
`\sublinenumincrement`  
`\linenumberlist`

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

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}
```

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.

`\Xsublinesep` You can change the separator between the line number and the subline number ei-  
`\Xsublinesepside` ther by using `\Xsublinesep` without any optional argument (7.2.10 p. 47) or by using `\Xsublinesepside`. 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 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. 57.

`\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: 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>}` 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 Adding annotations to line numbers

You may want to have two or more numbers associated with a line of text. Consider, for example, the following cases:

- You want to split a line of verse into two parts depending upon some stylistic / rhythmic / linguistic convention.
- You want to add the line number used by a previous edition of the work.
- You want to typeset biblical text, and use division in verse.

In such instances, you must add the second number manually, as `reledmac` can't determine a general pattern for such numbering, which depends too heavily upon the edited text.<sup>14</sup>

`\linenumannotation` To resolve this issue, `reledmac` allows you to annotate line numbers using the following command:

```
\linenumannotation{<annotation>}
```

The annotation can contain any textual value (whether number, text, or other) such as the line number from the older edition. Here are some characteristics of line number annotation:

- An annotation is associated with a line of text. It is reset with each new line of text, unless you use the `noresetlinenumannotation` option of `reledmac`.
- It is printed alongside the line number in the margin.
- It is printed following the line number in critical footnotes and endnotes.
- And of particular interest: the annotation is printed in the critical note *only* when the `\edtext` is issued after the `\linenumannotation`.

<sup>14</sup>However, you may create your own commands dealing with such patterns which will in turn call `reledmac` commands.

- If two or more annotations are called before an `\edtext`, the last one is used in associated notes.
- If an annotation is called within an `\edtext`, it is printed as an annotation for the end of the lemma.

Here is an example, supposing we are on line 1:

```
\edtext{first lemma}
\linenumannotation{A}
\edtext{second lemma}{...}
\edtext{third%
  \linenumannotation{B}
lemma}{...}
\edtext{fourth lemma}&
```

In the critical notes, the line number will be followed

- by nothing in the first lemma, as there is no annotation for this line;
- by “A” for the second lemma;
- by “A-B” for the third lemma, as it starts while annotation A is still in effect and it finishes after annotation B has already taken effect;
- by “B” for the fourth lemma.

### 5.5.1 Modifying annotation associated with a specific note

`\lineannot` The `\linenum` (6.2.5 p. 32) macro changes the line number passed to a note. The `\lineannot` macro does the same for the annotations. It takes one argument, containing the beginning and ending annotations separated by a pipe (`|`). As with `\linenum`, if one wishes to change some specific part of the annotation, one can use

```
\lineannot{[b]}% to change the ending annotation
\lineannot{a|}% to change the beginning annotation
\lineannot{a|b}% to change both
\lineannot{\empty|}% to delete the beginning annotation
```

### 5.5.2 Changing the position of the annotation

By default, annotations are placed after line numbers in both margin and footnote.

To change this behaviour, one can use one of the following commands:

- `\Xlinenumannotationposition[⟨s⟩]{⟨v⟩}` changes the position in critical footnotes with `⟨s⟩` for the series of footnotes to which it applies, where `⟨s⟩` may be a comma-separated list of series. When `⟨s⟩` is empty, the change applies also to annotations at the sides of the page, alongside the line number.

`\Xlinenumannotationposition`  
`\Xendlinenumannotationposition`

- `\Xlinenumannotationpositionside{⟨v⟩}` changes the position at the sides of the page, alongside the line numbers.

Note that `\Xlinenumannotationposition` without any optional argument will override this setting.

- `\Xendlinenumannotationposition[⟨s⟩]{⟨v⟩}` changes the position in critical endnotes, `⟨s⟩` for the series of endnotes to which it applies, where `⟨s⟩` may once again be a comma-separated list of series.

In each of these three commands, `⟨v⟩` can be *after* or *before* to indicate position with regards to the line number.

### 5.5.3 Changing the macro in which annotations are wrapped

By default, annotations are wrapped in the `\textsuperscript` macro. You can modify this using one of the following:

- `\Xwraplinenumannotation[⟨s⟩]{⟨cmd⟩}` for critical footnotes; `⟨s⟩` indicates the series of notes to which it applies and may be a comma-separated list of series. When `⟨s⟩` is empty, the change applies to the annotations in the margins also—around the line number—as well as to the annotations referenced using the `\annotationref` command of the crossref mechanism (11 p. 63).

- `\Xwraplinenumannotationside{⟨cmd⟩}` for annotations in the margins only.

Note that `\Xwraplinenumannotation` without any optional argument will override this setting.

- `\Xwraplinenumannotationref{⟨cmd⟩}` for annotations referenced using the `\annotationref` command of the crossref mechanism (11 p. 63).

Note that `\Xwraplinenumannotation` without any optional argument will override this setting.

- `\Xendwraplinenumannotation[⟨s⟩]{⟨cmd⟩}` for critical endnotes, where `⟨s⟩` indicates the series of notes to which it applies, which may be a comma-separated list.

`⟨cmd⟩` is a command which can take one argument; the backslash is required though.

For example, if you do not wish to have annotations in the margins, but do want to have them in bold in the critical footnotes and endnotes, you say:<sup>15</sup>

```
\makeatletter
\Xwraplinenumannotation{\textbf}
\Xendwraplinenumannotation{\textbf}
\Xwraplinenumannotationside{\@gobble}
\makeatother
```

---

<sup>15</sup>The `\@gobble` command takes one argument, and returns nothing.

### 5.5.4 Not printing the second identical annotations

If you print only line number annotation, you may get in critical notes something like 6-6, because the starting and ending annotations are the same. You can avoid that using `\Xnoidenticallinenumannotation[⟨s⟩]`, for critical footnotes or `\Xendnoidenticallinenumannotation[⟨s⟩]`

`\Xnoidenticallinenumannotation`  
`\Xendnoidenticallinenumannotation`

### 5.5.5 Separator between annotations

If there are more than one annotation in the same line, they are separated by a comma. If you want to change the separator, you can use `\setlinenumannotationsep{⟨sep⟩}`

`\setlinenumannotationsep`

### 5.5.6 Annotations in the apparatus

Some other options for annotations in the apparatus are described in 7.2.2 p. 44

## 5.6 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.

## 5.7 Executing code at a specific line

Sometime, you want to execute a code at a precise point in the flow of your text. For example, if the current header of the page corresponds to the position inside the text, you may want to use `\markboth` (or similar) command. The main problem is that you need to execute `\markboth` when `reledmac` typesets the text, and not when it reads it, because that could create damage with vertical spacing.

`\doinsidethislinehook`

The `\doinsidethislinehook{⟨code⟩}` macro allows `⟨code⟩` to be executed when the text is typeset. The `⟨code⟩` will be executed in the line equivalent to the position of `\doinsidethislinehook` in the flow of the text.

For example, with the following code:

```
\beginnumbering
\pstart
...
1.\doinsidelinehook{\markboth{1}{1}} a
\pend
```

`\markboth{1}{1}` will be executed at the line the 1. will be typeset. The `⟨code⟩` of `\doinsidethislinehook` will be executed only at the second run of `ℒTEX`.

See the file `examples/2-subdivision-number-in-header.tex` for an example. Especially, the use of this command with  $\TeX$  counters is quite complex, but the example show you how to deal with this problem.

## 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 footnotes are maintained; each macro takes one argument like `\Afootnote{<text>}`. When all of the five 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. 41.

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. 7.2.6 p. 46).
- `nonum` disables line numbering for this note. A horizontal blank space is added instead. You can use `\Xinplaceofnumber` to set it (7.2.14 p. 48).
- `nosep` to disable the lemma separator for this note. A horizontal blank space is added instead. You can use `\Xinplaceoflemmaseparator` to set it (7.4.1 p. 50).
- `linrangesep=<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>}`.

`<option>` can contain a comma-separated list of values. Allowed values are:

- `fulllines` to disable `\Xendtwolines` and `\Xendmorethantwolines` features for this particular note (cf. 7.2.6 p. 46).
- `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 `\Xendingplaceoflemmaseparator` to set it (7.4.2 p. 50).
- `linrangesep=<c>` to change to `<c>` 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{<s>}`, where `<s>` is the letter of the series to be printed. Put this command where you want the corresponding set of endnotes printed. In this case, all the endnotes of the `<s>` series are printed, for all numbered 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{<s>}`. For each value of `<s>`, the first call of the command will print the notes for the first series, the second call will print the notes for the second series etc. For example, do:

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

Note that by default inside endnotes no separator is used between the lemma and the content. However you can use the `\Xendlemmaseparator` macro to define one (7.4.2 p. 50).

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[<series>]{<content>}` where `<series>` is a comma-separated list of the series of endnotes where `<content>` must be inserted. If `<series>` is empty, then `<content>` 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>16</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:

```
\Atoendnotes*{\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:

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

`\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.

```
\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.

<sup>16</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.



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  $\langle lemma \rangle$  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. 63) 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>17</sup>

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

## 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 with the `\sameword` command:

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

<sup>17</sup>We use `\newcommand` and `\newcommandx` instead of classical `\let` command because the `edtabular` environments have to modify the notes definition, and we need to use the newest definition of notes. Read the handbook of `xargs` to know more about `\newcommandx`.

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 Case setting

By default, `\sameword` is sensitive to the case. E.g. “foo” is considered as a different word to “Foo”.

You can use the `swcaseinsensitive` when loading the package to make `\sameword` insensitive to the case, i.e. to consider “foo” as identical “Foo”.

### 6.3.3 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 French, “é” has two possible Unicode codepoints:

- LATIN SMALL LETTER E (U+0065) + COMBINING ACUTE ACCENT (U+0301)
- LATIN SMALL LETTER E WITH ACUTE (U+00E9)

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>18</sup> with the `buffer` option set to true.

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

### 6.3.4 Notes about right-to-left texts

Due to some internal limits of  $\text{XeTeX}$ , `\sameword` does not work with right-to-left text with this engine.

If you need to use `\sameword` with right-to-left text, you must use  $\text{LuaTeX}$ .

<sup>18</sup><https://github.com/michal-h21/uninormalize>.

### 6.3.5 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}
    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 `[⟨X⟩]`. `⟨X⟩` is the depth of the `\edtext` where the `\lemma` is used. So if the `\lemma` is called in a `\edtext` inside another `\edtext`, `⟨X⟩` is equal to 2. If the `\lemma` is called in a `\edtext` “of first level”, `⟨X⟩` is equal to 1. If the lemma is called in both 1 and 2 `\edtext` depth, `⟨X⟩` is 1, 2. If that word is referenced in the lemma of every `\edtext` depth, `⟨X⟩` can also be set to `inlemma`.

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

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

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

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

In the following 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`.

<div style="display: flex; align-items: center;"> <div style="border: 1px solid black; padding: 2px 5px; margin-right: 5px;"> 1<sub>inlemma</sub> </div> <div style="border: 1px solid black; padding: 2px 5px; margin-right: 5px;"> 2<sub>3</sub> </div> </div>	1...3 4 5 <sub>1</sub>	1...5
--	------------------------	-------

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: 100%;"/>	
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

### 6.3.6 Sameword for a group of words

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

First, you can consider only the individual words, and not groups of word. For 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

In this case it is clear which “per causam” is meant.

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

### 6.3.7 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}%
}
```

### 6.3.8 Problems with some macros

`\swnoexpands` Macros inside \sameword that are not fully expandable, mainly macros which manipulate font features, write on full or have optional argument, may cause problems during compilation. Custom commands inside \sameword may therefore result in errors saying that “Use of sameword doesn’t match its definition.” To solve this, include a redefinition of your custom commands in the \swnoexpands macro. In order to not include any content of a macro during comparison, identify the command with \@gobble For example:

```
\makeatletter
\appto{\swnoexpands}{%
  \let\somemacro\@gobble%
}
\makeatother
```

This will drop the content of `\somemacro` during comparison.

To include the content of the first and only one argument of a custom command in sameword comparison, use the `\@firstofone` command. For example, this is how `\emph` is handled:

```
\makeatletter
\appto{\swnoexpands}{%
  \let\emph\@firstofone%
}
\makeatother
```

To include command which can take optional argument, use `\RenewExpandableDocumentCommand` of `\xparse`. For example, this is how `\edindex` is handled:

```
\makeatletter
\appto{\swnoexpands}{%
  \RenewExpandableDocumentCommand{\edindex}{om}{}%
}
\makeatother
```

### 6.3.9 Automatic sameword annotation

All potentially ambiguous apparatus entries must be annotated manually. That annotation process is laborious and includes a risk of errors. *Samewords* is a Python script that can automate this step of the process. It can be installed via the *Python Package Index*, but see <https://github.com/stenskjær/samewords> for more info and documentation. The script is still at a beta stage, so comments and questions as well as error reports are very welcome at <https://github.com/stenskjær/samewords/issues>.

Please note that the maintainer of this script is not identical with the maintainer of `reledmac`.

## 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 L<sup>A</sup>T<sub>E</sub>X 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 “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. 43). However, the following *additional* commands are available.

`\setmsdataseries` The series used by default for the apparatus of manuscripts is series A. However, you can change it with `\setmsdataseries{<series>}`.

`\setmsdatalabel` 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 `\setmsdatalabel{<txt>}`.

`\setmsdataposition` 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:  
`\msdataposition{regular-msdata}`  
 And restore the default behaviour using `\msdataposition{msdata-regular}`

## 6.5 Familiar notes

### 6.5.1 Basic use

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

The footnote can take a first optional argument to manually define the footnote number or footnote symbol. For example:

```
\footnoteA[22]{This footnote will be numbered 22}
```

### 6.5.2 Customizing mark

`\thefootnoteA` Each series uses a set of macros for styling the marks. The mark numbering scheme of series A is defined by the `\thefootnoteA` macro; the default is:

`\bodyfootmarkA` `\renewcommand*{\thefootnoteA}{\arabic{footnoteA}}`

`\footfootmarkA` The appearance of the mark in the text is controlled by `\bodyfootmarkA` which is defined as:

```
\newcommand*{\bodyfootmarkA}{%
  \hbox{\textsuperscript{\normalfont\@nameuse{@thefnmarkA}}}}
```

The command `\footfootmarkA` controls the appearance of the mark at the start of the footnote text. It is defined as:

```
\newcommand*{\footfootmarkA}{\textsuperscript{\@nameuse{@thefnmarkA}}}
```

There are similar command triples for the other series.



### 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.

`\multfootsep` `\multfootsep` 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  $\text{\TeX}$ '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`      However in some specific cases, you need to change the series order at some point  
`\seriesatend`      inside the document. You can use `\seriesatbegin{⟨s⟩}` to pull up a given series `⟨s⟩` to  
the beginning, or `\seriesatend{⟨s⟩}` to push it down to the end.

## 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:

`⟨series⟩⟨type⟩`

`⟨series⟩` is a series letter (A,B,C etc.), while `⟨type⟩` 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  $\TeX$ .

## 7 Apparatus customization

Some commands can be used to change the display of the footnotes. All can have an optional argument [*s*], which is the letter of the series — or a list of letters separated by comma — depending on which option is applied. If the optional argument is omitted or empty, the setting will 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 L<sup>A</sup>T<sub>E</sub>X to have:  $a - c \leq m \leq a + b$ . If you use some relative unit<sup>19</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. 80.
- `twocol` formats them as separate paragraphs, but in two columns;
- `threecol`, in three columns.
- `normal`, restore normal arrangement.

You should set up the page layout parameters, and in particular the `\baselineskip` of the footnotes, before you call this macro because its action depends on these; too much or too little space will be allotted for the notes on the page if these macros use the wrong values.

---

<sup>19</sup>Like `em` which is the width of an ‘m’ in a given font.

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.

`\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 inside every note. If you want to print it only the first time for a given line number (i.e., once for line 1, once 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  $\langle symbol \rangle$  must be robust. Use `\robustify` to robustify a non-robust command.

`\Xendnumberonlyfirstinline`

`\Xendnumberonlyfirstintwolines`

`\Xendsymlinenum`

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

### 7.2.2 Print line number annotation only the first time

`\Xlinenumannotationonlyfirst`

By default, the line number annotation (5.5 p. 25) is printed in every note. If you want to print it only the first time for a given annotation you can use `\Xlinenumannotationonlyfirst[⟨s⟩]`.

Note the two following points:

- If you use the `noresetlinenumannotation` option of `reledmac`, the `\Xlinenumannotationonlyfirst` won't take account of the official line number.
- The `\Xlinenumannotationonlyfirst` works for consecutive lemmas with the same annotation. That is, if you have for example a lemma with an A annotation, then a lemma with a B annotation, then a lemma with an A annotation, each lemma will have its own annotation in the apparatus.

`\Xlinenumannotationonlyfirstintwo`

Suppose you have a lemma on annotation 1 and a lemma between annotation 1 and annotation 3. With `\Xlinenumannotationonlyfirst`, the second lemma is considered to have the same annotation as the first lemma. But if you use both

`\Xlinenumannotationonlyfirst[⟨s⟩]` and `\Xlinenumannotationonlyfirstintwo[⟨s⟩]`, a distinction is made.

`\Xsymlinenumannotation`

To use a particular symbol in place of the line number annotation, you can use `\Xsymlinenumannotation[⟨s⟩]{⟨symbol⟩}` in combination with `\Xlinenumannotationonlyfirst[⟨s⟩]`. From the second lemma with the same line number annotation, the symbol will be used instead of the annotation. Note that any command called in `⟨symbol⟩` must be robust. Use `\robustify` to robustify a non-robust command.

`\Xlinenumannotationonlyfirst`

`\Xlinenumannotationonlyfirstintwo`

`\Xendsymlinenumannotation`

For endnotes, `\Xendlinenumannotationonlyfirst`; `\Xendlinenumannotationonlyfirstintwo` and `\Xendsymlinenumannotation` are the equivalents of `\Xlinenumannotationonlyfirst`; `\Xlinenumannotationonlyfirstintwo` and `\Xsymlinenumannotation`.

### 7.2.3 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:

`\Xendpagenumberonlyfirst`

- `\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 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.

### 7.2.4 Arbitrary text before line number

`\Xtxtbeforenumber` `\Xtxtbeforenumber[⟨s⟩]{⟨txt⟩}` allows the insertion of `⟨txt⟩` before the line number only when the line number is printed, so taking into account `\Xnumberonlyfirstinline` and the like.

`\Xendbeforepagenumber` **For endnotes** `\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, it will use the value defined by `\Xendlineprefixsingle`.

### 7.2.5 Separator for line range

`\Xlinerangeseparator` `\Xendlinerangeseparator` 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.

### 7.2.6 Abbreviate line range

`\Xtwolines` `\Xmorethantwolines` 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)”.

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 ‘[fullines]’ argument in the note macro cf. 6.2.2 p. 30.

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

### 7.2.7 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.

### 7.2.8 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.

`Xpstartonlyfirst` You can use `\Xpstartonlyfirst[⟨s⟩]` to print the pstart number only the first time it appears in the apparatus. When using `\Xpstart`, you can use `\Xpstartseparator[⟨s⟩]{⟨text⟩}` to print `⟨text⟩` after the pstart number. Default value is empty.

### 7.2.9 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.

`Xstanzaonlyfirst` You can use `\Xstanzaonlyfirst[⟨s⟩]` to print the stanza number only the first time it appears in the apparatus.

### 7.2.10 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 changes the separator in side numbering.**

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

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

The default value is `\textnormal{.}`.

### 7.2.11 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.

### 7.2.12 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, neither is the space. 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.

### 7.2.13 Space around line symbol

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

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

`\Xendbeforesymmlinenum` `\Xendbeforesymmlinenum` and `\Xendaftersymmlinenum` are the equivalents of `\Xbeforesymmlinenum` and `\Xaftersymmlinenum` for the endnotes.

### 7.2.14 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.

### 7.2.15 Boxing line number and line symbol

`\Xboxlinenum` It could be useful to put the line number inside a fixed box: the content of the note will be printed after this box. You can use `\Xboxlinenum[⟨s⟩]{⟨l⟩}` to do that. To subsequently



disable this feature, use `\Xboxlinenum` with length equal to 0 pt. One use of this feature is to print line number in a column, and the note in an other column:

```
\Xhangindent{1em}
\Xafternumber{0em}
\Xboxlinenum{1em}
```

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

`\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 Arbitrary code around line number

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

`\Xendahooklinenumber` `\Xendahooklinenumber[⟨s⟩]{⟨code⟩}` is used to execute code after line number in endnotes. The code is executed after the `\Xendafternumber` 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.

## 7.4 Separator between the lemma and the note

### 7.4.1 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>20</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 a 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.

### 7.4.2 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>\langle Xendlemmaseparator \rangle</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 choose to remove the lemma separator. The default value is 0.5 em.

<sup>20</sup>For polyglossia, when the lemma is RTL, the bracket automatically switches to a left bracket.

## 7.5 Font style

### 7.5.1 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> .

### 7.5.2 For the lemma

<code>\lemmadisablefontselection</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>\lemmadisablefontselection</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.

### 7.5.3 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> .

## 7.6 Wrapping notes

### 7.6.1 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 `bid` package, to ensure having a lemma written right to left, use `\Xwraplemma{\RL}`.

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

### 7.6.2 Wrapping contents

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

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

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

`\wrapcontentX`      `\wrapcontentX[\s]{\cmd}` is the same for familiar footnotes.

## 7.7 Indent of notes content

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

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

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

## 7.8 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[\s]{\code}` is to be used at the beginning of each critical footnote.

`\bhooknoteX`      `\bhooknoteX[\s]{\code}` is to be used at the beginning of each familiar footnote.

`\Xendbhooknote`      `\Xendbhooknote[\s]{\code}` is to be used at the beginning of each endnote.

## 7.9 Arbitrary code before inserting note

`\Xbeforeinserting`      `\Xbeforeinserting[\s]{\code}` and `\beforeinsertingX[\s]{\code}` 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 (7.6.1 p. 51).

Note that the changes are local to the footnote.

## 7.10 Options for footnotes in columns

### 7.10.1 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.

*s* must be one of the following command:

`\justifying` to have text justified, as usual with  $\text{\LaTeX}$ . You can also let *s* 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`).

### 7.10.2 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 \hspace`.

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

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

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

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

### 7.11.1 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 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.

### 7.11.2 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.

## 7.12 Options for block of notes

### 7.12.1 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. 44).

`\Xgroupbylineseparetwolines` Note that the `\Xafternote` and `\Xparafootsep` settings are used to determine space and content between footnotes (7.11 p. 53). 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. 44).

### 7.12.2 Text before notes

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

`\txtbeforenotesX` You can add text before familiar footnotes with `\txtbeforenotesX[⟨s⟩]{⟨text⟩}`.

`\Xendtxtbeforenotes` You can add text before endnotes with `\Xendtxtbeforenotes[⟨s⟩]{⟨text⟩}`. The text will be typeset only if there are endnotes.

`\Xtxtbeforenotesonlyonce` By default, such texts are inserted at the beginning of the groups of notes on each page. You can add `\Xtxtbeforenotesonlyonce` (for critical footnotes) and `\txtbeforenotesonlyonceX` (for familiar footnotes) to insert them only the first time notes are typeset.

### 7.12.3 Code before notes

`\Xbhookgroup` While `\Xtxtbeforenotes` is for typesetting text before notes, `\Xbhookgroup[⟨s⟩]{⟨code⟩}` and `\bhookgroupX[⟨s⟩]{⟨code⟩}` (for critical and familiar respectively) are for executing code before a group of notes, between the rules and the printing of the notes.

### 7.12.4 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.

### 7.12.5 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.**

### 7.12.6 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 todoclist.

### 7.12.7 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`, `\Xsizethreecol`, `\hsizetwocolX`, `\hsizethreecolX`, see 7.10.2 p. 53.

### 7.13 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.

### 7.14 Line number annotation

The way line number annotations are typeset can be changed using hooks as described in 5.5.2 p. 26 and 5.5.3 p. 27.

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

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

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

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

## 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`. Otherwise, the brackets will be interpreted as delimitation of an optional argument (cf. 9.9 p. 60)

### 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  
`\setstanzaindent{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. 9.7 p. 60.

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 A.3 p. 402.**

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 macro makes no restriction on the number of lines in a stanza.

Sometime, it is useful to repeat the indents only after some lines of verse. You can set the counter `startstanzaindentrepetition` to do that. Its default value is 1, that means the repetition starts with line of verse number 1.

For example, you may want to have no indent for the first line, and all other lines been indented by 1. So do

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

```
\setcounter{stanzaindentrepetition}{1}
\setcounter{startstanzaindentrepetition}{2}
```

## 9.4 Notes about the scope of the indent settings

Stanza indentation values (and penalty values) obey  $\TeX$ 's grouping conventions, so if one stanza among several has a different structure, its indentations (penalties) may be set within a group; the prior values will be restored when the group ends.

## 9.5 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`.

## 9.6 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 the 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.

## 9.7 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}
```

## 9.8 Long verse and page break

If you want to prevent page breaks inside long verses, use the option `nopbinverse` when loading package, or use `\lednopbinverse>true`. Read 18.2 p. 78 for further details.

## 9.9 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.
```

`\AtEveryStanza`

- Use `\AtEveryStanza{<arg>}` to automatically add content before stanzas (not in the same paragraph).

Note that a `\noindent` will be inserted before the argument, and, consequently, a `\parskip`. You can use the starred version of `\AtEveryStanza` to avoid this `\noindent`.

`\AtStartEveryStanza`

- Use `\AtStartEveryStanza` to automatically add content at the beginning of stanzas (in the same paragraph).

- `&` can be replaced by `\newverse` with two optional arguments (in brackets). The first will be printed after the current verse, the second before the next verse.

A `\noindent` is automatically inserted before the contents of these optional arguments.

Use a third and fourth optional argument to not add these `\noindents` (to add content respectively after the current verse / before the next verse).

- Use `\AtEveryPend{⟨arg⟩}` to automatically add content after verses (including the final one) and `\AtEveryPstart{⟨arg⟩}` to automatically add content before verses (including the first one).
- `\&` can take an optional argument (in brackets). Its content will be printed after the stanza.

`\AtEveryStopStanza`

- Use `\AtEveryStopStanza` to automatically add content after the end of stanzas (not in the same paragraph).

Note that a `\noindent` will be inserted before the argument, and, consequently, a `\parskip`. You can use the starred version of `\AtEveryStopStanza` to avoid this `\noindent`.

`\BeforeEveryStopStanza`

- Use `\BeforeEveryStopStanza` to automatically add content at the end of stanzas (in the same paragraph).

## 9.10 Numbering stanza

`\numberstanzatrue`

If you want to automatically number stanzas, use `\numberstanzatrue`. In this case, the line number will restart at each `\stanza`.

`\numberstanzafalse`

If you want to disable this feature again, use `\numberstanzafalse`.

You can use this feature in combination with `\Xstanza` (7.2.9 p. 47).

`\thestanza`

You can redefine `\thestanza` to change the aspect of stanza number. Default value is:

```
\renewcommand{\thestanza}{%
  \textbf{\arabic{stanza}}%
}
```

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}%
}
```

`\antilabe`

## 9.11 Antilabe

In dramatic text, it may happen that a line of verse is split between two speakers. That is called “antilabe”. In this case, the typesetting of the second half must start at the horizontal position where the typesetting of the first half stops.

`Reledmac` provides tool for such problematic, but you must enable them with the `antilabe` option when loading package, as this option is quite resource consuming.

The `\antilabe{<name>}` must be called at the beginning of the second half of the line of verse.

Its `<name>` argument is the name of the speaker. Because of the way  $\TeX$  expand argument, you can use one the following syntax:

- `\antilabe{<name>}`, if you typeset the character name;
- `\antilabe<command>`, if the character name is defined in a command, like with the `thalie` package;
- `\antilabe{}`, if you do not type the character name.

See the file `examples/2-antilabe.tex` for an example of use.

`\beforeantilabe` You can execute any arbitrary code before or after `\antilabe` macro just with a re-definition of `\beforeantilabe` or `\afterantilabe`. For example, to add more space, just do:

`\afterantilabe`

```
\renewcommand{\afterantilabe}{\hspace{1em}}
```

## 9.12 Various tools

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

## 9.13 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 (9.9 p. 60).

# 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}[\langle pos \rangle]{\langle width \rangle}`.

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

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

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

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

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

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

## 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{\langle lab \rangle}`.  $\langle lab \rangle$  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>21</sup>

`\edpageref``\edlineref``\sublineref``\pstartref``\annotationref`

Elsewhere in the text, either before or after the `\edlabel`, you can refer to its location with `\edpageref{\langle lab \rangle}`, `\edlineref{\langle lab \rangle}`, `\edsublineref{\langle lab \rangle}`, `\pstartref{\langle lab \rangle}` or `\annotationref{\langle lab \rangle}`, that will produce, respectively, the page, line, sub-line, `pstart`, the annotation, on which the `\edlabel{\langle lab \rangle}` 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.)

<sup>21</sup>More precisely, you should stick to characters in the `TEX` categories of “letter” and “other”.

## 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. 66).

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

`\xpageref`  
`\xlineref`  
`\xsublineref`  
`\xpstartref`  
`\xannotationref`

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. 32).

For this situation, `reledmac` supplies variants of the reference commands, with the `x` prefix: `\xpageref`, `\xlineref`, `\xsublineref`, `\xpstartref` and `\xannotationref`. They have the following 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` (6.2.5 p. 32) and `\lineannot` (5.5.1 p. 26) and sets the beginning page, line, subline numbers and line annotations to those of the place where `\edlabel{mouse}` was placed, and the ending ones 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}\Afootnote{}}
\pend

\endnumbering

```

## 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 $\LaTeX$ 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 two labels 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 end-notes.

`\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. 32) 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.

### 11.6.3 Settings

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

`\setSerefonlypageprefixsingle` 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. 66). So if you use `\Xendbeforepagenumber` without any optional argument, it will override this setting.**

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

These commands are the following:

- `\Xlineflag` (for `reledpar`), enabled by default.

- `\Xlinerangeseparator`
- `\Xmorethantwolines`
- `\Xsublinesep`
- `\Xtwolines`
- `\Xtwolinesbutnotmore`
- `\Xtwolinesonlyinsamepage`
- `\Xlinenumannotationposition`
- `\Xwraplinenumannotation`
- `\Xnoidenticallinenumannotation`

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`
- `\Xendlinenumannotationposition`
- `\Xendwraplinenumannotation`
- `\Xemdnoidenticallinenumannotation`

**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. 30).

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. 30).

### 11.6.4 Combining multiple references

When combining multiple cross references into a list, it is possible to prevent prefixes reappearing by adding an optional argument to `\appref` and `\Seref`. The available options are:

- `noprefix` to remove any prefix set by `\setapprefprefixsingle` or the equivalent for a single reference;
- `prefixmore` to force the plural version of a prefix.

For example, a reference to ‘lines 65–66, 72, and 75’ might use `\Seref[prefixmore]{ref1}`, `\Seref` to achieve this result.

## 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 Sidenotes

### 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 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 setting of `\sidenotemargin{<location>}`. The permissible value for `<location>` is one out of the list `left`, `right`, `inner`, or `outer`, for example `\sidenotemargin{outer}`.  
`\ledrightnote`  
`\ledsidenote`  
`\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  $\text{\LaTeX}$  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  
`\ledrsnotefontsetup` are:

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

These can of course be changed to suit.

### 12.2.5 Separator between notes

`\setsidenotesep` If you have two or more sidenotes for the same line, they are separated by a comma. But if you want to change this separator, you can use `\setsidenotesep{<sep>}`.

## 13 Indexing

### 13.1 Basics

`\edindex`  $\TeX$  provides the `\index{⟨item⟩}` command for specifying that  $\langle item \rangle$  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  $\langle item \rangle$  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.

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

### 13.2 Use with `imakeidx` or `indextools`

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`.

### 13.3 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.

`\Xinnotemark` You can redefine the way the reference is made using :

`\innotemarkX`

- `\Xinnotemark[⟨series⟩]{⟨definition⟩}` for critical notes;
- `\innotemarkX[⟨series⟩]{⟨definition⟩}` for familiar notes;

In the  $\langle definition \rangle$  argument, you can use the following codes:

- #1, which stands for footnote series;
- ##1, which stands for reference (ie. page+line)

For example:

```
\Xinnotemark{##1\textsuperscript{#1}}
```

### 13.4 Separator between page and line numbers

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

- is the default separator used by the MAKEINDEX program.

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>22</sup>.

```
page_compositor ":"
```

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

### 13.5 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-linenummer`

An example of such a file is provided in the "examples" folder. Read the `xindy` handbook to learn how to use it.<sup>23</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 choose to use both `xindy` and the `hyperref` package, you must do three more things:

<sup>22</sup>For further detail, you can read <http://tex.stackexchange.com/a/32783/7712>.

<sup>23</sup>Or, for people who read French, read <http://geekographie.maieul.net/174>.

1. Use `xindy+hyperref` option when loading the `reledmac` package. When you run (Xe/Lua)TeX with this option, a `.xdy` configuration file will be generated with all the settings needed to allow internal hyperlinking in each index entry which is created by `\edindex`.
2. Use `hyperindex=false` option when loading `hyperref`.
3. Uncomment — by removing the semicolons at the beginning of the relevant lines — some lines in the `<code>.xdy</code>` file provided in the “examples” folder in order to restore internal links in the index to be used by the standard `index` command.<sup>24</sup>

### 13.6 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 (13.4 p. 71).

### 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>]
```

---

<sup>24</sup>These are the recommended lines to provide the best possible compatibility between `hyperref` and `xindy`, even without using `reledmac`.



## 15 Tabular material

L<sup>A</sup>T<sub>E</sub>X's normal `tabular` and `array` environments cannot be used where line numbering is being done; more precisely, they can be used but with odd results, so don't use them. However, `reledmac` provides some simple tabulation environments that can be line numbered. The environments can also be used in normal unnumbered text.

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

```

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

```

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

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

For example:

```

\beginnumbering
\pstart
\begin{edtabularl}
\textbf{\Large I} & wish I was a little bug\edindex{bug} &
\textbf{\Large I} & eat my peas with honey\edindex{honey} \\
& With whiskers \edtext{round}{\Afootnote{around}} my tummy &
& I've done it all my life. \\
& I'd climb into a honey\edindex{honey} pot &
& It makes the peas taste funny \\
& And get my tummy gummy.\edindex{gummy} &
& But it keeps them on the knife.
\end{edtabularr}
\pend
\endnumbering

```

produces the following parallel pair of verses.

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

`\edtabcolsep` The distance between the columns is controlled by the length `\edtabcolsep`.

`\spreadmath`     `\spreadmath{⟨math⟩}` typesets  $\{⟨math⟩\}$  but the  $\{⟨math⟩\}$  has no effect on the calculation of column widths. `\spreadtext{⟨text⟩}` is the analagous command for use in edtabular environments.

```

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

```

$$\begin{array}{cccc}
1 & 2 & 3 & 4 \\
& F + G + C & & \\
a & bb & ccc & dddd
\end{array}$$

`\edrowfill`     The macro `\edrowfill{⟨start⟩}{⟨end⟩}{⟨fill⟩}` fills columns number  $\langle start \rangle$  to  $\langle end \rangle$  inclusive with  $\langle fill \rangle$ . The  $\langle fill \rangle$  argument can be any horizontal ‘fill’. For example, `\hrulefill` or `\upbracefill`.

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

The `\edrowfill` macro can be used in both tabular and array environments. The typeset appearance of the following code is shown below.

```

\begin{edtabularr}
1 & & & & & & & & & & \\
Q & & & & fd & & h & & & & qwertziohg \\
v & & & & wptz & & x & & y & & vb \\
g & & & & nnn & & & & & & \edrowfill{3}{5}{\upbracefill} & & \\
\edrowfill{1}{3}{\downbracefill} & & & & & & & & pq & & dgh \\
k & & & & & & 1 & & co & & ghweropjklmnbvcxys \\
1 & & & & 2 & & 3 & & & & \edrowfill{4}{5}{\hrulefill} & & \\
\end{edtabularr}

```

$$\begin{array}{ccccccccc}
1 & & 2 & & 3 & & 4 & & & & 5 \\
Q & & & & & & fd & & h & & qwertziohg \\
v & & wptz & & x & & y & & & & vb \\
g & & nnn & & & & & & & & \\
k & & & & & & & & pq & & dgh \\
1 & & 2 & & 3 & & & & & & \\
\end{array}$$

You can also define your own ‘fill’. For example:

```

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

```

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

```

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

```

$$\begin{array}{cccc} 1 & 2 & 3 & 4 \\ a & \perp & & d \\ A & B & C & D \end{array}$$

`\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}
```

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

```

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. 19):

```

\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

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.**

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 p.  $n$ , and l. 444 at 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

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 `\lednopbinversetrue` 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

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.

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.

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 7.12.6 p. 55 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. 77 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`

$\TeX$  is a three-pass system, but even after a document has been processed three times, there are some tricky situations in which the page breaks decided by  $\TeX$  never settle down. At each successive run, `reledmac` may oscillate between two different sets of page decisions. To stop this happening, should it arise, Wayne Sullivan suggested the inclusion of the quantity `\ballast`. The amount of `\ballast` will be subtracted from the penalties which apply to the page breaks calculated on the *previous* run through  $\TeX$ , thus reinforcing these breaks. So if you find your page breaks oscillating, 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 Paraphed footnotes

The restriction on explicit line-breaking in paraphed footnotes, mentioned on 7.1 p. 43, and described in more detail on XIII.6.3 p. 197, 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 paraphed 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 paraphed 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` (7.12.6 p. 55), 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>25</sup> you will find  $\TeX$  whinging ‘Missing { inserted’, and then things start to fall apart. The trick in this case is to specify either:

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

or

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

(`\@secondoftwo` is an internal  $\TeX$  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

<sup>25</sup>Reported by Dirk-Jan Dekker in the CTT thread ‘Incompatibility of “color” package’ on 2003/08/28.

## I Implementation overview

We present the `reledmac` code in roughly the order in which it is used during a run of  $\TeX$ . The order is *exactly* that in which it is read when you load the `reledmac` package, because the same file is used to generate this manual and to generate the  $\LaTeX$  package file.

Most of what follows consists of macro definitions, but there are some commands that are executed immediately—especially at the start of the code. The documentation generally describes the code from the point of view of what happens when the macros are executed, though. As each macro is introduced, its name is printed in the margin.

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

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

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

## II Preliminaries

### II.1 Links with original `edmac`

Generally, these are the modifications to the original. `edmac` code:

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

### II.2 Package declaration

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

```

1 %<*code>
2 \NeedsTeXFormat{LaTeX2e}
3 \ProvidesPackage{reledmac}[2020/10/10 v2.34.6 typesetting critical editions]
4 %

```

## II.3 Package options

`\ifparledgroup` Use this to remember which option is used, set and execute the options with final as the default. We use `xkeyval` in order to manage options with argument.

```

\ifsameparallelpagenumber
\ifprevpgnotnumbered
\ifledfinal
\ifnocritical@
\if@noeled@sec
\ifnoend@
\ifnofamiliar@
\ifnoledgroup@
\ifparapparatus@
\ifnoquotation@
\iflednopbinverse
\ifwidthliketwocolumns
\ifxindy@
\ifxindyhyperref@
\ifeledmaccompat@
\ifantilabe@

```

### II.3.1 Options of reledpar

Here, we define some booleans specific to `reledpar` options, but which have consequences on `reledmac` internal commands.

```

7 \newif\ifparledgroup
8 \newif\ifsameparallelpagenumber%
9 \newif\ifprevpgnotnumbered%%
10 %
11 \DeclareOptionX{series}[A,B,C,D,E]{\xdef\default@series{#1}}
12 \ExecuteOptionsX{series}%
13
14 \newif\if@noeled@sec%
15 \DeclareOptionX{noeledsec}{\@noeled@sectrue}
16
17 \newif\ifnocritical@%
18 \DeclareOptionX{nocritical}{\nocritical@true}%
19
20 \newif\ifnofamiliar@%
21 \DeclareOptionX{nofamiliar}{\nofamiliar@true}%
22
23 \newif\ifnoledgroup@%
24 \DeclareOptionX{noledgroup}{\noledgroup@true}%
25
26 \newif\ifnoend@%
27 \DeclareOptionX{noend}{%
28   \let\l@dend@open\@gobble%
29   \let\l@dend@close\relax%
30   \global\let\l@dend@stuff=\relax%
31   \noend@true%
32 }%

```

### II.3.2 Options of reledmac

```

33 \newif\ifnoquotation@
34 \DeclareOptionX{noquotation}{\noquotation@true}
35
36 \newif\ifledfinal
37 \DeclareOptionX{final}{\ledfinaltrue}
38 \DeclareOptionX{draft}{\ledfinalfalse}
39 \ExecuteOptionsX{final}
40
41 \newif\ifparapparatus@
42 \DeclareOptionX{parapparatus}{\parapparatus@true}
43
44 \newif\iflednopbinverse
45 \DeclareOptionX{nopbinverse}{\lednopbinversetrue}
46
47 \newif\ifwidthliketwocolumns%
48 \DeclareOptionX{widthliketwocolumns}{\widthliketwocolumnstrue}%
49
50 \newif\ifcontinuousnumberingwithcolumns
51 \DeclareOptionX{continuousnumberingwithcolumns}{\
52   continuousnumberingwithcolumnstrue}%
53
54 \newif\ifxindy@
55 \DeclareOptionX{xindy}[eledmac-markup-attr.xdy]{%
56   \AtBeginDocument{\immediate\openout\eledmac@xindy@out=#1}%
57   \newwrite\eledmac@xindy@out%
58   \xindy@true%
59   \gdef\eledmacmarkuplocdepth{:depth 1}%
60   \AtEndDocument{\immediate\closeout\eledmac@xindy@out}%
61 }%
62
63 \newif\ifxindyhyperref@
64 \DeclareOptionX{xindy+hyperref}{%
65   \xindyhyperref@true%
66 }%
67
68 \newif\ifeledmaccompat@%
69 \DeclareOptionX{eledmac-compat}{%
70   \eledmaccompat@true%
71 }%
72 \DeclareOptionX{nopenalties}{%
73   \AtBeginDocument{\let\add@penalties\relax}%
74 }
75 \def\l@auxdir{}%
76 \DeclareOptionX{auxdir}{%
77   \xdef\l@auxdir{#1}%
78 }%
79
80 \newif\ifsw@caseinsensitive%
81 \DeclareOptionX{swcaseinsensitive}{%

```

```

82 \sw@caseinsensitivetrue%
83 }%
84 \newif\ifnoresetlinenumannotation@
85 \DeclareOptionX{noresetlinenumannotation}{%
86 \noresetlinenumannotation@true%
87 }%
88 \newif\ifantilabe@%
89 \DeclareOptionX{antilabe}{%
90 \antilabe@true%
91 }%
92 %

```

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.

```

93 \ProcessOptionsX*\relax
94
95 %

```

## II.4 Loading packages

Loading package `xargs` to declare commands with optional arguments. Loading package `xparse` to declare fully expandable commands with optional argument. Ideally, we should use only `xparse` and not `xargs`. For historical reasons, we use both. `Etoolbox` is also used to make code clearer - for example, in dynamic command names (which can replace `\csname` etc.). Use suffix to declare commands with a starred version, `xstring` to work with strings, `ifluatex` and `ifxetex` to test if Lua<sub>TeX</sub> or X<sub>TeX</sub> is running, and `ragged2e` to manage ragged justification for paragraphed notes.

```

96 \RequirePackage{xargs}
97 \RequirePackage{xparse}[2017/03/07]%
98 \RequirePackage{etoolbox}
99 \@ifl@t@r\fmtversion{2015/10/01}
100 {\ifboolexpr{not test{\@ifl@t@r\fmtversion{2016/03/31}}} or (test{\
101 ifdefstring{\fmtversion}{2016/03/31}} and test {\ifnumless{\patch@level
102 }{3}})}%
103 {\PackageWarning{reledmac}{You are using a LaTeX version older than
104 2016/03/31 patch 3.%
105 \MessageBreak You are strongly encouraged to use a newer version.}}%
106 {}%
107 }%
108 {\RequirePackage{etex}%
109 \csname reserveinserts\endcsname{32}%
110 }%
111 \RequirePackage{suffix}
112 \RequirePackage{xstring}
113 \RequirePackage{ifluatex}
114 \RequirePackage{ragged2e}

```

```

112 \RequirePackage{ifxetex}%
113 %

```

## II.5 Compatibility with Lua<sub>T</sub><sub>E</sub>X

Here, we enable some primitives for Lua<sub>T</sub><sub>E</sub>X.

```

114 \ifx\directlua\undefined\else%
115   \directlua{tex.enableprimitives("",{"texdir","pdir","bodydir"})}
116 \fi
117 \ifluatex%
118   \ifnum\luatexversion<110%
119     \PackageWarning{reledmac}{You need to use LuaTeX 1.1.0 or higher}{\@ehc}
120   }%
121   \fi%
122 %

```

## II.6 Boolean flags

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

```

123 \newif\ifl@dmemoir
124 \@ifclassloaded{memoir}{\l@dmemoirtrue}{\l@dmemoirfalse}
125
126 %

```

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

```

127 \newif\if@ledgroup%
128 %

```

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

```

129 \newif\ifl@imakeidx
130 \@ifpackageloaded{imakeidx}{\l@imakeidxtrue}{\l@imakeidxfalse}%False is the default value
131 %

```

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

```

132 \newif\ifl@indextools%
133 \@ifpackageloaded{indextools}{%
134   \l@indextoolstrue%
135   \l@imakeidxtrue%
136   \let\imki@wrindexentry\indtl@wrindexentry%
137 }{}%
138 %

```

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.

```
139 \newif\ifl@footmisc
140 \ifpackageloaded{footmisc}{\l@footmisctrue}{}%False is the default value
141 %
```

`\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.

```
142 \ifdef{\if@RTL}{\newif\if@RTL}
143 %
```

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

```
144 \newif\if@firstlineofpage%
145 \newif\if@firstlineofpageR%
146 %
```

## 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.

```
147 \newcommand{\reledmac@warning}[1]{\PackageWarning{reledmac}{#1}}
148 %
```

`\reledmac@error` Write an error message.

```
149 \newcommand{\reledmac@error}[2]{\PackageError{reledmac}{#1}{#2}}
150 %
```

```
\led@err@NumberingStarted51 \newcommand*\led@err@NumberingStarted{%
d@err@NumberingNotStarted52 \reledmac@error{Numbering has already been started}{\@ehc}}
NumberingShouldHaveStarted53 \newcommand*\led@err@NumberingNotStarted{%
154 \reledmac@error{Numbering was not started}{\@ehc}}
155 \newcommand*\led@err@NumberingShouldHaveStarted{%
156 \reledmac@error{Numbering should already have been started}{\@ehc}}
157 %
```

```

\led@err@samewordRTL158 \newcommand*{\led@err@samewordRTL}{%
159   \reledmac@error{You can't use \string\sameword \MessageBreak with XeLaTeX
\MessageBreak when typesetting RTL text.\MessageBreak Please use LuaTeX
instead.}\@ehc}%
160 }%
161 %

```

```

\led@err@edtextoutsidepstart162 \newcommand*{\led@err@edtextoutsidepstart}{%
\led@err@PstartInEdtext163   \reledmac@error{\string\edtext\space outside numbered paragraph (\pstart\
\led@err@PendInEdtext   space ... \space\pend)}{\@ehc}}%
164
165 \newcommand{\led@err@PstartInEdtext}[1]{%
166   \reledmac@error{\string\pstart\space in \string\edtext\space #1 argument
}\@ehc}%
167 }%
168
169 \newcommand{\led@err@PendInEdtext}[1]{%
170   \reledmac@error{\string\pend\space in \string\edtext\space #1 argument}{\
\@ehc}%
171 }%
172 %

```

```

\led@mess@NotesChanged173 \newcommand*{\led@mess@NotesChanged}{%
174   \typeout{\reledmac reminder: }%
175   \typeout{ The number of the footnotes in this section
176     has changed since the last run.}%
177   \typeout{ You will need to run LaTeX two more times
178     before the footnote placement}%
179   \typeout{ and line numbering in this section are
180     correct.}}
181 %

```

```

\led@mess@SectionContinued182 \newcommand*{\led@mess@SectionContinued}[1]{%
183   \message{Section #1 (continuing the previous section)}%
184 %

```

```

\led@err@LineationInNumbered185 \newcommand*{\led@err@LineationInNumbered}{%
186   \reledmac@error{You can't use \string\lineation\space within
187     a numbered section}{\@ehc}}
188 %

```

```

\led@warn@BadLineation189 \newcommand*{\led@warn@BadLineation}{%
\led@warn@BadLinenummargin190   \reledmac@warning{Bad \string\lineation\space argument}}
\led@warn@BadLockdisp191 \newcommand*{\led@warn@BadLinenummargin}{%
\led@warn@BadSubblockdisp

```



```

192 \reledmac@warning{Bad \string\linenummargin\space argument}}
193 \newcommand*{\led@warn@BadLockdisp}{%
194 \reledmac@warning{Bad \string\lockdisp\space argument}}
195 \newcommand*{\led@warn@BadSublockdisp}{%
196 \reledmac@warning{Bad \string\sublockdisp\space argument}}
197 %

```

```

\led@warn@NoFile% \newcommand*{\led@warn@NoFile}[1]{%
199 \reledmac@warning{File `#1' not found}}
200 %

```

```

\led@warn@LineFileObsolete% \newcommand*{\led@warn@Obsolete}[1]{%
202 \reledmac@warning{Line-list file #1 was obsolete. We have not read it.
Please run LaTeX again.}}
203 %

```

```

\led@warn@BadAdvancelineSubline% \newcommand*{\led@warn@BadAdvancelineSubline}{%
\led@warn@BadAdvancelineLine% \reledmac@warning{\string\advanceline\space produced a sub-line
206 number less than zero.}}
207 \newcommand*{\led@warn@BadAdvancelineLine}{%
208 \reledmac@warning{\string\advanceline\space produced a line
209 number less than zero.}}
210 %

```

```

\led@warn@BadSetline% \newcommand*{\led@warn@BadSetline}{%
\led@warn@BadSetlinenum% \reledmac@warning{Bad \string\setline\space argument}}
213 \newcommand*{\led@warn@BadSetlinenum}{%
214 \reledmac@warning{Bad \string\setlinenum\space argument}}
215 %

```

```

\led@err@PstartNotNumbered% \newcommand*{\led@err@PstartNotNumbered}{%
\led@err@PstartInPstart% \reledmac@error{\string\pstart\space must be used within a
\led@err@PendNotNumbered% numbered section %
\led@err@PendNoPstart% (\string\beginnumbering\space ...\space\string\
\led@err@AutoparNotNumbered endnumbering)}{\@ehc}}%
\led@err@NumberingWithoutPstart% \newcommand*{\led@err@PstartInPstart}{%
220 \reledmac@error{\string\pstart\space encountered while another
221 \string\pstart\space was in effect}{\@ehc}}
222 \newcommand*{\led@err@PendNotNumbered}{%
223 \reledmac@error{\string\pend\space must be used within a
224 numbered section}{\@ehc}}
225 \newcommand*{\led@err@PendNoPstart}{%
226 \reledmac@error{\string\pend\space must follow a \string\pstart}{\@ehc}}
227 \newcommand*{\led@err@AutoparNotNumbered}{%
228

```

```

229 \reledmac@error{\string\autopar\space must be used within a
230         numbered section}{\@ehc}}
231 \newcommand*\led@err@NumberingWithoutPstart{%
232 \reledmac@error{\string\beginnumbering...\string\endnumbering\space
without \string\pstart}{\@ehc}}%
233 %

```

```

\led@warn@BadAction34 \newcommand*\led@warn@BadAction{%
235 \reledmac@warning{Bad action code, value \next@action.}}
236 %

```

```

\led@warn@DuplicateLabel237 \newcommand*\led@warn@DuplicateLabel}[1]{%
\led@warn@AppLabelOutSecondArgEdtext238 \reledmac@warning{Duplicate definition of label `#1'\@gobble}%
\led@warn@RefUndefined239 \@latex@warning{no@line{Label `#1' multiply defined}%
\led@warn@RefUndefined240 }%
241 \newcommand*\led@warn@AppLabelOutSecondArgEdtext}[1]{%
242 \reledmac@warning{\string\applabel\space outside of the second argument
of an \string\edtext\space `#1' on page \thepage.}}%
243 \newcommand*\led@warn@RefUndefined}[1]{%
244 \G@refundefinedtrue%
245 \reledmac@warning{Reference `#1' on page \thepage\space undefined.%
246         Using `000'.}%
247 \@latex@warning{Reference `#1' undefined\on@line}%
248 }%
249 \newcommand*\led@warn@pairRefUndefined}[1]{%
250 \G@refundefinedtrue%
251 \reledmac@warning{Reference `#1:start' and/or `#1:end' on page \thepage\
space undefined.
252         Using `??'.}%
253 \@latex@warning{Reference `#1:start' and/or `#1:end' undefined\on@line}%
254 }
255 %

```

```

\led@warn@NoMarginpars56 \newcommand*\led@warn@NoMarginpars{%
257 \reledmac@warning{You can't use \string\marginpar\space in numbered text
}}
258 %

```

```

\led@warn@BadHookSeries59 \newcommand*\led@warn@BadHookSeries}[2]{%
260 \reledmac@warning{Bad series #1 for #2 hook. Skipped.}%
261 }%
262 %

```

```

\led@warn@BadSidenotemargin63 \newcommand*\led@warn@BadSidenotemargin{%
264 \reledmac@warning{Bad \string\sidenotemmargin\space argument}}
265 %

```

```
\led@warn@NoIndexFile266 \newcommand*{\led@warn@NoIndexFile}[1]{%
267   \reledmac@warning{Undefined index file #1}}
268   %
```

```
\led@warn@SeriesStillExist269 \newcommand{\led@warn@SeriesStillExist}[1]{%
270   \reledmac@warning{Series #1 is still existing !}}%
271   %
272   %
```

```
\led@err@BadAction273 \newcommand*{\led@err@StanzaIndentNotDefined}{%
274   \reledmac@error{You have not defined the indentation for the line \number
\stanza@count}{\@ehc}}%
275   %
```

```
\led@err@ManySidenotes276 \newcommand{\led@err@ManySidenotes}{%
\led@err@ManyLeftnotes277   \ifledRcol{%
\led@err@ManyRightnotes278   \reledmac@warning{\itemcount@\space sidenotes on line \the\line@numR\
space p. \the\page@numR}%
279   \else%
280   \reledmac@warning{\itemcount@\space sidenotes on line \the\line@num\
space p. \the\page@num}%
281   \fi%
282  }%
283   \newcommand{\led@err@ManyLeftnotes}{%
284   \ifledRcol{%
285   \reledmac@warning{\itemcount@\space leftnotes on line \the\line@numR\
space p. \the\page@numR}%
286   \else%
287   \reledmac@warning{\itemcount@\space leftnotes on line \the\line@num\
space p. \the\page@num}%
288   \fi%
289  }%
290   \newcommand{\led@err@ManyRightnotes}{%
291   \ifledRcol{%
292   \reledmac@warning{\itemcount@\space rightnotes on line \the\line@numR\
space p. \the\page@numR}%
293   \else%
294   \reledmac@warning{\itemcount@\space rightnotes on line \the\line@num\
space p. \the\page@num}%
295   \fi%
296  }%
297   %
```

```
\led@err@TooManyColumns298 \newcommand*{\led@err@TooManyColumns}{%
\led@err@UnequalColumns299   \reledmac@error{Too many columns}{\@ehc}}
\led@err@LowStartColumn
\led@err@HighEndColumn
\led@err@ReverseColumns
```

```

300 \newcommand*\led@err@UnequalColumns}{%
301   \reledmac@error{Number of columns is not equal to the number
302     in the previous row (or \protect\\ \space forgotten?)}{\@ehc}}
303 \newcommand*\led@err@LowStartColumn}{%
304   \reledmac@error{Start column is too low}{\@ehc}}
305 \newcommand*\led@err@HighEndColumn}{%
306   \reledmac@error{End column is too high}{\@ehc}}
307 \newcommand*\led@err@ReverseColumns}{%
308   \reledmac@error{Start column is greater than end column}{\@ehc}}
309 %

```

```

\led@err@toendnotes@outsidenumbering10 \newcommand\led@err@toendnotes@outsidenumbering}{%
311   \reledmac@error{\string\toendnotes\space and related commands must be
called inside a numbered texte (\string\beginnumbering\space ... \space\
string\endnumbering)}{\@ehc}%
312 }%
313 %

```

```

\led@err@EdtextWithoutFootnote14 \newcommand\led@err@EdtextWithoutFootnote}{%
315   \reledmac@error{edtext without Xfootnote. Check syntax}{\@ehc}%
316 }%
317 %

```

```

\led@err@FootnoteNotInSecondArgEdtext18 \newcommand\led@err@FootnoteNotInSecondArgEdtext}[1]{%
319   \reledmac@error{#1footnote outside of the second argument of an edtext.
Check syntax}{\@ehc}%
320 }%
321 %

```

```

\led@error@PackageAfterEledmac22 \newcommand\led@error@PackageAfterEledmac}[1]{%
323   \reledmac@error{#1 must be loaded before reledmac}{\@ehc}%
324 }%
325 %

```

```

\led@error@fail@patch@@makecol26 \newcommand\led@error@fail@patch@@makecol}{%
327   \reledmac@error{Fail to patch \string\@makecol\space command}{\@ehc}%
328 }%
329 %

```

```

\led@error@fail@patch@@reinserts30 \newcommand\led@error@fail@patch@@reinserts}{%
331   \reledmac@error{Fail to patch \string\@reinserts\space command}{\@ehc}%
332 }%
333 %

```

```

r@fail@patch@@doclearpage 34 \newcommand{\led@error@fail@patch@@doclearpage}{%
335 \reledmac@error{Fail to patch \string\@doclearpage\space command}{\@ehc}%
336 }%
337 %

```

```

r@fail@patch@@iiiminipage 38 \newcommand{\led@error@fail@patch@@iiiminipage}{%
339 \reledmac@error{Fail to patch \string\@iiiminipage\space command}{\@ehc}%
340 }%
341 %

```

```

r@fail@patch@endminipage 42 \newcommand{\led@error@fail@patch@endminipage}{%
343 \reledmac@error{Failed to patch the \string\endminipage\space command}{\
344 \@ehc}%
345 }%

```

```

r@fail@patch@endminipage 46 \newcommand{\led@error@fail@patch@makeindex}{%
347 \reledmac@error{Failed to patch the \string\makeindex\space command}{\
348 \@ehc}%
349 }%

```

```

r@fail@patch@setcounter 50 \newcommand{\led@error@fail@patch@setcounter}{%
351 \reledmac@error{Failed to patch the \string\setcounter\space command}{\
352 \@ehc}%
353 }%

```

```

r@fail@patch@addtocounter 54 \newcommand{\led@error@fail@patch@addtocounter}{%
355 \reledmac@error{Failed to patch the \string\addtocounter\space command}{\
356 \@ehc}%
357 }%

358 \newcommand{\led@error@fail@patch@stepcounter}{%
359 \reledmac@error{Failed to patch the \string\stepcounter\space command}{\
360 \@ehc}%
361 }%

```

```

n@edinde@outsidenumbering 62 \newcommand{\led@warn@edinde@outsidenumbering}{%
363 \reledmac@warning{\string\edindex\space called outside of \string\
beginnumbering\space ... \space \string\endnumbering. \MessageBreak
Automatically switched to \string\index.}%
364 }%
365 %

```

```

\led@warning@hsizeX@deprecated 366 \newcommand{\led@warning@hsizeX@deprecated}{%
367   \reledmac@warning{\string\hsizeX\space command deprecated, use \string\
widthX\space instead.}%
368 }%
369 %

```

```

\led@warning@Xhsize@deprecated 370 \newcommand{\led@warning@Xhsize@deprecated}{%
371   \reledmac@warning{\string\Xhsize\space command deprecated, use \string\
Xwidth\space instead.}%
372 }%
373 %

```

```

\led@warning@msdatawithoutstop 374 \newcommand{\led@warning@msdatawithoutstop}{%
375   \reledmac@warning{\string\msdata\space without corresponding \string\
stopmsdata}%
376 }%
377 %

```

```

\led@warning@preXnotes@deprecated 378 \newcommand{\led@warning@preXnotes@deprecated}{%
379   \reledmac@warning@preXnotes@deprecated%
380 }%
381 %

```

## II.8 Gobbling

Here, we define some commands which gobble their arguments.

```

\@gobblethree 382 \providecommand*\@gobblethree}[3]{}
\@gobblefour 383 \providecommand*\@gobblefour}[4]{}
\@gobbleseven 384 \providecommand*\@gobbleseven}[7]{}
385 %

```

## II.9 Wrapping content

`\l@wrap@ifnotemptybox`  
`\l@wrap@pc@ifnotemptybox`

The `\l@wrap@ifnotemptybox` macro wraps its second argument in the command passed as the first argument, but only if the second argument does not produce an empty box. That is useful for example when we need to wrap something on a `\textsuperscript`.

Indeed `a\textsuperscript{b}` does not produce the same typographical result as `ab`.<sup>26</sup> The `\l@wrap@pc@ifnotemptybox` does the same, but take as argument a control sequence name, so without backslash.

<sup>26</sup>Compare the distance between letters in the first case (ab) and in the second one (ab).

```

386 \newcommand{\l@wrap@ifnotemptybox}[2]{%
387   \setbox0=\hbox{#2}%
388   \ifdim\wd0=\z@ \else%
389     #1{#2}%
390   \fi%
391 }%
392 \newcommand{\l@wrapcs@ifnotemptybox}[2]{%
393   \l@wrap@ifnotemptybox{\csname #1\endcsname}{#2}%
394 }%
395 %

```

## II.10 Miscellaneous commands

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

```

396 \ifl@dfinal
397   \newcommand*{\showlemma}[1]{#1}
398 \else
399   \newcommand*{\showlemma}[1]{\underline{#1}}
400 \fi
401
402 %

```

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

Initialize it as `\empty`.

```

403 \let\linenumberlist=\empty
404
405 %

```

`\@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).

```

406 \newcount\@l@tempcnta \newcount\@l@tempcntb
407 %

```

## II.11 Prepare reledpar

`\ifl@dpairing` In preparation for the `reledpar` package, these are related to the ‘right’ text of parallel  
`\ifl@dpaging` texts (when `\ifl@dpairing` is TRUE). They are explained in the `reledpar` manual.  
`\ifl@dprintingpages`  
`\ifl@dprintingcolumns` `\newif\ifl@dpairing`  
`\ifpst@rtedL` `\newif\ifl@dpaging%`  
`\l@dnumppstartsl` `\newif\ifl@dprintingpages%`

```

411 \newif\ifl@dprintingcolumns%
412 \newif\ifpstartL
413 \newcount\l@dnumstartL
414 %

```

`\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`.

```

415 \newif\ifledRcol
416 \newif\ifledRcol@
417 %

```

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

```

418 \newif\ifnumberingR
419 %

```

`\ifafterendnumberingR` The `\ifafterendnumberingR` boolean is set to TRUE at `\endnumberingR` and to FALSE at next `\beginnumberingR` or real `\beginnumbering` (not `\resumenummering`). It is mainly used for the `continuousnumberingwithcolumns`, to avoid incrementing right column line counters after a `\endnumberingR`.

```

420 \newif\ifafterendnumberingR%
421 %

```

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

```

422 \newif\ifXnote@%
423 %

```

## II.12 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.

```

424 \providebool{indtl@innote}%
425 \providebool{indtl@notenumber}%
426 %

```

`ifinstanzaL` `ifinstanzaR` These boolean tests are switched by the `\stanza` command, using either the left or right side.

```

427 \newif\ifinstanzaL%
428 \newif\ifinstanzaR%
429 %

```



`ifinastanzaL` These boolean tests are switched by the `astanza` environment, using either the left  
`ifinastanzaR` or right side.

```
430 \newif\ifinastanzaL%
431 \newif\ifinastanzaR%
432 %
```

### 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.

```
433 \newcount\section@num
434 \section@num=0
435 \let\extensionchars=\empty
436 %
```

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

```
437 \newif\ifnumbering
438 %
```

`\beginnumbering` `\beginnumbering` begins a section of numbered text. When it is executed we increment  
`\initnumbering@reg` 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.

```

439 \newcommand*{\beginnumbering}{%
440   \ifnumbering
441     \led@err@NumberingStarted
442   \endnumbering
443 \fi
444 \global\numberingtrue
445 \global\afterendnumberingRfalse%
446 \global\advance\section@num \@ne
447 \initnumbering@reg
448 \message{Section \the\section@num }%
449 \line@list@stuff{\jobname.\extensionchars\the\section@num}%
450 \ifcontinuousnumberingwithcolumns%
451   \unless\ifafterendnumberingR%
452     \unless\ifl@dpairing%
453       \ledRcoltrue%
454       \global\advance\section@numR \@ne%
455       \message{Section \the\section@numR R (continuousnumbering)}%
456       \line@list@stuffR{\jobname.\extensionchars\the\section@numR R}%
457       \ledRcolfalse%
458     \fi%
459   \fi%
460 \fi%
461 \l@dend@stuff
462 \setcounter{pstart}{1}
463 \ifl@dpairing
464   \global\l@dnumpstartsL \z@
465   \global\pst@rtedLfalse
466 %

```

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.

```

467 \else
468   \begingroup
469   \global\@afterindenttrue%In order to reestablish normal feature if the \
beginninggroup was not here
470   \initnumbering@quote
471   \ifwidthliketwocolumns%
472     \setwidthliketwocolumns%

```

```

473     \csuse{setpositionliketwocolumns@\columns@position}%
474     \fi%
475 \fi
476 \gdef\eled@sections@@{}%
477 \if@noeled@sec\else%
478     \makeatletter%
479     \InputIfFileExists%
480         {\l@auxdir\jobname.eledsec\the\section@num}%
481         {}%
482         {\led@warn@NoFile{\l@auxdir\jobname.eledsec\the\section@num}}%
483     \makeatother%
484     \immediate\openout\eled@sectioning@out=\l@auxdir\jobname.eledsec\the\
section@num\relax%
485     \fi%
486 }
487 \newcommand*{\initnumbering@reg}{%
488     \global\pst@rtedLfalse
489     \global\l@dnumstartsL \z@
490     \global\absline@num \z@
491     \gdef\normal@page@break{}
492     \gdef\l@prev@pb{}
493     \gdef\l@prev@nopb{}
494     \global\line@num \z@
495     \global\subline@num \z@
496     \global\@lock \z@
497     \global\sub@lock \z@
498     \global\sublines@false
499     \global\let\next@page@num=\relax
500     \global\let\this@section@next@page@num=\relax%
501     \global\let\sub@change=\relax
502     \global\last@page@num=-10000%
503     \ifdefined\line@numR%
504         \line@numR=\z@%
505         \last@page@numR=\z@%
506     \fi%
507     \resetprevline@
508     \resetprevpage@num
509     \global\stopmsdata@inserted@true%
510     \global\let\@msdata@list\relax%
511     \global\csundef{@msdata@add@msd@c @data}%
512 }
513
514 %

```

`\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. We define it using `\def` and not `\newcommand` because  $\TeX$  does not allow defining command whose name starts by “end” except if we are currently creating an environment, which is not the case here.

```

515 \def\endnumbering{%
516   \ifnumbering
517     \global\numberingfalse
518     \normal@pars
519     \ifnum\l@dnumpstartsL=0%
520       \led@err@NumberingWithoutPstart%
521     \fi%
522 %

```

For the continuousnumberingwithcolumn options, we need to store a global value for the next \endnumbering. We don't modify this counter globally, because that creates problem for nested \edtext.

```

523   \global\page@num=\this@section@page@num%
524   \global\last@page@num=\this@section@last@page@num%
525   \global\let\next@page@num\this@section@next@page@num%
526   \ifl@dpairing
527     \global\pst@rtedLfalse
528   \else
529     \ifx\insertlines@list\empty\else
530       \global\noteschanged@true
531     \fi
532     \ifx\line@list\empty\else
533       \global\noteschanged@true
534     \fi
535   \fi
536   \ifnoteschanged@
537     \led@mess@NotesChanged
538   \fi
539 \else
540   \led@err@NumberingNotStarted
541 \fi
542 \autoparfalse
543 \if@noeled@sec\else%
544   \immediate\closeout\eled@sectioning@out%
545 \fi%
546 \ifl@dpairing\else
547   \global\l@dnumpstartsL=\z@%
548 \endgroup
549 \fi
550 }
551 %

```

`\pausenumbers` The `\pausenumbers` macro is just the same as `\endnumbering`, but with the `\ifnumbering` flag set to true, to show that numbering continues across the gap.<sup>27</sup> The `\pausenumbers@page@num` counter stores the `\this@section@next@page@num` when the `\pausenumbers` is called.

```

552 \newcount\pausenumbers@page@num%

```

<sup>27</sup>Peter Wilson's thanks to Wayne Sullivan, who suggested the idea behind these macros.

```

553 \newcommand{\pausenumbering}{%
554   \ifx\this@section@next@page@num\relax%
555     \global\pausenumbering@page@num=0%
556   \else%
557     \global\pausenumbering@page@num=\this@section@next@page@num%
558   \fi%
559   \ifautopar\global\autopar@pausettrue\fi%
560   \endnumbering\global\numberingtrue}
561 %

```

The `\resumenumbering` macro is a bit more involved, but not much. It does most of the same things as `\beginnumbering`, but without resetting the various counters. Note that no check is made by `\resumenumbering` to ensure that `\pausenumbering` was actually invoked. The boolean `\ifresumenumbering@start` is set to true. That allows us to avoid resetting the line number at the first line of `\resumenumbering` if the lineation is by page. This boolean is set to false after the first action.

```

562 \newif\ifresumenumbering@start%
563 \newcommand*{\resumenumbering}{%
564   \ifnumbering
565     \ifautopar@pause\autopar\fi
566     \global\pst@rtedLtrue
567     \global\advance\section@num \@ne
568     \global\resumenumbering@starttrue%
569     \led@mess@SectionContinued{\the\section@num}%
570     \set@continuousnumberingforL%
571     \line@list@stuff{\jobname.\extensionchars\the\section@num}%
572     \ifcontinuousnumberingwithcolumns%
573       \unless\ifafterendnumberingR%
574         \unless\ifl@dpairing%
575           \ledRcoltrue%
576           \global\advance\section@numR \@ne%
577           \message{Section \the\section@numR R (continuousnumbering)}%
578           \line@list@stuffR{\jobname.\extensionchars\the\section@numR R}%
579           \ledRcolfalse%
580         \fi%
581       \fi%
582     \fi%
583     \l@dend@stuff
584     \ifl@dpairing\else%
585       \begin@group%
586       \initnumbering@quote%
587       \ifwidthliketwocolumns%
588         \setwidthliketwocolumns%
589         \csuse{setpositionliketwocolumns@\columns@position}%
590       \fi%
591     \fi%
592   \else
593     \led@err@NumberingShouldHaveStarted
594   \endnumbering

```

```

595 \beginnumbering
596 \fi}
597
598
599 %

```

`\set@continuousnumberingforL` `\set@continuousnumberingforl` sets `pstart` counters at a `\beginnumbering` or a `\resumenumbering` in order to have continuous numbering with single column text.

```

600 \newcommand{\set@continuousnumberingforL}{%
601 \ifcontinuousnumberingwithcolumns%
602 \ifl@dpairing%
603 \unless\ifl@dpaging%
604 \global\c@pstartL=\c@pstart%
605 \fi%
606 \fi%
607 \fi%
608 }%
609 %

```

## 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 `TeX3` 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.

```

610 \newcommand*{\list@create}[1]{%
611 \global\let#1=\empty%
612 }%
613 %

```

`\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 .

```

614 \newcommand*{\list@clear}[1]{%
615   \global\let#1=\empty%
616 }
617 %

```

**\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.

**\led@toksb**

```

618 \newtoks\led@toksa \newtoks\led@toksb
619 \global\led@toksa={\}
620 \long\def\xright@appenditem#1\to#2{%
621   \global\led@toksb=\expandafter{#2}%
622   \xdef#2{\the\led@toksb\the\led@toksa\expandafter{#1}}%
623   \global\led@toksb={}}
624 %

```

**\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.

```

625 \long\def\xleft@appenditem#1\to#2{%
626   \global\led@toksb=\expandafter{#2}%
627   \xdef#2{\the\led@toksa\expandafter{#1}\the\led@toksb}%
628   \global\led@toksb={}}
629 %

```

**\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.

```

630 \def\gl@p#1\to#2{\expandafter\gl@poff#1\gl@poff#1#2}
631 \long\def\gl@poff\#1#2\gl@poff#3#4{\gdef#4{#1}\gdef#3{#2}}
632
633 %

```

## 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.

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

```

\ifbypstart@
\byppstart@true
\byppstart@false
  \ifbypage@
    \bypage@true
    \bypage@false

```

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

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

```
634 \newif\ifbypage@
635 \newif\ifbypstart@
636 %
```

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@R37 \newif\ifbypage@R
\ifbypstart@R38 \newif\ifbypstart@R
639 %
```

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

```
640 \newcommand*{\lineation}[1]{%
641 %
```

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

```
642 \ifnumbering
643 \led@err@LineationInNumbered
644 \else
645 %
```

If the argument is page.

```
646 \def\@tempa{#1}\def\@tempb{page}%
647 \ifx\@tempa\@tempb
648 \global\bypage@true
649 \global\bypstart@false
650 \unless\ifnocritical@%
651 \Xpstart[] [false]%
652 \fi%
653 %
```

If the argument is pstart.

```
654 \else
655 \def\@tempb{pstart}%
656 \ifx\@tempa\@tempb
657 \global\bypage@false
658 \global\bypstart@true
659 \unless\ifnocritical@%
660 \Xpstart%
661 \fi%
662 %
```

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



```

663     \else
664         \def\@tempb{section}
665         \ifx\@tempa\@tempb
666             \global\bypage@false
667             \global\bystart@false
668             \unless\ifnocritical@%
669                 \Xpstart[] [false]%
670             \fi%
671 %

```

In other case, it is an error.

```

672     \else
673         \led@warn@BadLineation
674     \fi
675 \fi
676 \fi
677 \fi}}
678 %

```

## 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@dgetline@margin`

```

679 \newcount\line@margin%
680 \newcount\line@margin@columns%Only for parallel typesetting
681 \line@margin@columns=\m@ne%
682
683 \newcommand*\linenummargin}[1]{%
684     \l@dgetline@margin{#1}%
685     \ifnum\@l@dttempcntb>\m@ne
686         \ifledRcol
687             \global\line@marginR=\@l@dttempcntb
688             \led@warn@setting@in@rightside{\linenummargin}%
689         \else
690             \global\line@margin=\@l@dttempcntb
691         \fi
692     \fi}}
693
694 \newcommand*\l@dgetline@margin}[1]{%
695     \def\@tempa{#1}\def\@tempb{left}%
696     \ifx\@tempa\@tempb
697         \@l@dttempcntb \z@
698     \else
699         \def\@tempb{right}%
700         \ifx\@tempa\@tempb
701             \@l@dttempcntb \@ne

```

```

702 \else
703 \def\@tempb{outer}%
704 \ifx\@tempa\@tempb
705 \l@dttempcntb \tw@
706 \else
707 \def\@tempb{inner}%
708 \ifx\@tempa\@tempb
709 \l@dttempcntb \thr@@
710 \else
711 \led@warn@BadLinenummargin
712 \l@dttempcntb \m@ne
713 \fi
714 \fi
715 \fi
716 \fi}
717
718 %

```

### 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; `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.

```

719 \newcounter{firstlinenum}
720 \setcounter{firstlinenum}{5}
721 \newcounter{linenumincrement}
722 \setcounter{linenumincrement}{5}
723 %

```

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

```

724 \newcounter{firstsublinenum}
725 \setcounter{firstsublinenum}{5}
726 \newcounter{sublinenumincrement}
727 \setcounter{sublinenumincrement}{5}
728
729 %

```

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

```

730 \newcommand*{\firstlinenum}[1]{%
731 \ifledRcol%
732 \setcounter{firstlinenumR}{#1}%
733 \led@warn@setting@in@rightside{\firstlinenum}%
734

```

```

735 \else%
736 \setcounter{firstlinenum}{#1}%
737 \fi%
738 }
739 \newcommand*\linenumincrement}[1]{%
740 \ifledRcol%
741 \setcounter{linenumincrementR}{#1}%
742 \led@warn@setting@in@rightside{\linenumincrement}%
743 \else%
744 \setcounter{linenumincrement}{#1}%
745 \fi%
746 }
747 \newcommand*\firstsublinenum}[1]{%
748 \ifledRcol%
749 \setcounter{firstsublinenumR}{#1}%
750 \led@warn@setting@in@rightside{\firstsublinenum}%
751 \else%
752 \setcounter{firstsublinenum}{#1}%
753 \fi%
754 }
755 \newcommand*\sublinenumincrement}[1]{%
756 \ifledRcol%
757 \setcounter{sublinenumincrementR}{#1}%
758 \led@warn@setting@in@rightside{\sublinenumincrement}%
759 \else%
760 \setcounter{sublinenumincrement}{#1}%
761 \fi%
762 }
763 %
764 %

```

## 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.

```

765 \newcount\lock@disp
766 \newcommand*\lockdisp}[1]{%
767 \l@getlock@disp{#1}%
768 \ifnum\l@dtempcntb>\m@ne
769 \global\lock@disp=\l@dtempcntb
770 \else
771 \led@warn@BadLockdisp
772 \fi}}
773 \newcommand*\l@getlock@disp}[1]{
774 \def\@tempa{#1}\def\@tempb{first}%

```

```

775 \ifx\@tempa\@tempb
776 \l@dttempcntb \z@
777 \else
778 \def\@tempb{last}%
779 \ifx\@tempa\@tempb
780 \l@dttempcntb \@ne
781 \else
782 \def\@tempb{all}%
783 \ifx\@tempa\@tempb
784 \l@dttempcntb \tw@
785 \else
786 \l@dttempcntb \m@ne
787 \fi
788 \fi
789 \fi}
790
791 %

```

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

```

792 \newcount\sublock@disp
793 \newcommand{\sublockdisp}[1]{\{%
794 \l@dtgetlock@disp{#1}%
795 \ifnum\l@dttempcntb>\m@ne
796 \global\sublock@disp=\l@dttempcntb
797 \else
798 \led@warn@BadSublockdisp
799 \fi}}
800
801 %

```

## V.5 Line number style

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

```

802 \newcommand*{\linenumberstyle}[1]{\%
803 \def\linenumrep##1{\@nameuse{##1}}}%
804 \newcommand*{\sublinenumberstyle}[1]{\%
805 \def\sublinenumrep##1{\@nameuse{##1}}}%
806 %

```

Initialise the number styles to arabic.

```

807 \linenumberstyle{arabic}
808 \let\linenumr@p\linenumrep
809 \sublinenumberstyle{arabic}
810 \let\sublinenumr@p\sublinenumrep
811
812 %

```

## V.6 Line number printing

`\leftlinenum` `\rightlinenum` and `\rightlinenum` are the macros that are called to print marginal line numbers on a page, for left- and right-hand margins respectively. They are made easy to access and change, since you may want to change the styling in some way. These standard versions illustrate the general sort of thing that will be needed; they are based on the `\leftheadline` macro in *The TeXbook*, p. 416.

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

`\ledlinenum` typesets the line (and subtitle) number.

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

```

813 \newlength{\linenumsep}
814 \setlength{\linenumsep}{1pc}
815 \newcommand*{\numlabfont}{\normalfont\scriptsize}
816 \newcommand*{\ledlinenum}{%
817   \bgroup%
818   \ifluatex%
819     \texdir TLT%
820   \fi%
821   \numlabfont%
822   \ifdefstring{\Xlinenumannotationposition@side}{before}{%
823     \l@wrap@ifnotemptybox{\Xwraplinenumannotation@side}{%
824       \csuse{annot@the\absline@num @the\section@num}%
825     }%
826   }{%
827     \linenumrep{\line@num}%
828     \ifsublines@
829       \ifnum\subline@num>0\relax
830         \unskip%
831         \Xsublinesep@side%
832         \sublinenumrep{\subline@num}%
833       \fi
834     \fi%
835     \ifdefstring{\Xlinenumannotationposition@side}{after}{%
836       \l@wrap@ifnotemptybox{\Xwraplinenumannotation@side}{%

```

```

837 \csuse{annot@the\absline@num @the\section@num}%
838 }%
839 }{}%
840 \egroup%
841 }%
842
843 \newcommand*{\leftlinenum}{%
844 \ledlinenum
845 \kern\linenumsep}
846 \newcommand*{\rightlinenum}{%
847 \kern\linenumsep
848 \ledlinenum}
849
850 %

```

## 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`.

```

851 \newcount\line@num
852 %

```

**`\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.

```

853 \newcount\subline@num
854 %

```

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

**`\sublines@true`**  
**`\sublines@false`** You may wonder why we 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.

```
855 \newif\ifsublines@
856 %
```

**\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.

```
857 \newcount\absline@num
858 %
```

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.

```
859 \newcount\@lock
860 \newcount\sub@lock
861 %
```

## V.9 Line number associated to lemma

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

- **\line@list**: the page and line numbers for every lemma marked by `\edtext`. There are seven pieces of information, separated by vertical bars:
  1. the starting page,
  2. line, and
  3. sub-line numbers, followed by the
  4. ending page,
  5. line, and
  6. sub-line numbers, and then the
  7. font specifier for the lemma.

These line numbers are all visible numbers. The font specifier is a set of four codes for font encoding, family, series, and shape, separated by / characters. Thus a lemma that started on page 23, line 35 and went on until page 24, line 3 (with no sub-line numbering), and was typeset in a normal roman font would have a line list entry like this:

```
23|35|0|24|3|0|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.

- `\annot@list` stores line number annotations for the start and the end lines of each `\edtext`, as `\line@list` does for line numbers. We don't store that in the same list as line numbers as `\Xprintlines` already have height arguments, and a `TEX` macro can take at most nine arguments.

Here are the commands to create these lists:

```

862 \list@create{\line@list}
863 \list@create{\insertlines@list}
864 \list@create{\actionlines@list}
865 \list@create{\actions@list}
866 \list@create{\annot@list}%
867 %

```

```

\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
868 \newcount\page@num
869 \newcount\endpage@num
870 \newcount\endline@num
871 \newcount\endsubline@num
872 %

```

`\this@section@page@num` The `\this@section@page@num` stores the page number on which a numbering section ends.

```
873 \newcount\this@section@page@num%
874 %
```

`\ifnoteschanged@`  
`\noteschanged@true`  
`\noteschanged@false`

If the number of the footnotes in a section is different from what it was during the last run, or if this is the very first time you've run  $\text{\LaTeX}$ , on this file, the information from the line-list used to place the notes will be wrong, and some notes will probably be misplaced. When this happens, we prefer to give a single error message for the whole section rather than messages at every point where we notice the problem, because we do not really know where in the section notes were added or removed, and the solution in any case is simply to run  $\text{\LaTeX}$  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.

```
875 \newif\ifnoteschanged@
876 %
```

`\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@77 \newcommand*\resetprevline@{%
878     \def\do##1{\global\csundef{prevline##1}}%
879     \dolistloop{\@series}%
880 }
881 %
```

`\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@82 \newcommand*\resetprevpage@num{%
883     \def\do##1{%
884         \ifcsdef{prevpage##1@num}{%
885             \global\csname prevpage##1@num\endcsname=\z@%
886             \global\csname prevpage##1@numR\endcsname=\z@%
887         }%
888         {}%
889     \ifcsdef{##1prevpage@num}{%
890         \global\csname ##1prevpage@num\endcsname=\z@%
891         \global\csname ##1prevpage@numR\endcsname=\z@%
892     }%
893     {}%
```

```

894 }%
895 \dolistloop{\@series}%
896 }
897 %

```

## 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.

```

898 \newread\@inputcheck
899 \newcommand*\read@linelist}[1]{%
900   \ifledRcol%
901     \list@clearing@regR%
902   \else%
903     \list@clearing@reg%
904   \fi%
905 %

```

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).

```

906   \list@clear{\maxlinesinpar@list}
907 %

```

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 `\resumenumbers`, those things should still have the values they had when `\pausenumbering` was executed.

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

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

```

908 \get@linelistfile{#1}%
909 \@stopmsd%Security if last \endms{} is forgotten
910 \unless\ifledRcol%Get the last line of the last page
911   \csnumgdef{\lastabsline@forpage@the\page@num}{\the\absline@num}%
912   \csnumgdef{\lastline@forpage@the\page@num}{\the\line@num}%
913   \csnumgdef{\lastsubline@forpage@the\page@num}{\the\subline@num}%
914 \else%
915   \csnumgdef{\lastabsline@forpageR@the\page@numR}{\the\absline@numR}%

```

```

916 \csnumgdef{@lastline@forpageR@the\page@numR}{\the\line@numR}%
917 \csnumgdef{@lastsubline@forpageR@the\page@numR}{\the\subline@numR}%
918 \fi%
919 \endgroup
920 %

```

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.

```

921 \ifledRcol
922 \global\page@numR=\m@ne
923 \ifx\actionlines@listR\empty
924 \gdef\next@actionlineR{1000000}%
925 \else
926 \glp\actionlines@listR\to\next@actionlineR
927 \glp\actions@listR\to\next@actionR
928 \fi
929 \else
930 \global\page@num=\m@ne
931 \ifx\actionlines@list\empty
932 \gdef\next@actionline{1000000}%
933 \else
934 \glp\actionlines@list\to\next@actionline
935 \glp\actions@list\to\next@action
936 \fi
937 \fi
938 }
939 %

```

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

```

940 \newcommand*{\list@clearing@reg}{%
941 \list@clear{\line@list}%
942 \list@clear{\insertlines@list}%
943 \list@clear{\actionlines@list}%
944 \list@clear{\actions@list}%
945 \list@clear{\linesinpar@listL}%
946 \list@clear{\linesonpage@listL}%
947 }%
948 %

```

`\get@linelistfile` `reledmac` can take advantage of the L<sup>A</sup>T<sub>E</sub>X ‘safe file input’ macros to get the line-list file.

```

949 \newcommand*{\get@linelistfile}[1]{%
950 \InputIfFileExists{\l@auxdir#1}{%
951 \global\noteschanged@false
952 \begingroup

```

```

953 \catcode`\[=1 \catcode`\]=2
954 \makeatletter \catcode`\^M=9}{%
955 \led@warn@NoFile{\l@auxdir#1}%
956 \global\noteschanged@true
957 \begingroup}%
958 }
959
960 %

```

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 `\pausenumbers` and `\resumenumbers` macros to help you if you run into macro memory limitations (see 5.2.7 p. 21 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 refer 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.

```

961 \newcommand{\line@list@version}[1]{%
962 \IfStrEq{#1}{\this@line@list@version}%
963 {}%
964 {\ifledRcol%
965 \led@warn@Obsolete{\jobname.\extensionchars\the\section@num}%
966 \else%
967 \led@warn@Obsolete{\jobname.\extensionchars\the\section@num}%
968 \fi%
969 \endinput%
970 }%

```

```

971 }%
972 %

```

`\@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 `\@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`.

```

973
974 \newcommand*{\@nl}[2]{%
975   \@page{#1}%
976   \ifledRcol%
977     \@nl@regR%
978   \else%
979     \@nl@reg%
980   \fi%
981 }
982 \newcommand*{\@nl@reg}{%
983   \ifx\l@dchset@num\relax \else
984     \advance\absline@num \@ne
985     \csgdef{l@dchset@num@the\absline@num}{}%To remember this line have
been marked by a \setlinenum
986     \set@line@action
987     \let\l@dchset@num=\relax
988     \advance\absline@num \m@ne
989     \advance\line@num \m@ne
990   \fi
991 %

```

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

```

992 \reset@current@annot%
993 \advance\absline@num \@ne
994   \ifx\next@page@num\relax \else
995     \page@action
996     \let\next@page@num=\relax
997   \fi
998   \ifx\sub@change\relax \else
999     \ifnum\sub@change>\z@

```

```

1000         \sublines@true
1001     \else
1002         \sublines@false
1003     \fi
1004     \sub@action
1005     \let\sub@change=\relax
1006 \fi
1007 %

```

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

```

1008     \ifcase\@lock
1009     \or
1010         \@lock \tw@
1011     \or \or
1012         \@lock \z@
1013     \fi
1014     \ifcase\sub@lock
1015     \or
1016         \sub@lock \tw@
1017     \or \or
1018         \sub@lock \z@
1019     \fi
1020 %

```

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

```

1021     \ifsublines@
1022         \ifnum\sub@lock<\tw@
1023             \advance\subline@num \@ne
1024         \fi
1025     \else
1026         \ifnum\@lock<\tw@
1027             \advance\line@num \@ne \subline@num \z@
1028         \fi
1029     \fi}
1030 %
1031 %

```

`\last@page@num`    `\@page` determines whether or not a new page has been started, based on the page values held by `\@nl`.

```

\this@section@last@page@num
\@page
1032 \newcount\last@page@num
1033 \last@page@num=-10000
1034
1035 \newcount\this@section@last@page@num%
1036 \this@section@last@page@num=-10000%
1037
1038 \newcommand*{\@page}[1]{%
1039     \ifledRcol
1040         \ifnum #1=\last@page@numR

```

```

1041 \else
1042 \csnumgdef{\lastabsline@forpage@the\page@numR}{\the\absline@numR}%
1043 \csnumgdef{\lastline@forpage@the\page@numR}{\the\line@numR}%
1044 \csnumgdef{\lastsubline@forpage@the\page@numR}{\the\subline@numR}%
1045 \ifbypage@R
1046 \ifx\l@dchset@num\relax%Not resetting if preceded by a \setlinenum
1047 \ifbool{expr{%
1048 bool{resumenumberingR@start}%
1049 and test {\ifnumequal{\last@page@numR}{-10000}}}%
1050 }%
1051 {}%
1052 {%
1053 \line@numR \z@%
1054 \subline@numR \z@%
1055 \global\csdef{reset@lineR\the\numexpr\absline@numR+1\relax @\
the\section@numR}{}%
1056 }%
1057 \fi%
1058 \fi
1059 \page@numR=#1%
1060 \global\this@section@page@numR=#1%
1061 \last@page@numR=#1%
1062 \global\this@section@last@page@numR=#1%
1063 \def\next@page@numR{#1}%
1064 \gdef\this@section@next@page@numR{#1}%
1065 \fi
1066 \else
1067 \ifnum #1=\last@page@num
1068 \else
1069 \csnumgdef{\lastabsline@forpage@the\page@num}{\the\absline@num}%
1070 \csnumgdef{\lastline@forpage@the\page@num}{\the\line@num}%
1071 \csnumgdef{\lastsubline@forpage@the\page@num}{\the\subline@num}%
1072 \ifbypage@
1073 \ifx\l@dchset@num\relax%Not resetting if preceded by a \setlinenum
1074 \line@num \z@%
1075 \subline@num \z@%
1076 \global\csdef{reset@line\the\numexpr\absline@num+1\relax @the\
section@num}{}%
1077 \fi%
1078 \fi
1079 \page@num=#1%
1080 \global\this@section@page@num=#1%
1081 \last@page@num=#1%
1082 \global\this@section@last@page@num=#1%
1083 \def\next@page@num{#1}%
1084 %

```

And we set a flag that tells \@nl that a new page number is to be set, because other associated actions shouldn't occur until the next line-start occurs.

```

1085 \gdef\this@section@next@page@num{#1}%

```



```

1086 \listxadd{\normal@page@break}{\the\absline@num}
1087 \fi
1088 \fi}
1089 %

```

**\@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**

```

1090 \newcommand*{\@pend}[1]{}
1091 \newcommand*{\@pendR}[1]{}
1092 \newcommand*{\@lopL}[1]{}
1093 \newcommand*{\@lopR}[1]{}
1094
1095 %

```

**\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.

```

1096 \newcommand*{\sub@on}{\ifsublines@
1097 \let\sub@change=\relax
1098 \else
1099 \def\sub@change{1}%
1100 \fi}
1101 \newcommand*{\sub@off}{\ifsublines@
1102 \def\sub@change{-1}%
1103 \else
1104 \let\sub@change=\relax
1105 \fi}
1106
1107 %

```

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

```

1108
1109 \newcommand*{\@adv}[1]{%
1110 \ifsublines@
1111 \ifledRcol
1112 \advance\subline@numR by #1\relax
1113 \ifnum\subline@numR<\z@
1114 \led@warn@BadAdvancelineSubline
1115 \subline@numR \z@
1116 \fi
1117 \else
1118 \advance\subline@num by #1\relax
1119 \ifnum\subline@num<\z@
1120 \led@warn@BadAdvancelineSubline
1121 \subline@num \z@

```

```

1122     \fi
1123     \fi
1124     \else
1125     \ifledRcol
1126     \advance\line@numR by #1\relax
1127     \ifnum\line@numR<\z@
1128     \led@warn@BadAdvancelineLine
1129     \line@numR \z@
1130     \fi
1131     \else
1132     \advance\line@num by #1\relax
1133     \ifnum\line@num<\z@
1134     \led@warn@BadAdvancelineLine
1135     \line@num \z@
1136     \fi
1137     \fi
1138     \fi
1139     \set@line@action}
1140
1141 %

```

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

```

1142
1143 \newcommand*{\@set}[1]{%
1144     \ifledRcol
1145     \ifsublines@
1146     \subline@numR=#1\relax
1147     \else
1148     \line@numR=#1\relax
1149     \fi
1150     \set@line@action
1151     \else
1152     \ifsublines@
1153     \subline@num=#1\relax
1154     \else
1155     \line@num=#1\relax
1156     \fi
1157     \set@line@action
1158     \fi}
1159
1160 %

```

**\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.

```

1161

```

```

1162 \newcommand*{\l@d@set}[1]{%
1163   \ifledRcol
1164     \line@numR=#1\relax
1165     \advance\line@numR \@ne
1166     \def\l@dchset@num{#1}
1167   \else
1168     \line@num=#1\relax
1169     \advance\line@num \@ne
1170     \def\l@dchset@num{#1}
1171   \fi}
1172 \let\l@dchset@num\relax
1173
1174 %

```

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

```

1175
1176 \newcommand*{\page@action}{%
1177   \ifledRcol
1178     \xright@appenditem{\the\absline@numR}\to\actionlines@listR
1179     \xright@appenditem{\next@page@numR}\to\actions@listR
1180   \else
1181     \xright@appenditem{\the\absline@num}\to\actionlines@list
1182     \xright@appenditem{\next@page@num}\to\actions@list
1183   \fi}
1184 %

```

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

```

1185
1186 \newcommand*{\set@line@action}{%
1187   \ifledRcol
1188     \xright@appenditem{\the\absline@numR}\to\actionlines@listR
1189     \ifsublines@
1190       \@l@tempcnta=-\subline@numR
1191     \else
1192       \@l@tempcnta=-\line@numR
1193     \fi
1194     \advance\@l@tempcnta by -5000\relax
1195     \xright@appenditem{\the\@l@tempcnta}\to\actions@listR
1196   \else
1197     \xright@appenditem{\the\absline@num}\to\actionlines@list
1198     \ifsublines@
1199       \@l@tempcnta=-\subline@num
1200     \else
1201       \@l@tempcnta=-\line@num
1202     \fi
1203     \advance\@l@tempcnta by -5000\relax
1204     \xright@appenditem{\the\@l@tempcnta}\to\actions@list

```

```

1205 \fi}
1206 %

```

**\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.

```

1207
1208 \newcommand*\sub@action{%
1209   \ifledRcol
1210     \xright@appenditem{\the\absline@numR}\to\actionlines@listR
1211     \ifsublines@
1212       \xright@appenditem{-1001}\to\actions@listR
1213     \else
1214       \xright@appenditem{-1002}\to\actions@listR
1215     \fi
1216   \else
1217     \xright@appenditem{\the\absline@num}\to\actionlines@list
1218     \ifsublines@
1219       \xright@appenditem{-1001}\to\actions@list
1220     \else
1221       \xright@appenditem{-1002}\to\actions@list
1222     \fi
1223   \fi}
1224 %

```

**\lock@on** \lock@on adds an entry to the action-code list to turn line number locking on. The current setting of the sub-lineation flag tells us whether this applies to line numbers or sub-line numbers.

Adding commands to the action list is slow, and it is very often the case that a lock-on command is immediately followed by a lock-off command in the line-list file, and therefore really does nothing. We use a look-ahead scheme here to detect such pairs, and add nothing to the line-list in those cases.

```

1225 \newcommand*\lock@on{\futurelet\next\do@lockon}
1226
1227 \newcommand*\do@lockon{%
1228   \ifx\next\lock@off
1229     \global\let\lock@off=\skip@lockoff
1230   \else
1231     \ifledRcol
1232       \do@lockonR
1233     \else
1234       \do@lockonL
1235     \fi
1236   \fi}
1237
1238
1239 \newcommand*\do@lockonL{%
1240   \xright@appenditem{\the\absline@num}\to\actionlines@list

```

```

1241 \ifsublines@
1242 \xright@appenditem{-1005}\to\actions@list
1243 \ifnum\sub@lock=\z@
1244 \sub@lock \@ne
1245 \else
1246 \ifnum\sub@lock=\thr@@
1247 \sub@lock \@ne
1248 \fi
1249 \fi
1250 \else
1251 \xright@appenditem{-1003}\to\actions@list
1252 \ifnum\@lock=\z@
1253 \@lock \@ne
1254 \else
1255 \ifnum\@lock=\thr@@
1256 \@lock \@ne
1257 \fi
1258 \fi
1259 \fi}
1260 %
1261 %

```

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

```

\do@lockoff
\do@lockoffL
\skip@lockoff
1262 \newcommand*{\do@lockoffL}{%
1263 \xright@appenditem{\the\absline@num}\to\actionlines@list
1264 \ifsublines@
1265 \xright@appenditem{-1006}\to\actions@list
1266 \ifnum\sub@lock=\tw@
1267 \sub@lock \thr@@
1268 \else
1269 \sub@lock \z@
1270 \fi
1271 \else
1272 \xright@appenditem{-1004}\to\actions@list
1273 \ifnum\@lock=\tw@
1274 \@lock \thr@@
1275 \else
1276 \@lock \z@
1277 \fi
1278 \fi}
1279
1280 \newcommand*{\do@lockoff}{%
1281 \reset@current@annot%
1282 \ifledRcol
1283 \do@lockoffR
1284 \else
1285 \do@lockoffL
1286 \fi}
1287 \newcommand*{\skip@lockoff}{\global\let\lock@off=\do@lockoff}

```

```
1288 \global\let\lock@off=\do@lockoff
```

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

```

1291 \newcommand*{\n@num}{%
1292   \ifledRcol%
1293     \xright@appenditem{\the\absline@numR}\to\actionlines@listR
1294     \xright@appenditem{-1007}\to\actions@listR
1295   \else%
1296     \xright@appenditem{\the\absline@num}\to\actionlines@list%
1297     \xright@appenditem{-1007}\to\actions@list%
1298   \fi%
1299 }%
1300
1301 %

```

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

```

1302 \newcommand*{\n@num@stanza}{%
1303   \ifledRcol%
1304     \xright@appenditem{\the\absline@numR}\to\actionlines@listR%
1305     \xright@appenditem{-1008}\to\actions@listR%
1306   \else%
1307     \xright@appenditem{\the\absline@num}\to\actiononlines@list%%
1308     \xright@appenditem{-1008}\to\actions@list%
1309   \fi%
1310 }
1311 %

```

```

\ifl@dhdidnumber \hidenumbering hides number in margin. It uses action code 1009. \hidenumberingonleftpage
\hidenumbering and \hidenumberingonrightpage are variant, using action code only conditionnaly
\hidenumberingonleftpage
\hidenumberingonrightpage
\newif\ifl@dhdidnumber

```

```

1313 \newcommand*{\hidenumbering}{%
1314   \ifledRcol%
1315     \write\linenum@outR{\string\hide@num}%
1316   \else%
1317     \write\linenum@out{\string\hide@num}%
1318   \fi%
1319 }%
1320 \newcommand*{\hide@num}{%
1321   \ifledRcol%
1322     \xright@appenditem{\the\absline@numR}\to\actionlines@listR%
1323     \xright@appenditem{-1009}\to\actions@listR%
1324   \else%

```

```

1325 \xright@appenditem{\the\absline@num}\to\actionlines@list%%
1326 \xright@appenditem{-1009}\to\actions@list%
1327 \fi%
1328 }
1329 \newcommand*{\hidenumberingonleftpage}{%
1330 \ifledRcol%
1331 \write\linenum@outR{\string\hide@num@left}%
1332 \else%
1333 \write\linenum@out{\string\hide@num@left}%
1334 \fi%
1335 }%
1336
1337 \newcommand*{\hide@num@left}{%
1338 \ifledRcol%
1339 \ifodd\page@numR\else%
1340 \xright@appenditem{\the\absline@numR}\to\actionlines@listR%
1341 \xright@appenditem{-1009}\to\actions@listR%
1342 \fi%
1343 \else%
1344 \ifodd\page@num\else%
1345 \xright@appenditem{\the\absline@num}\to\actionlines@list%%
1346 \xright@appenditem{-1009}\to\actions@list%
1347 \fi%
1348 \fi%
1349 }%
1350
1351 \newcommand*{\hidenumberingonrightpage}{%
1352 \ifledRcol%
1353 \write\linenum@outR{\string\hide@num@right}%
1354 \else%
1355 \write\linenum@out{\string\hide@num@right}%
1356 \fi%
1357 }%
1358
1359 \newcommand*{\hide@num@right}{%
1360 \ifledRcol%
1361 \ifodd\page@numR%
1362 \xright@appenditem{\the\absline@numR}\to\actionlines@listR%
1363 \xright@appenditem{-1009}\to\actions@listR%
1364 \fi%
1365 \else%
1366 \ifodd\page@num%
1367 \xright@appenditem{\the\absline@num}\to\actionlines@list%%
1368 \xright@appenditem{-1009}\to\actions@list%
1369 \fi%
1370 \fi%
1371 }%
1372
1373 %

```

`\@ref` `\@ref` marks the start of a passage, for creation of a footnote reference. It takes two arguments:

- #1, the number of entries to add to `\insertlines@list` for this reference. This value, here and within `\edtext`, which computes it and writes it to the line-list file, will be stored in the count `\insert@count`.

```
1374 \newcount\insert@count
1375 %
```

- #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.

```
1376 \newcommand*\@dummy@ref}[2]{#2}
1377 %
```

`\@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.

```
1378 \newcommand*\@ref}[2]{%
1379   \ifledRcol%
1380     \@ref@regR{#1}{#2}%
1381   \else%
1382     \@ref@reg{#1}{#2}%
1383   \fi%
1384 }%
1385 \newcommand*\@ref@reg}[2]{%
1386   \global\insert@count=#1\relax
1387   \global\advance\@edtext@level by 1%
1388   \loop\ifnum\insert@count>\z@
1389     \xright@appenditem{\the\absline@num}\to\insertlines@list
1390     \global\advance\insert@count \m@ne
1391   \repeat
1392 %
```

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.

```
1393 \begingroup
1394   \let\@ref=\@dummy@ref
1395   \let\@lopL\@gobble
1396   \let\page@action=\relax
1397   \let\sub@action=\relax
1398   \let\set@line@action=\relax
1399   \let\@lab=\relax
```



```

1400 \let\@lemma=\relax%
1401 \let\@sw\@gobblethree%
1402 \let\store@annot@to@absline\@gobble%
1403 #2
1404 \global\endpage@num=\page@num
1405 \global\endline@num=\line@num
1406 \global\endsubline@num=\subline@num
1407 \global\let\endcurrent@annot=\current@annot%
1408 \endgroup
1409 %

```

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

```

1410 \xright@appenditem%
1411   {\the\page@num|\the\line@num|%
1412    \ifsublines@ \the\subline@num \else 0\fi|%
1413    \the\endpage@num|\the\endline@num|%
1414    \ifsublines@ \the\endsubline@num \else 0\fi}\to\line@list
1415 \xright@appenditem%
1416   {\current@annot|\endcurrent@annot}\to\annot@list%
1417 %

```

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

```

1418 \@ref@reg@parse{#2}%
1419 %

```

Decrease edtext level counter.

```

1420 \global\advance\@edtext@level by -1%
1421 }
1422 %

```

**`\@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.

```

1423 \newcommand{\@ref@reg@parse}[1]{%
1424   \expandafter\list@create\expandafter{\csname sw@list@edtext@tmp@\the\
1425   @edtext@level\endcsname}%
1426   \providebool{lemmacommand@\the\@edtext@level}%
1427   \boolfalse{lemmacommand@\the\@edtext@level}%
1428 }
1429 %

```

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

```

1428 #1%
1429 %

```

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.

```

1430 \ifnum\@edtext@level>0%
1431 \def\create@this@edtext@level{\expandafter\list@create\expandafter{\
csname sw@list@edtext@\the\@edtext@level\endcsname}}%
1432 \ifcsundef{sw@list@edtext@\the\@edtext@level}{\create@this@edtext@level
}%
1433 \letcs{\@tmp}{sw@list@edtext@\the\@edtext@level}%
1434 \letcs{\@tmpp}{sw@list@edtext@tmp@\the\@edtext@level}
1435 \xright@appenditem{\expandonce\@tmpp}\to\@tmp%
1436 \global\cslet{sw@list@edtext@\the\@edtext@level}{\@tmp}%
1437 \fi%
1438 %

1439 }
1440
1441 %

```

**\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.

```

1442 \newcommand{\@ref@later}[1]{%
1443 \global\advance\@edtext@level by \@ne%
1444 \ifledRcol%
1445 \@ref@reg@parseR{#1}%
1446 \else%
1447 \@ref@reg@parse{#1}%
1448 \fi%
1449 \global\advance\@edtext@level by -\@ne%
1450 }%
1451 %

```

## 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`.

```

1452 \newwrite\linenum@out
1453 %

```

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

**\first@linenum@out@true**

**\first@linenum@out@false**

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.

```
1454 \newif\iffirst@linenum@out@
1455 \first@linenum@out@true
1456 %
```

`\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.

```
1457 \newcommand{\this@line@list@version}{8}%
1458 %
```

`\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.

`\next@line@list@stuff`

```
1459 \let\next@line@list@stuff\relax%
1460 \newcommand*{\line@list@stuff}[1]{%
1461 %
```

First, define a toggle set to true when we are not in the first run.

```
1462 \global\newtoggle{notfirststrun@#1}%
1463 \IfFileExists{\l@auxdir#1}%
1464 {\global\toggletrue{notfirststrun@#1}}%
1465 {\global\togglefalse{notfirststrun@#1}}%
1466 %
```

A internal hook (not used yet).

```
1467 \next@line@list@stuff%
1468 \global\let\next@line@list@stuff\relax%
1469 %
```

Use the commands of the previous section to interpret the line-list file from the last run.

```
1470 \read@linelist{#1}%
1471 %
```

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. `reledmac` and `reledpar` can fill the `\next@line@list@stuff` hook between a `\endnumbering` (associated with numbered file  $n$ ) and a `\beginnumbering` (associated with numbered file  $n + 1$ ). It allows adding content to the numbered file  $n + 1$  and not  $n$ .

```

1472 \iffirst@linenum@out@
1473   \global\first@linenum@out@false%
1474   \immediate\openout\linenum@out=\l@auxdir#1\relax%
1475   \immediate\write\linenum@out{\string\line@list@version{\
this@line@list@version}}}%
1476   \ifl@dpaging%
1477     \immediate\write\linenum@out{\string\@par@sync@option{\
@par@this@sync@option}}}%
1478   \fi%
1479 \else
1480 %

```

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.

```

1481   \if@minipage%
1482     \leavevmode%
1483     \fi%
1484     \closeout\linenum@out%
1485     \openout\linenum@out=\l@auxdir#1\relax%
1486     \write\linenum@out{\string\line@list@version{\this@line@list@version}}
%
1487     \ifl@dpaging%
1488       \write\linenum@out{\string\@par@sync@option{\@par@this@sync@option}}
%
1489     \fi%
1490   \fi%
1491 }%
1492
1493 %

```

`\new@line` The `\new@line` macro sends the `\@n1` command to the line-list file, to mark the start of a new text line, and its page number. It writes the two forms of the page number:

- Raw form (`\the\c@page`, which can be used for numeric tests).
- Formatted form (for example, in Roman).

Actually, only the first form is used by `\@n1`. If we use the `\sameparallelpagenunder` option of `reledpar`, we must write not the real page number (i.e. page counter, defined in standard  $\text{\LaTeX}$ ) but the printed page number (i.e. `par@page` counter, defined only on `reledmac`).

- For the raw form, we use `\the\c@page@page` macro, because the `{par@page}` counter is increased for each page.
- For the formatted version, `\thepage` is patched through `\par@patch@thepage`. So we have nothing to change.

```

1494 \newcommand*{\new@line}{%
1495   \ifnumberline%
1496     \IfStrEq{\led@pb@setting}{after}%
1497       {\xifinlist{\the\absline@num}{\l@prev@nopb}%
1498         {\xifinlist{\the\absline@num}{\normal@page@break}%
1499           {\numgdef{\@next@page}{\c@par@page+\@ne}%
1500             \write\linenum@out{\string\@nl[\@next@page][\@next@page]}}%
1501           }%
1502         {\write\linenum@out{\string\@nl[\the\@this@c@page][\thepage]}}}%
1503       }%
1504     {\write\linenum@out{\string\@nl[\the\@this@c@page][\thepage]}}}%
1505   {}%
1506   \IfStrEq{\led@pb@setting}{before}%
1507     {\numdef{\next@absline}{\the\absline@num+\@ne}%
1508     \xifinlist{\next@absline}{\l@prev@nopb}%
1509       {\xifinlist{\the\absline@num}{\normal@page@break}%
1510         {\numgdef{\nc@page}{\c@par@page+\@ne}%
1511           \write\linenum@out{\string\@nl[\nc@page][\nc@page]}}%
1512         }%
1513       {\write\linenum@out{\string\@nl[\the\@this@c@page][\thepage]}}}%
1514     }%
1515     {\write\linenum@out{\string\@nl[\the\@this@c@page][\thepage]}}}%
1516   }%
1517   {}%
1518   \IfStrEqCase{\led@pb@setting}%
1519     {%
1520       {before}{\relax}%
1521       {after}{\relax}%
1522     }[%
1523       \write\linenum@out{\string\@nl[\the\@this@c@page][\thepage]}}%
1524   ]%
1525   \fi%
1526 }
1527
1528 \newcount\@this@c@page%%
1529 \newcommand{\set@this@c@page}{%
1530   \ifboolexpr{%
1531     bool{sameparallelpagenumber}%
1532     or bool{prevpgnotnumbered}%
1533   }%
1534   {\global\@this@c@page=\c@par@page}%
1535   {\global\@this@c@page=\c@page}%
1536 }%
1537 %

```

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

**\flag@start** We enclose a lemma marked by \edtext in \flag@start and \flag@end: these send  
**\flag@end** the \@ref command to the line-list file. \edtext is responsible for setting the value of

`\insert@count` appropriately; it actually gets done by the various footnote macros.

```

1538 \newif\if@noneed@Footnote%
1539
1540 \newcommand*{\flag@start}{%
1541   \ifledRcol%
1542     \edef\next{\write\linenum@outR{%
1543       \string\@ref[\the\insert@countR] []}}%
1544     \next%
1545     \ifnum\insert@countR<1%
1546       \if@noneed@Footnote\else%
1547         \led@err@EdtextWithoutFootnote%
1548       \fi%
1549     \fi%
1550   \else%
1551     \edef\next{\write\linenum@out{%
1552       \string\@ref[\the\insert@count] []}}%
1553     \next%
1554     \ifnum\insert@count<1%
1555       \if@noneed@Footnote\else%
1556         \led@err@EdtextWithoutFootnote%
1557       \fi%
1558     \fi%
1559   \fi}%
1560
1561 \newcommand*{\flag@end}{%
1562   \ifledRcol%
1563     \write\linenum@outR{}}%
1564   \else%
1565     \write\linenum@out{}}%
1566   \fi}%
1567
1568
1569 %

```

`\flag@start@RTL` With  $\text{\LaTeX}$ , 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>28</sup>

`\flag@end@RTL`

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

<sup>28</sup>This problem is caused by the way  $\text{\LaTeX}$  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.

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.

```

1570 \newcommand{\flag@start@RTL}{%
1571   \edlabel{edtext:start:\csuse{thisedtext@the\@edtext@level}}}%
1572   \IfStrEq{\xabslineref{edtext:start:\csuse{thisedtext@the\@edtext@level}}}%
1573     {000}%
1574     {}%
1575     {%
1576     \ifnumequal%
1577       {\xabslineref{edtext:start:\csuse{thisedtext@the\@edtext@level}}}%
1578       {\xabslineref{edtext:end:\csuse{thisedtext@the\@edtext@level}}}%
1579       {\flag@end}%
1580       {\flag@start}%
1581     }%
1582   }%
1583
1584 \newcommand{\flag@end@RTL}{%
1585   \edlabel{edtext:end:\csuse{thisedtext@the\@edtext@level}}}%
1586   \IfStrEq{\xabslineref{edtext:start:\csuse{thisedtext@the\@edtext@level}}}%
1587     {000}%
1588     {}%
1589     {%
1590     \ifnumequal%
1591       {\xabslineref{edtext:start:\csuse{thisedtext@the\@edtext@level}}}%
1592       {\xabslineref{edtext:end:\csuse{thisedtext@the\@edtext@level}}}%
1593       {\flag@start}%
1594       {\flag@end}%
1595     }%
1596   }%
1597   %

```

`\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`

```

1598 \newcommand*{\flag@start@later}{%
1599   \ifledRcol%
1600     \write\linenum@outR{\string\@ref@later []}%
1601   \else%
1602     \write\linenum@out{\string\@ref@later []}%
1603   \fi%

```

```

1604 }%
1605 \newcommand{\flag@end@later}{%
1606   \iflabeledRcol%
1607     \write\linenum@outR{}}%
1608   \else%
1609     \write\linenum@out{}}%
1610   \fi%
1611 }
1612 %

```

**\startsub** **\endsub** `\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.

```

1613
1614
1615 \newcommand*{\startsub}{\dimen0\lastskip
1616   \ifdim\dimen0>Opt \unskip \fi
1617   \iflabeledRcol \write\linenum@outR{\string\sub@on}%
1618   \else \write\linenum@out{\string\sub@on}%
1619   \fi
1620   \ifdim\dimen0>Opt \hskip\dimen0 \fi}
1621 \def\endsub{\dimen0\lastskip
1622   \ifdim\dimen0>Opt \unskip \fi
1623   \iflabeledRcol \write\linenum@outR{\string\sub@off}%
1624   \else \write\linenum@out{\string\sub@off}%
1625   \fi
1626   \ifdim\dimen0>Opt \hskip\dimen0 \fi}
1627
1628 %

```

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

```

1629 \newcommand*{\advanceline}[1]{\leavevmode%
1630   \iflabeledRcol \write\linenum@outR{\string\@adv[#1]}%
1631   \else \write\linenum@out{\string\@adv[#1]}%
1632   \fi%
1633 }
1634 %

```



**\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.

```

1635
1636 \newcommand*{\setline}[1]{%
1637 \leavevmode%
1638 \ifnum#1<\z@
1639 \led@warn@BadSetline%
1640 \else%
1641 \ifledRcol%
1642 \write\linenum@outR{\string\@set[#1]}%
1643 \else%
1644 \write\linenum@out{\string\@set[#1]}%
1645 \fi%
1646 \fi%
1647 }%
1648
1649 %

```

**\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.

```

1650
1651 \newcommand*{\setlinenum}[1]{%
1652 \ifnum#1<\z@%
1653 \led@warn@BadSetlinenum%
1654 \else%
1655 \ifledRcol%
1656 \write\linenum@outR{\string\l@d@set[#1]}%
1657 \else%
1658 \write\linenum@out{\string\l@d@set[#1]}%
1659 \fi%
1660 \fi%
1661 }%
1662
1663 %

```

**\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.

```

1664
1665 \newcommand*{\startlock}{%
1666 \ifledRcol \write\linenum@outR{\string\lock@on}%
1667 \else \write\linenum@out{\string\lock@on}%
1668 \fi}
1669 \def\endlock{%
1670 \ifledRcol \write\linenum@outR{\string\lock@off}%
1671 \else \write\linenum@out{\string\lock@off}%
1672 \fi}
1673 %

```

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

```

\l@dskipnumbertrue
\l@dskipnumberfalse
\skipnumbering
1674 \newif\ifl@dskipnumber
1675 \newif\ifl@dskipversenumber%
1676 \newcommand*{\skipnumbering}{%
1677   \leavevmode%
1678   \ifledRcol%
1679     \ifinstanza%
1680       \write\linenum@outR{\string\n@num@stanza}%
1681     \else%
1682       \write\linenum@outR{\string\n@num}%
1683     \fi%
1684     \advanceline{-1}%
1685   \else%
1686     \ifinstanza%
1687       \write\linenum@out{\string\n@num@stanza}%
1688     \else%
1689       \write\linenum@out{\string\n@num}%
1690     \fi%
1691     \advanceline{-1}%
1692   \fi%
1693 }%
1694
1695 %

```

## 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. 162). 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.

```
1696 \list@create{\end@lemmas}
1697 %
```

`\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.

```
1698 \newcommand{\dummy@edtext}[2]{#1}
1699 %
```

`\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`.

```
1700 \newcommand{\dummy@edtext@showlemma}[2]{\showlemma{#1}}%
1701 %
```

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` We need to turn off macro expansion for certain sorts of macros we are likely to see  
`\morenoexpands` 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>29</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 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

<sup>29</sup>Since ‘control sequences equivalent to characters are not expandable’—*The TeXbook*, answer to Exercise 20.14.

macro adds any spaces to the text when it runs, extra space will appear in the main text when `\edtext` is used.

The `\newseries` command also adds `\let\footnote⟨X⟩\@gobble` to the end of the `\noexpands` macro for the series `⟨X⟩`.

(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<sub>TEX</sub> or Xe<sub>TEX</sub>.)

```

1702 \newcommand*{\noexpands}{%
1703   \let\select@lemfont=0%
1704   \let\startsub=\relax \let\endsub=\relax
1705   \let\startlock=\relax \let\endlock=\relax
1706   \let\edlabel=\@gobble
1707   \let\setline=\@gobble \let\advanceline=\@gobble
1708   \let\sameword\samewordinedtext%
1709   \let\edtext=\dummyedtext
1710   \let\edindex\dummyedindex%
1711   \l@dtabnoexpands
1712   \l@noexpandsedgl%
1713   \let\linenumannotation=\@gobble%
1714   \morenoexpands}
1715 \let\morenoexpands=\relax
1716
1717 %

```

`\@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.

```

1718 \newcommand{\@tag}{}
1719 %

```

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

```

1720 \newcount\@edtext@level%
1721 \@edtext@level=0%
1722 %

```

`\ifedtext@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.

```

1723 \newif\ifedtext@secondarg@%
1724 %

```

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

```
1725 \newcounter{edtext}
1726 \renewcommand{\theedtext}{edtxt@\the\c@edtext}%
1727 %
```

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

```
1728 \newcommand{\edtext}[2]{\leavevmode%
1729 %
```

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

```
1730 \ifnumberedpar{}%
1731 %
```

Check the content of the arguments, to be certain there is no forbidden command inside.

```
1732 \@check@edtext@args{#1}{#2}%
1733 %
```

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

```
1734 \global\advance\@edtext@level by 1%
1735 %
```

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  $\XeLaTeX$ .

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`.

```
1736 \stepcounter{edtext}%
1737 \csxdef{thisedtext@\the\@edtext@level}{\theedtext}%
1738 %
```

By default, we do not use `\lemma`

```
1739 \global\@lemmacommand@false%
1740 %
```

```
1741 \begingroup%
1742 %
```

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`.

```
1743 \ifledRcol%
1744 \ifcvoid{sw@list@edtextR@\the\@edtext@level}%
1745 { \global\let\sw@inthisedtext\empty}%
1746 { \expandafter\gl@p\cswname sw@list@edtextR@\the\
@edtext@level\endcsname\to\sw@inthisedtext}%
1747 \else%
1748 \ifcvoid{sw@list@edtext@\the\@edtext@level}%
```

```

1749         {\global\let\sw@inthisedtext\empty}%
1750         {\expandafter\glp\cswname sw@list@edtext@the\@edtext@level
\endcsname\to\sw@inthisedtext}%
1751     \fi%
1752 %

```

`\@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.

```

1753     \global\renewcommand{\@tag}{%
1754         \no@expands #1%
1755     }%
1756 %

```

`\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`.

```

1757     \set@line%
1758 %

```

`\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`.

```

1759     \ifledRcol \global\insert@countR \z@%
1760     \else      \global\insert@count \z@ \fi%
1761 %

```

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.

```

1762     \@edtext@secondarg@true%
1763     \ignorespaces #2\relax%
1764     \@edtext@secondarg@false%
1765 %

```

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.

```

1766     \if@RTL%
1767         \flag@start@RTL%
1768     \else%

```

```

1769     \flag@start%
1770     \fi%
1771 %

```

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

```

1772     \if@lemmacommand%
1773     \ifledRcol%
1774     \write\linenum@outR{\string\@lemma}%
1775     \else%
1776     \write\linenum@out{\string\@lemma}%
1777     \fi%
1778     \fi%
1779 %

```

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.

```

1780     \endgroup%
1781     \ifdef{\hypertarget}%
1782     {%
1783     \Hy@raisedlink@left{\hypertarget{\csuse{thisedtext@the\
@edtext@level}:start}{}}%
1784     \showlemma{#1}%
1785     \Hy@raisedlink{\hypertarget{\csuse{thisedtext@the\@edtext@level}:
end}{}}%
1786     }%
1787     {%
1788     \showlemma{#1}%
1789     }%
1790 %

```

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.

```

1791     \ifx\end@lemmas\empty \else%
1792     \gl@p\end@lemmas\to\x@lemma%
1793     \x@lemma%
1794     \global\let\x@lemma=\relax%
1795     \fi%
1796     \if@RTL%
1797     \flag@end@RTL%
1798     \else%
1799     \flag@end%
1800     \fi%
1801 %

```



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`.

```
1802 \global\@noneed@Footnotefalse%
1803 \global\advance\@edtext@level by -1%
1804 \global\@lemmacommand@false%
1805 %
```

We also reset `\@beforeinsertofthisedtext`

```
1806 \global\let\@beforeinsertofthisedtext\relax%
1807 %
```

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

```
1808 \else%
1809 \showlemma{#1} (\textbf{\textsc{Edtext outside numbered paragraph}})\
1810 \led@err@edtextoutsidestart%
1811 \fi%
1812 }%
1813 %
1814 %
```

`\@check@edtext@args` A macro which just checks the arguments of the `\edtext` and let know if there is some probleme, like, for example, `\pstart` inside.

```
1815 \newcommand{\@check@edtext@args}[2]{%
1816 \begingroup%
1817 \noexpandarg%
1818 \IfSubStr{#1}{\pstart}{\led@err@PstartInEdtext{first}}{}%
1819 \IfSubStr{#1}{\pend}{\led@err@PendInEdtext{first}}{}%
1820 \IfSubStr{#2}{\pstart}{\led@err@PstartInEdtext{second}}{}%
1821 \IfSubStr{#2}{\pend}{\led@err@PendInEdtext{second}}{}%
1822 \endgroup%
1823 }%
1824 %
1825 %
```

`\@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`.

```
1826 \let\@beforeinsertofthisedtext\relax
1827 %
```

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

```
1828 \newif\ifnumberline
1829 \numberlinetrue
1830 %
```

`\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.

```
1831 \newcommand*{\set@line}{%
1832   \ifl@edRcol
1833     \ifx\line@list\empty
1834       \global\noteschanged@true
1835       \xdef\l@d@nums{000|000|000|000|000|000|000|\edfont@info}%
1836     \else
1837       \gl@p\line@list\to\@tempb
1838       \xdef\l@d@nums{\@tempb|\edfont@info}%
1839       \global\let\@tempb=undefined
1840     \fi
1841     \ifx\annot@list\empty%
1842       \xdef\l@current@annotR{||}%
1843     \else%
1844       \gl@p\annot@list\to\@tempb%
1845       \xdef\l@current@annotR{\@tempb}%
1846       \global\let\@tempb=undefined%
1847     \fi%
1848   \else
1849     \ifx\line@list\empty
1850       \global\noteschanged@true
1851       \xdef\l@d@nums{000|000|000|000|000|000|000|\edfont@info}%
1852     \else
1853       \gl@p\line@list\to\@tempb
1854       \xdef\l@d@nums{\@tempb|\edfont@info}%
1855       \global\let\@tempb=undefined
1856     \fi
1857     \ifx\annot@list\empty%
1858       \xdef\l@current@annotR{||}%
1859     \else%
1860       \gl@p\annot@list\to\@tempb%
1861       \xdef\l@current@annotR{\@tempb}%
1862       \global\let\@tempb=undefined%
1863     \fi%
1864   \fi}
```

```
1865 %
1866 %
```

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

```
1867 \newcommand*\edfont@info{\f@encoding/\f@family/\f@series/\f@shape}
1868 %
1869 %
```

## 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. 143).

```
1870 \newcommand*\lemma[1]{%
1871   \global\@lemmacommand@true%
1872   \global\renewcommand{\@tag}{%
1873     \no@expands #1%
1874   }%
1875   \ignorespaces%
1876 }%
1877 %
```

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

```
1878 \newcommand{\@lemma}{%
1879   \booltrue{lemmacommand@the\@edtext@level}%
1880 }%
1881 %
```

**\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.

```
1882 \newif\if@lemmacommand@%
1883 %
```

## 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. 111): 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.

```
1884 \newcommand*{\linenum}[1]{%
1885   \xdef\@tempa{#1|}|}|}|}|noexpand\\l@d@nums}%
1886   \global\let\l@d@nums=\empty
1887   \expandafter\line@set\@tempa|\\ignorespaces}
1888 %
```

`\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.

```
1889 \def\line@set#1|#2\\#3|#4\\{%
1890   \gdef\@tempb{#1}%
1891   \ifx\@tempb\empty
1892     \l@d@add{#3}%
1893   \else
1894     \l@d@add{#1}%
1895   \fi
1896   \gdef\@tempb{#4}%
1897   \ifx\@tempb\empty\else
1898     \l@d@add{|}\line@set#2\\#4\\%
1899   \fi}
1900 %
```

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

```
1901 \newcommand{\l@d@add}[1]{\xdef\l@d@nums{\l@d@nums#1}}
1902
1903 %
```

`\lineannot` Last but not least, `\lineannot` allows us to substitute line number annotation. It is different from `\linenum` for backward compatibility with older versions of `reledmac`. It calls `\lineannot@set` to determine whether we must change only one annotation or two, or none.

`\lineannot@set`

```
1904 \newcommand*{\lineannot}[1]{%
1905   \lineannot@set#1|}%
1906 }%
1907 \def\lineannot@set#1|#2|{%
1908   \expandafter\parse@annot#1|#2|%
1909   \IfStrEq{#1}{-}%
1910     {\let\@tempa\annot@start}%
1911     {\def\@tempa{#1}}%
1912   \IfStrEq{#2}{-}%
1913     {\let\@tempb\annot@start}%
1914 }
```

```

1914 {\def\@tempb{#2}}%
1915 \xdef\l@current@annot{\@tempa|\@tempb}%
1916 }%
1917 %

```

## 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. First, getting a sanitized form of the argument of `\sameword`

```
\get@sw@txt 18 \newcommand{\get@sw@txt}[1]{%
1919 \begingroup%
1920 \swnoexpands%
1921 %.
```

Using case sensibility option.

```
1922 \ifsw@caseinsensitive%
1923 \def\tmpa##1{\lowercase{##1}}%
1924 \else%
1925 \def\tmpa##1{##1}%
1926 \fi%
1927 %
```

And now, define `\sw@txt`.

```
1928 \tmpa{\protected@xdef\sw@txt{#1}}%
1929 \endgroup%
1930 }%
1931 %
```

Allow some macros inside `\sameword`. We use `\RenewExpandableDocumentCommand` to get expandable command with optional argument. Cf. <https://tex.stackexchange.com/a/384783/7712>.

```
\sw@noexpand 32 \newcommand{\swnoexpands}{%
1933 \let\sameword\l@secondmandarg% Allow to have nested \sameword
1934 \let\emph\@firstofone%
1935 \let\textit\@firstofone%
1936 \let\textbf\@firstofone%
1937 \let\textsc\@firstofone%
1938 \let\framebox\@firstofone%
1939 \let\edtext\dummy@edtext%
1940 \RenewExpandableDocumentCommand{\edindex}{om}{}%
1941 \ifdefined\index%
1942 \RenewExpandableDocumentCommand{\index}{om}{}%
1943 \fi%
1944 \let\selectlanguage\@gobble%
```

```

1945 \let\foreignlanguage\@secondoftwo%
1946 \ifdefined\xpg@loaded%
1947 \renewcommand\do[1]{%
1948 \expandafter\RenewExpandableDocumentCommand\expandafter{\csname
text##1\endcsname}{om}{####2}%
1949 }%
1950 \expandafter\docsvlist\expandafter{\xpg@loaded}%
1951 \fi%
1952 }%
1953 %

```

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

```

1954 \newcommandx\sameword}[2][1,usedefault]{%
1955 \ifxetex%
1956 \if@RTL%
1957 \led@err@samewordRTL%
1958 \fi%
1959 \fi%
1960 \leavevmode%
1961 \get@sw@txt{#2}%
1962 %

```

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

```

1963 \unless\ifledRcol%
1964 \csnumgdef{sw@\sw@txt}{\csuse{sw@\sw@txt}+\@ne}%
1965 %

```

Then, write its value to the numbered file.

```

1966 \protected@write\linenum@out{\string\@sw{\sw@txt}{\csuse{sw@\sw@txt}
}}{#1}}%
1967 %

```

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

```

1968 \else%
1969 \csnumgdef{sw@\sw@txt}{\csuse{sw@\sw@txt}+\@ne}%
1970 \protected@write\linenum@outR{\string\@sw{\sw@txt}{\csuse{sw@\sw@txt}
}}{#1}}%
1971 \fi%
1972 %

```

And print the word.

```

1973 #2%
1974 }%
1975 %

```

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



```
\if@addsw76 \newif\if@addsw%
1977 %
```

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

```
1978 \newcommand{\@sw}[3]{%
1979   \get@sw@txt{#1}%
1980   \unless\ifledRcol%
1981 %
```

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

```
1982   \csxdef{sw@\sw@txt @\the\absline@num @\the\section@num}{#2}%
1983 %
```

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

```
1984   \numdef{\prev@line}{\the\absline@num-1}%
1985   \ifcsundef{sw@\sw@txt @\prev@line @\the\section@num}{%
1986     \csnumgdef{sw@\sw@txt @\prev@line @\the\section@num}{#2-1}%
1987   }{%
1988 %
```

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

```
1989   \numdef{\the@sw}{#2-\csuse{sw@\sw@txt @\prev@line @\the\section@num}}%
1990 %
```

And do the same thing for the right side.

```
1991 \else%
1992   \csxdef{sw@\sw@txt @\the\absline@numR @\the\section@numR @R}{#2}%
1993   \numdef{\prev@line}{\the\absline@numR-1}%
1994   \ifcsundef{sw@\sw@txt @\prev@line @\the\section@numR @R}{%
1995     \csnumgdef{sw@\sw@txt @\prev@line @\the\section@numR @R}{#2-1}%
1996   }{%
1997   \numdef{\the@sw}{#2-\csuse{sw@\sw@txt @\prev@line @\the\section@numR @R}}%
1998 \fi%
1999 %
```

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

```
2000 \@tempcnta=\@edtext@level
2001 \@whilenum{\@tempcnta>0}\do{%
2002   \ifcsdef{sw@list@edtext@tmp@\the\@tempcnta}%
2003   {%
2004     \@addswfalse%
2005     \notbool{lemmacommand@\the\@tempcnta}%
2006     {\@addswtrue}%
2007     {\IfStrEq{#3}{inlemma}%
2008       {\@addswtrue}%
2009     }%
```

```

2010         \def\do##1{%
2011             \ifnumequal{##1}{\the\@tempcnta}%
2012             {\@addswtrue\listbreak}%
2013             {}%
2014         }%
2015         \docsvlist{#3}%
2016     }%
2017 }%
2018 \if@addsw%
2019     \letcs{\@tmp}{sw@list@edtext@tmp@\the\@tempcnta}%
2020     \ifledRcol%
2021         \xright@appenditem{{\the@sw}{\the\absline@numR}}\to\@tmp%
2022     \else%
2023         \xright@appenditem{{\the@sw}{\the\absline@num}}\to\@tmp%
2024     \fi%
2025     \cslet{sw@list@edtext@tmp@\the\@tempcnta}{\@tmp}%
2026     \fi%
2027 }%
2028 {}%
2029 \advance\@tempcnta by -1%
2030 }%
2031 }%
2032 %

```

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

```

2033 \newcommandx{\sameword@inedtext}[2][1,usedefault]{%
2034     \get@sw@txt{#2}%
2035     \unless\ifledRcol%
2036 %

```

Just a precaution.

```

2037     \ifx\sw@list@inedtext\empty%
2038         \def\the@sw{999}%
2039         \def\this@absline{-99}%
2040     \else%
2041 %

```

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

```

2042     \gl@p\sw@list@inedtext\to\@tmp%
2043     \edef\the@sw{\expandafter\@firstoftwo\@tmp}%
2044     \edef\this@absline{\expandafter\@secondoftwo\@tmp}%
2045     \fi%
2046 %

```

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

```

2047     \ifcsdef{sw@\sw@txt @\this@absline @\the\section@num}{%
2048         \numdef{\prev@line}{\this@absline-1}%

```

```

2049 \numdef{\sw@atthisline}{\csuse{sw\sw@txt @\this@absline @\the\
section@num}-\csuse{sw\sw@txt @\prev@line @\the\section@num}}%
2050 }%
2051 {\numdef{\sw@atthisline}{0}}%
2052 %

```

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

```

2053 \ifnumgreater{\sw@atthisline}{1}%
2054 {\showwordrank{#2}{\the@sw}}%
2055 {#2}%
2056 %

```

And the same for right side.

```

2057 \else%
2058 \ifx\sw@list@inedtext\empty%
2059 \def\the@sw{999}%
2060 \def\this@absline{-99}%
2061 \else%
2062 \gl@p\sw@list@inedtext\to\@tmp%
2063 \edef\the@sw{\expandafter\@firstoftwo\@tmp}%
2064 \edef\this@absline{\expandafter\@secondoftwo\@tmp}%
2065 \fi%
2066 \ifcsdef{sw\sw@txt @\this@absline @\the\section@numR @R}{%
2067 \numdef{\prev@line}{\this@absline-1}%
2068 \numdef{\sw@atthisline}{\csuse{sw\sw@txt @\this@absline @\the\
section@numR @R}-\csuse{sw\sw@txt @\prev@line @\the\section@numR @R}}%
2069 }%
2070 {\numdef{\sw@atthisline}{0}}%
2071 \ifnumgreater{\sw@atthisline}{1}%
2072 {\showwordrank{#2}{\the@sw}}%
2073 {#2}%
2074 \fi%
2075 }%
2076 %

```

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

```

2077 \newcommand{\showwordrank}[2]{%
2078 #1\textsuperscript{#2}%
2079 }%
2080 %

```

## 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.

```
2081 \newbox\raw@text
2082 \newif\ifnumberedpar@
2083 \newcount\num@lines
2084 \newbox\one@line
2085 \newcount\par@line
2086 %
```

`\pstarts@typeset@L` The `\pstarts@typeset@L` counts the number of LEFT `\pstart` typeset. A left `\pstart` can be a `\pstart` typeset in normal mode, or `\pstart` typeset in left column or page in parallel mode (`withreledpar`). In the first case, its value is increased at every `\pstart` command. In the second case, its value is increased in `\Columns` or `\Pages` when we prepare to typeset a left `\pstart`. The `\pstarts@read@L` counts the number of LEFT `\pstart` read. A left `\pstart` can be a `\pstart` typeset in normal mode, or `\pstart` typeset in left column or page in parallel mode (`withreledpar`). In the first case, its value is increased at every `\pstart` command. In the second case, its value is increased at every `\pstarL`.

```
2087 \newcount\pstarts@typeset@L%
2088 \newcount\pstarts@read@L%
2089 %
```

`\pstart` `\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.

`\AtEveryPstart` `\pstart` needs to appear at the start of every paragraph that is to be numbered; the `\autopar` command below may be used to insert these commands automatically.

`\AtStartEveryPstart` 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.

`\numberpstarttrue`

`\numberpstartfalse`

`\labelpstarttrue`

`\labelpstartfalse`

`\thepstart`

`\ifat@every@pstart@star@`

```
2090 \newcommand{\AtStartEveryPstart}[1]{%
2091   \ifstrempy{#1}%
2092   {\gdef\@at@start@every@pstart{}}}%
2093
```

```

2094     {\gdef\@at@start@every@pstart{#1}}%
2095 }%
2096 \def\@at@start@every@pstart{}%
2097
2098 \newif\ifat@every@pstart@star%
2099 \newcommand{\AtEveryPstart}[1]{%
2100     \ifstrepty{#1}%
2101     {\gdef\at@every@pstart{}}%
2102     {\gdef\at@every@pstart{\noindent#1}}%
2103     \global\at@every@pstart@star@false%
2104 }%
2105 \WithSuffix\newcommand\AtEveryPstart*[1]{%
2106     \ifstrepty{#1}%
2107     {\gdef\at@every@pstart{}}%
2108     {\gdef\at@every@pstart{#1}}%
2109     \global\at@every@pstart@star@true%
2110 }%
2111 \def\at@every@pstart{}%
2112
2113 \newcounter{pstart}
2114 \renewcommand{\thepstart}{\bfseries\@arabic{c@pstart}. }
2115 \newif\ifnumberpstart
2116 \numberpstartfalse
2117 \newif\iflabelpstart
2118 \labelpstartfalse
2119 \newcommandx*\pstart}[2][1,2,usedefault]{%
2120     \normal@pars%
2121     \ifboolexpr{%
2122         test {\ifstrepty{#1}}%
2123         and test {\ifstrepty{#2}}%
2124     }%
2125     {\at@every@pstart}%
2126     {%
2127         \ifstrepty{#1}{\noindent#1}%
2128         \ifstrepty{#2}{#2}%
2129     }%
2130     \ifautopar%
2131         \autopar%
2132     \fi%
2133     \ifluatex%
2134         \edef\l@luatextextdir@L{\the\textdir}%
2135     \fi%
2136     \@nobraektrue%
2137     \ifnumbering \else%
2138         \led@err@PstartNotNumbered%
2139         \beginnumbering%
2140     \fi%
2141     \ifnumberedpar%
2142         \led@err@PstartInPstart%
2143         \pend%

```

```

2144 \fi%
2145 \list@clear{\inserts@list}%
2146 \global\let\next@insert=\empty%
2147 \begingroup\normal@pars%
2148 \global\advance \l@dnumstartL\@ne
2149 \global\advance \pstarts@typesetL\@ne%
2150 \global\advance \pstarts@readL\@ne%
2151 \global\setbox\raw@text=\vbox\bgroup%
2152 \if@nobreak%
2153   \if@afterindent\else%
2154     \noindent%
2155     \global\@afterindenttrue%
2156   \fi%
2157 \fi%
2158 \ifboolexpr{%
2159   bool{autopar}%
2160   and bool{by@autopar}%
2161 }%
2162 {}%
2163 {%
2164   \ifnumberpstart%
2165     \ifinstanza\else%
2166       \ifsidepstartnum\else%
2167         \thepstart%
2168       \fi%
2169     \fi%
2170   \fi%
2171 }%
2172 \numberedpar@true%
2173 \iflabelpstart%
2174   \protected@edef\@currentlabel{\p@pstart\thepstart}%
2175 \fi%
2176 \l@dzeropenalties%
2177 \@at@start@every@pstart%
2178 \global\by@autoparfalse%
2179 \ignorespaces%because not automatically ignored if an optional argument
    is used (classical TeX behavior for space after commands)
2180 }
2181 %

```

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

```

2182 \newcommand*{\pend}[2][1,2,usedefault]{\ifnumbering \else%
2183   \led@err@PendNotNumbered%
2184 \fi%
2185 \global\l@dskipversenumberfalse%
2186 \ifnumberedpar@ \else%
2187   \led@err@PendNoPstart%
2188 \fi%
2189 %

```

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.

```

2190 \l@dzeropenalties%
2191 \at@end@every@pend%
2192 \endgraf\global\num@lines=\prevgraf\egroup%
2193 \global\par@line=0%
2194 \get@firsthalf@antilabe@wd%
2195 %

```

Output the lines.

```

2196 \loop\ifvbox\raw@text%
2197   \do@line%
2198 \repeat%
2199 %

```

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

```

2200 \flush@notes%
2201 \endgroup%
2202 \ignorespaces%
2203 %

```

Increase `pstart` counter.

```

2204 \ifnumberpstart%
2205   \global\pstartnumtrue%
2206 \fi%
2207 \addtocounter{pstart}{1}%
2208 \ifcontinuousnumberingwithcolumns%
2209   \addtocounter{pstartL}{1}%
2210   \addtocounter{pstartR}{1}%
2211 \fi%
2212 %

```

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

```

2213 \normal@pars%
2214 \ifboolexpr{%
2215   test {\ifstrempy{#1}}%
2216   and test {\ifstrempy{#2}}%
2217 }%
2218   {\at@every@pend}%
2219   {%
2220     \ifstrempy{#1}{\noindent#1}%
2221     \ifstrempy{#2}{#2}%
2222   }%
2223 %

```

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.

```
2224 \@nobreakfalse%
2225 \ifautopar%
2226 \autopar%
2227 \fi%
2228 %
```

At the very end, we check if the lineation is made by `\pstart`. In this case, we reset the line number, using `\setlinenum`, as we are, technically, already out of `\pstart\pend` structure.

```
2229 \ifbypstart%
2230 \begingroup%
2231 \unless\ifinstanza%
2232 \let\leavevmode\relax%
2233 \setlinenum{0}%
2234 \fi%
2235 \endgroup%
2236 \resetprevline%
2237 \fi%
2238 }
2239 %
```

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.

```
\AtEveryPend40
\at@every@pend41
\ifat@every@pend@star@42 \newif\ifat@every@pend@star@%
2243 \newcommand{\AtEveryPend}[1]{%
2244 \ifstrempy{#1}%
2245 {\gdef\at@every@pend{}}%
2246 {\gdef\at@every@pend{\noindent#1}}%
2247 \global\at@every@pend@star@false%
2248 }%
2249 \WithSuffix\newcommand\AtEveryPend*[1]{%
2250 \ifstrempy{#1}%
2251 {\gdef\at@every@pend{}}%
2252 {\gdef\at@every@pend{#1}}%
2253 \global\at@every@pend@star@true%
2254 }%
2255 \xdef\at@every@pend{}%
2256
2257 %
```

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



```

2258 \newcommand{\AtEndEveryPend}[1]{%
2259   \ifstrempy{#1}%
2260     {\xdef\@at@end@every@pend{}}%
2261     {\gdef\@at@end@every@pend{#1}}%
2262 }%
2263 \def\@at@end@every@pend{}%
2264 %

```

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

```

2265 \newcommand*\l@dzeropenalties{%
2266   \brokenpenalty \z@ \clubpenalty \z@
2267   \displaywidowpenalty \z@ \interlinepenalty \z@ \predisplaypenalty \z@
2268   \postdisplaypenalty \z@ \widowpenalty \z@}
2269 %
2270 %

```

**\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 \beginnumbing 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.

The boolean \ifautopar is set to TRUE while \autopar is enabled, and the \ifby@autopar is set to TRUE at each \pstart automatically called by the \autopar feature. A manual \pstart will have a \ifby@autopar set to FALSE.

```

2271 \newif\ifautopar
2272 \newif\ifby@autopar%
2273 \newcommand*\autopar{%
2274   \ifledRcol

```

```

2275 \ifnumberingR \else
2276 \led@err@AutoparNotNumbered
2277 \beginnumberingR
2278 \fi
2279 \else
2280 \ifnumbering \else
2281 \led@err@AutoparNotNumbered
2282 \beginnumbering
2283 \fi
2284 \fi
2285 \autopartrue
2286 \everypar{\setbox0=\lastbox
2287 \endgraf \vskip-\parskip
2288 \global\by@autopartrue%
2289 \pstart \noindent \kern\wd0%
2290 \ifnumberpstart%
2291 \ifinstanza\else%
2292 \thepstart%
2293 \fi%
2294 \fi%
2295 \let\par=\pend}%
2296 \ignorespaces}
2297 %

```

**\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.

```

2298 \newcommand*{\normal@pars}{\everypar{}\let\par\endgraf}
2299
2300 %

```

**\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.

```

2301 \newif\ifautopar@pause
2302 %

```

## 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. The `\l@dunhbox@line` macro only `\unhboxes` one line, but packages like `\microtype` can override it as required.

```

2303 \newcommand*{\l@dunhbox@line}[1]{\unhbox #1}
2304 \newcommand*{\do@line}{%

```

```

2305 {\vbadness=10000
2306 \splittopskip=\z@
2307 \do@linehook
2308 \l@emptyd@ta
2309 \global\setbox\one@line=\vsplit\raw@text to\baselineskip}%
2310 \unvbox\one@line \global\setbox\one@line=\lastbox
2311 \getline@num
2312 \IfStrEq{\led@pb@setting}{before}{\led@check@pb\led@check@nopb}{\}
2313 \ifnum\@lock>\@ne
2314 \inserthangingsymboltrue
2315 \else
2316 \inserthangingsymbolfalse
2317 \fi
2318 \check@pb@in@verse
2319 \ifl@dhiddenumber%
2320 \global\l@dhiddenumberfalse%
2321 \f@x@l@cks%
2322 \else%
2323 \affixline@num%
2324 \fi%
2325 %

```

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

```

2326 \xifinlist{\the\pstarts@typeset@L}{\eled@sections@@}%
2327 {\print@eledsection}%
2328 {\print@line}%
2329 \IfStrEq{\led@pb@setting}{after}{\led@check@pb\led@check@nopb}{\}
2330 }%
2331 %

```

### VII.2.2 Process for “normal” line

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

```

2332 \def\print@line{
2333 %

```

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

```

2334 \affixpstart@num%
2335 %

```

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

```

2336 \hb@xt@ \linewidth{%
2337 %

```

User hooks.

```

2338 \do@insidelinehook%
2339 \csuse{insidethis@}\the\absline@num @\the\section@num}%

```

```

2340 \global\csundef{insidethis@the\absline@num @the\section@num}%
2341 %

```

Left line number

```

2342 \l@dld@ta%
2343 %

```

Prepare text to be inserted before notes.

```

2344 \if@firstlineofpage%
2345 \set@Xtxtbeforenotes%
2346 \set@txtbeforenotesX%
2347 \global\@firstlineofpagefalse%
2348 \fi%
2349 %

```

Insert footnotes made of manuscripts data and critical footnotes.

```

2350 \ifdefstring{\ms@data@position}{msdata-regular}{%
2351 \insert@msdata%
2352 \add@inserts%
2353 \add@Xgroupbyline%
2354 }{%
2355 \add@inserts%
2356 \add@Xgroupbyline%
2357 \insert@msdata%
2358 }%
2359 %

```

Insert marginal notes.

```

2360 \affixside@note%
2361 %

```

Print left notes.

```

2362 \l@dlsn@te
2363 %

```

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

```

2364 {\ledllfill\hb@xt@ \wd\one@line{%
2365 \new@line%
2366 %

```

If we use the continuousnumberingwithcolumns, we increase right line number.

```

2367 \continuousnumberingwithcolumns@sync@linenumber@singletext%
2368 %

```

If we use Lua $\TeX$  then restore the direction.

```

2369 \ifluatex%
2370 \textdir\l@luatextextdir@L%
2371 \fi%
2372 %

```

Insert, if needed, the hanging symbol.

```
2373 \inserthangingsymbol%
2374 %
```

And so, print the line.

```
2375 \l@dunhbox@line{\one@line}}%
2376 %
```

Right line number

```
2377 \ledrlfill\l@drd@ta%
2378 %
```

Print right notes.

```
2379 \l@drsn@te%
2380 }}%
2381 %
```

And reinsert penalties (for page breaking)...

```
2382 \add@penalties%
2383 }
2384 %
```

### 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.

```
2385 \def\print@eledsection{%
2386 \disable@edindex%
2387 \if@firstlineofpage%
2388 \set@Xtxtbeforenotes%
2389 \set@txtbeforenotesX%
2390 \global\@firstlineofpagefalse%
2391 \fi%
2392 \ifdefstring{\ms@data@position}{msdata-regular}{%
2393 \insert@msdata%
2394 \add@inserts%
2395 \add@Xgroupbyline%
2396 }{%
2397 \add@inserts%
2398 \add@Xgroupbyline%
2399 \insert@msdata%
2400 }%
2401 \affixside@note%
2402 \numdef{\temp@}{\pstarts@typeset@L-1}%
2403 \xifinlist{\temp@}{\eled@sections@}{\@nobreaktrue}{\@nobreakfalse}%

```

```

2404 \electedsectioningtrue%
2405 \csuse{eled@sectioning@the\pstarts@typeset@L}%
2406 \electedsectioningfalse%
2407 \global\csundef{eled@sectioning@the\pstarts@typeset@L}%
2408 \if@RTL%
2409   \hspace{-3\paperwidth}%
2410   {\hbox{\l@dunhbox@line{\one@line}} \new@line}%
2411 \else%
2412   \hspace{3\paperwidth}%
2413   {\new@line \hbox{\l@dunhbox@line{\one@line}}}%
2414 \fi%
2415 \vskip-\baselineskip%
2416 \continuousnumberingwithcolumns@sync@linenumber@singletext%
2417 \restore@edindex%
2418 }
2419 %

```

## VII.2.4 Hooks

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

```

2420 \newcommand*{\do@linehook}{ }
2421 \newcommand*{\do@insidelinehook}{ }
2422 %

```

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

```

2423 \newcommand*{\dolinehook}[1]{\gdef\do@linehook{#1}}%
2424 \newcommand*{\doinsidelinehook}[1]{\gdef\do@insidelinehook{#1}}%
2425 %
2426 %

```

**\doinsidethislinehook** The \doinsidethislinehook can be called directly in the main flood of a text. It will define a hook executed at a specific line, determined by the position of the command in the flood of the text.

```

2427 \newcommand{\doinsidethislinehook}[1]{%
2428   \leavevmode%In case it begins with a \pstart, ensure the \@insidethisline
   is written after \@nl
2429   \ifledRcol%
2430     \write\linenum@outR{\string\@insidethisline[\unexpanded{#1}]}%
2431   \else%
2432     \write\linenum@out{\string\@insidethisline[\unexpanded{#1}]}%
2433   \fi%
2434 }%
2435 %

```

The auxiliary files command just store the information to be executed when typesetting the specific line.

```

2436 \newcommand{\@insidethisline}[1]{%
2437   \ifledRcol%
2438     \csgappto{insidethisR@the\absline@numR @the\section@numR}{#1}%
2439   \else%
2440     \csgappto{insidethis@the\absline@num @the\section@num}{#1}%
2441   \fi%
2442 }%
2443 %

```

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

`\l@emptyd@ta` Nulls the `\. . .d@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`  
`\l@dcsnotetext@l`  
`\l@dcsnotetext@r`

```

2444 \newcommand*{\l@emptyd@ta}{%
2445   \gdef\l@dld@ta{}%
2446   \gdef\l@drd@ta{}%
2447   \gdef\l@dcsnotetext@l{}%
2448   \gdef\l@dcsnotetext@r{}%
2449   \gdef\l@dcsnotetext{}%
2450 %
2451 %

```

`\l@dlsn@te` Zero width boxes of the left and right sidenotes, together with their kerns, and, eventu-  
`\l@drsn@te` ally, with additional space if we are in parallel columns typesetting.

```

2452 \newcommand{\l@dlsn@te}{%
2453   \ifboolexpr{%
2454     bool {l@dprintingcolumns}%
2455     and bool {ledRcol@}%
2456   }{% If we are on a right column
2457     \@tempdima=\@morespace@leftnote@rightcolumn%
2458   }{%
2459     \@tempdima=\z@%
2460   }%
2461   \hb@xt@ \z@{\hss\box\l@dlp@rbox\kern\ledlsnotesep\hskip\@tempdima}%
2462 }%
2463 \newcommand{\l@drsn@te}{%
2464   \ifboolexpr{%
2465     bool {l@dprintingcolumns}%
2466     and not bool {ledRcol@}%
2467   }{% If we are on a left column
2468     \@tempdima=\@morespace@rightnote@leftcolumn%
2469   }{%
2470     \@tempdima=\z@%
2471   }%

```

```

2472 \hb@xt@ \z@{\hskip\@tempdima\kern\ledrsnotesep\box\l@drp@rbox\hss}%
2473 }%
2474
2475 %

```

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

**\ledrlfill**

```

2476 \newcommand*{\ledllfill}{\hfil}
2477 \newcommand*{\ledrlfill}{\hfil}
2478
2479 %

```

## 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.

```

2480 \newcommand*{\getline@num}{%
2481   \global\advance\absline@num \@ne%
2482   \do@actions
2483   \do@ballast
2484   \ifnumberline
2485     \ifsublines@
2486       \ifnum\sub@lock<\tw@
2487         \global\advance\subline@num \@ne
2488       \fi
2489     \else
2490       \ifnum\@lock<\tw@
2491         \global\advance\line@num \@ne
2492         \global\subline@num \z@
2493       \fi
2494     \fi
2495   \fi
2496 }
2497 %

```

**\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 XII.2 p. 182).

**\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**



```

2498 \newcount\ballast@count
2499 \newcounter{ballast}
2500 \setcounter{ballast}{0}
2501 %

```

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.

```

2502 \newcommand*{\do@ballast}{\global\ballast@count \z@
2503 \begingroup
2504 \advance\absline@num \@ne
2505 \ifnum\next@actionline=\absline@num
2506 \ifnum\next@action>-1001\relax
2507 \global\advance\ballast@count by -\c@ballast
2508 \fi
2509 \fi
2510 \endgroup}
2511 %

```

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

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

```

2512 \newcommand*{\do@actions}{%
2513 \global\let\do@actions@next=\relax
2514 \ifnum\absline@num<\next@actionline\else
2515 %

```

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.

```

2516 \ifnum\next@action>-1001
2517 \global\page@num=\next@action
2518 \ifresumenummering@start%
2519 \setbox0=\hbox{}%Required to get the correct page number, when the
\resumenummering is just after a \newpage
2520 \ifnum\pausenumbering@page@num<\page@num%
2521 \global\resumenummering@startfalse%
2522 \fi%
2523 \fi%
2524 \ifboolexpr{%
2525 bool{resumenummering@start}%
2526 and test {\ifdimgreater{\pagedepth}{\z@}}%
2527 }%
2528 {}%
2529 {\global\@firstlineofpagetrue}%
2530 \ifcsdef{reset@line\the\absline@num @\the\section@num}%

```

```

2531     {%
2532     \global\line@num=\z@%
2533     \global\subline@num=\z@%
2534     \resetprevline@%
2535     }%
2536     {%
2537     \global\resumenumbering@startfalse%
2538     \add@msdata@firstlineofpage%
2539 %

```

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`.)

```

2540 \else
2541 \ifnum\next@action<-4999
2542 \@l@dttempcnta=-\next@action
2543 \advance\@l@dttempcnta by -5001
2544 \ifsublines@
2545 \global\subline@num=\@l@dttempcnta
2546 \else
2547 \global\line@num=\@l@dttempcnta
2548 \fi
2549 %

```

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

```

2550 \else
2551 \@l@dttempcnta=-\next@action
2552 \advance\@l@dttempcnta by -1000
2553 \do@actions@fixedcode
2554 \fi
2555 \fi
2556 %

```

Now we get information about the next action off the list, and then set `\do@actions@next` so that we will call ourselves 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.

```

2557 \ifx\actionlines@list\empty
2558 \gdef\next@actionline{1000000}%
2559 \else
2560 \gl@p\actionlines@list\to\next@actionline
2561 \gl@p\actions@list\to\next@action
2562 \global\let\do@actions@next=\do@actions
2563 \fi
2564 \fi
2565 %

```

Make the recursive call, if necessary.

```

2566 \do@actions@next}

```

```
2567
2568 %
```

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

```
2569 \newcommand*{\do@actions@fixedcode}{%
2570   \ifcase\@l@dttempcnta
2571   \or%                               % 1001 = starting sublineation
2572     \global\sublines@true
2573   \or%                               % 1002 = ending sublineation
2574     \global\sublines@false
2575   \or%                               % 1003 = starting locking number
2576     \global\@lock=\@ne
2577   \or%                               % 1004 = ending locking number
2578     \ifnum\@lock=\tw@
2579       \global\@lock=\thr@@
2580     \else
2581       \global\@lock=\z@
2582     \fi
2583   \or%                               % 1005 = starting locking subnumber
2584     \global\sub@lock=\@ne
2585   \or%                               % 1006 = ending locking subnumber
2586     \ifnum\sub@lock=\tw@
2587       \global\sub@lock=\thr@@
2588     \else
2589       \global\sub@lock=\z@
2590     \fi
2591   \or%                               % 1007 = skipping numbering
2592     \l@dskipnumbertrue
2593   \or%                               % 1008 = skipping numbering in stanza
2594     \l@dskipversenumbertrue%
2595   \or%                               % 1009 = hiding number
2596     \l@dhidenumbertrue
2597   \or%                               % 1010 = inserting msdata
2598     \add@msdata%
2599   \else
2600     \led@warn@BadAction
2601   \fi}
2602
2603
2604 %
```

## VIII.1 Continuous line numbering between parallel typesetting and normal typesetting

`\nc@linenumber@singletext` The `continuousnumberingwithcolumns` option allows to alternate between single text and two parallel texts, keeping the same line numbers when switching from one layout to the other. The lines counter for the text in a single column and for the text in

the left column in parallel typesetting is the same. But the lines counter for the text in the same column is the not the same.

When typesetting single column text, if the option is enabled, we need to “simulate” typesetting right line, in order to keep the two counters synchronized. That is the aim of the

`\continuousnumberingwithcolumns@sync@linenumber@singletext` macro.

```

2605 \newcommand{\continuousnumberingwithcolumns@sync@linenumber@singletext}{%
2606   \ifcontinuousnumberingwithcolumns%
2607     \unless\ifafterendnumberingR%
2608       \new@lineR%
2609       \xappto\next@line@list@stuffR{%
2610         \unexpanded{\global\line@numR=}\the\line@num%
2611       }%
2612     \fi%
2613   \fi%
2614 }%
2615 %

```

## IX Line number annotation

The `\linenumannotation` allows users to add manual annotations to line numbers, for example to refer to line numbers from an older edition.

That implies that annotations be added

- in marginal line numbers. This feature is implemented by associating annotations with the absolute line numbers when reading the auxiliary numbered files (. 1, . 2 etc.).
- in critical footnotes. This feature is implemented by associating start / end line number annotations with each `\edtext` when reading the auxiliary numbered files.
- to crossref commands which refer to line numbers (to be completed when I will have resolved this issue).

`\linenumannotation` First, the user-level command, which only writes commands to the numbered files, storing the annotation.

```

2616 \newcommand{\linenumannotation}[1]{%
2617   \leavevmode%In case it begins with a \pstart, ensure the \@annot is
written after \@nl
2618   \iflabeledRcol%
2619     \write\linenum@outR{\string\@annot[#1]}%
2620   \else%
2621     \write\linenum@out{\string\@annot[#1]}%
2622   \fi%
2623 }%
2624 %

```

`\xlinenumannotationposition@side` The `\Xlinenumannotationposition@side` macro, to determine the position of line number annotations in lines printed in the side, relative to the line number position. The default value is “after”, but it can be changed to “before”, using the `\Xlinenumannotationposition` macro.

```
2625 \def\xlinenumannotationposition@side{after}%
2626 %
```

`\xwraplinenumannotation@ref` The `\Xwraplinenumannotation@ref` macro is applied to line annotations in crossref.  
`\xwraplinenumannotation@side` The `\Xwraplinenumannotation@side` macro is applied to line annotation in sides. They must be modified by users using `\Xwraplinenumannotation`. By default, they call `\textsuperscript`.

We also define toggle `Xnoidenticallinenumannotation@ref` and `Xnoidenticallinenumannotation@side`

```
2627 \def\xwraplinenumannotation@ref{\textsuperscript}%
2628 \def\xwraplinenumannotation@side{\textsuperscript}%
2629 \newtoggle{Xnoidenticallinenumannotation@ref}%
2630 \newtoggle{Xnoidenticallinenumannotation@side}%
2631 %
```

`\@annot` Then, the numbered files command `\@annot`, which

`\store@annot@to@absline`

- associates the annotation to the absolute lines number.
- stores the current annotation in a macro to be used by the `\@ref@reg` macro, which manages all things related to `\edtext` in numbered files. As we don’t want to store it multiple times in the event of nested `\edtext`, we use an auxiliary macro, `\store@annot@to@absline`, which is set to `\@gobble` when we parse nested `\edtext` in numbered auxiliary files.

```
2632 \newcommand{\@annot}[1]{%
2633   \store@annot@to@absline{#1}%
2634   \def\current@annot{#1}%
2635 }%
2636 \newcommand{\store@annot@to@absline}[1]{%
2637   \ifledRcol%
2638     \ifcsdef{annotR@the\absline@numR @the\section@numR}{%
2639       \csgappto{annotR@the\absline@numR @the\section@numR}{\
@linenumannotationsep#1}%
2640     }{%
2641       \csgdef{annotR@the\absline@numR @the\section@numR}{#1}%
2642     }%
2643   \else%
2644     \ifcsdef{annot@the\absline@num @the\section@num}{%
2645       \csgappto{annot@the\absline@num @the\section@num}{\
@linenumannotationsep#1}%
2646     }{%
2647       \csgdef{annot@the\absline@num @the\section@num}{#1}%
2648     }%

```

```

2649 \fi%
2650 }%
2651 %

```

**\current@annot** By default, there is no annotation to a line number, so we store an empty \current@annot

```

2652 \let\current@annot=\empty%
2653 %

```

**\reset@current@annot** \reset@current@annot is called at each \@n1 in numbered auxiliary files. It resets the annotation of line numbers at each line.

```

2654 \newcommand{\reset@current@annot}{%
2655 \unless\ifnoresetlinenumannotation@
2656 \let\current@annot=\empty%
2657 \fi
2658 }%
2659 %

```

**\parse@annot** In the \annot@list list, each individual \edtext is associated with a starting and an ending line number annotation, which is stored this way: start annotation|end annotation. The \parse@annot reads the format and defines two global macros.

```

2660 \def\parse@annot#1|#2|{%
2661 \gdef\annot@start{#1}%
2662 \gdef\annot@end{#2}%
2663 }%
2664 %

```

**\setlinenumannotationsep** The separator between the annotations and the way to redefine it.

```

\@linenumannotationsep
2665 \newcommand{\setlinenumannotationsep}[1]{\gdef\@linenumannotationsep{#1}}%
2666 \def\@linenumannotationsep{, }%
2667 %

```

## X 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 as required.

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

$$\begin{aligned}
 n &= \text{int}((\text{linenum} - \text{firstlinenum}) / \text{linenumincrement}) \\
 m &= \text{firstlinenum} + (n \times \text{linenumincrement})
 \end{aligned}$$

(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  $\backslash\text{line@num} \leq \backslash\text{firstlinenum}$ , we compare the two directly instead of making these calculations.

We compute, in the scratch counter  $\backslash\text{@l@tempcnta}$ , 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  $\backslash\text{@l@tempcntb}$  for comparison.

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

```
2668 \newcommand*{\affixline@num}{%
2669 %
```

No number is attached if  $\backslash\text{ifl@dskipnumber}$  is TRUE (and then it is set to its normal FALSE value). No number is attached if  $\backslash\text{ifnumberline}$  is FALSE (the normal value is TRUE).

```
2670 \ifledgroupnotesL@else
2671 \ifnumberline
2672 \ifl@dskipnumber
2673 \global\l@dskipnumberfalse
2674 \else
2675 \ifsublines@
2676 \l@tempcntb=\subline@num
2677 \ifnum\subline@num>\c@firstsublinenum
2678 \l@tempcnta=\subline@num
2679 \advance\l@tempcnta by-\c@firstsublinenum
2680 \divide\l@tempcnta by\c@sublinenumincrement
2681 \multiply\l@tempcnta by\c@sublinenumincrement
2682 \advance\l@tempcnta by\c@firstsublinenum
2683 \else
2684 \l@tempcnta=\c@firstsublinenum
2685 \fi
2686 %
```

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.

```
2687 \ch@cksub@l@ck
2688 %
```

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

```
2689 \else
2690 \l@tempcntb=\line@num
2691 %
```

Check on the  $\backslash\text{linenumberlist}$  If it is  $\backslash\text{empty}$  use the standard algorithm.

```
2692 \ifx\linenumberlist\empty
2693 \ifnum\line@num>\c@firstlinenum
2694 \l@tempcnta=\line@num
2695 \advance\l@tempcnta by-\c@firstlinenum
2696 \divide\l@tempcnta by\c@linenumincrement
2697 \multiply\l@tempcnta by\c@linenumincrement
```

```

2698         \advance\l@ldtempcnta by\c@firstlinenum
2699     \else
2700         \l@ldtempcnta=\c@firstlinenum
2701     \fi
2702 \else
2703 %

```

The `\linenumberlist` was not `\empty`, so here is Wayne's numbering mechanism. This takes place in  $\TeX$ 's mouth.

```

2704         \l@ldtempcnta=\line@num
2705         \edef\rem@inder{\linenumberlist,\number\line@num,}%
2706         \edef\sc@n@list{\def\noexpand\sc@n@list
2707             ###1,\number\l@ldtempcnta,###2|{\def\noexpand\rem@inder
2708 {####2}}}%
2709         \sc@n@list\expandafter\sc@n@list\rem@inder|
2710         \ifx\rem@inder\empty%
2711             \advance\l@ldtempcnta\@ne
2712         \fi
2713 %

```

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

```

2714         \ch@ck@l@ck
2715     \fi
2716 %

```

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

```

2717         \ifnum\l@ldtempcnta=\l@ldtempcntb
2718         \ifl@dskipversenumber\else
2719 %

```

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  $\LaTeX$  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`

```

2720         \if@twocolumn
2721         \if@firstcolumn
2722             \gdef\l@dld@ta{\llap{\leftlinenum}}%
2723         \else
2724             \gdef\l@drd@ta{\rlap{\rightlinenum}}%
2725         \fi

```



```

2726         \else
2727             \ifboolexpr{bool {l@printingcolumns} and test {\
ifnumgreater{\line@margin@columns}{\m@ne}}}%
2728                 {\@l@tempcntb=\line@margin@columns}%
2729                 {\@l@tempcntb=\line@margin}%
2730             \ifnum\@l@tempcntb>\@ne
2731                 \advance\@l@tempcntb \page@num
2732             \fi
2733             \ifboolexpr{%
2734                 bool {l@printingcolumns}%
2735                 and (%
2736                     (test {\ifdefstring{\linenum@OnlyPages@ForColumns}{left
}}%
2737                     and test {\ifnumodd{\page@num}}%
2738                     )%
2739                     or%
2740                     (test {\ifdefstring{\linenum@OnlyPages@ForColumns}{right
}}%
2741                     and not test {\ifnumodd{\page@num}}%
2742                     )%
2743                 )%
2744             }%
2745             {%
2746                 \ifodd\@l@tempcntb%
2747                     \gdef\l@drd@ta{\rlap{\rightlinenum}}}%
2748                 \else%
2749                     \gdef\l@dld@ta{\llap{\leftlinenum}}}%
2750                 \fi%
2751             }%
2752         \fi
2753     \fi
2754 \fi
2755 %
2756 %

```

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

```

2757     \f@x@l@cks
2758     \fi
2759     \fi
2760     \fi
2761 }
2762 %
2763 %

```

`\ch@cksub@l@ck` These macros handle line number locking for `\affixline@num`. `\ch@cksub@l@ck`  
`\ch@ck@l@ck` checks subline locking. If it fails, then we disable the line-number display by setting the  
`\f@x@l@cks` counters to arbitrary but unequal values.

```

2764 \newcommand*{\ch@cksub@l@ck}{%
2765   \ifcase\sub@lock
2766   \or
2767     \ifnum\sublock@disp=\@ne
2768       \@l@tempcntb=\z@ \@l@tempcnta=\@ne
2769     \fi
2770   \or
2771     \ifnum\sublock@disp=\tw@ \else
2772       \@l@tempcntb=\z@ \@l@tempcnta=\@ne
2773     \fi
2774   \or
2775     \ifnum\sublock@disp=\z@
2776       \@l@tempcntb=\z@ \@l@tempcnta=\@ne
2777     \fi
2778   \fi}
2779 %

```

Similarly for line numbers.

```

2780 \newcommand*{\ch@ckl@ck}{%
2781   \ifcase\@lock
2782   \or
2783     \ifnum\lock@disp=\@ne
2784       \@l@tempcntb=\z@ \@l@tempcnta=\@ne
2785     \fi
2786   \or
2787     \ifnum\lock@disp=\tw@ \else
2788       \@l@tempcntb=\z@ \@l@tempcnta=\@ne
2789     \fi
2790   \or
2791     \ifnum\lock@disp=\z@
2792       \@l@tempcntb=\z@ \@l@tempcnta=\@ne
2793     \fi
2794   \fi}
2795 %

```

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

```

2796 \newcommand*{\f@x@l@cks}{%
2797   \ifcase\@lock
2798   \or
2799     \global\@lock=\tw@
2800   \or \or
2801     \global\@lock=\z@
2802   \fi
2803   \ifcase\sub@lock
2804   \or
2805     \global\sub@lock=\tw@
2806   \or \or
2807     \global\sub@lock=\z@
2808   \fi}

```

```
2809
2810 %
```

## XI Pstart number printing inside

Inside, the printing of the pstart number runs like the printing of the line number. There are only a few differences:

- The pstarts counter is upgraded 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.

```
\leftpstartnum11
\rightpstartnum12 \newif\ifsidepstartnum
\ifsidepstartnum13 \newcommand*{\affixpstart@num}{%
2814   \ifsidepstartnum
2815     \if@twocolumn
2816       \if@firstcolumn
2817         \gdef\l@dld@ta{\llap{\leftpstartnum}}}%
2818       \else
2819         \gdef\l@drd@ta{\rlap{\rightpstartnum}}}%
2820       \fi
2821     \else
2822       \@l@tempcntb=\line@margin%
2823       \ifnum\@l@tempcntb>\@ne
2824         \advance\@l@tempcntb \page@num
2825       \fi
2826       \ifodd\@l@tempcntb
2827         \gdef\l@drd@ta{\rlap{\rightpstartnum}}}%
2828       \else
2829         \gdef\l@dld@ta{\llap{\leftpstartnum}}}%
2830       \fi
2831     \fi
2832   \fi
2833 }
2834 %
2835
2836
2837 \newif\ifpstartnum
2838 \pstartnumtrue
2839 \newcommand*{\leftpstartnum}{
2840   \ifpstartnum\thepstart
2841   \kern\linenumsep\fi
```

```

2842 \global\pstartnumfalse
2843 }
2844 \newcommand*{\rightpstartnum}{
2845   \ifpstartnum
2846     \kern\linenumsep
2847     \thepstart
2848     \fi
2849   \global\pstartnumfalse
2850 }
2851 %

```

## XII 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.

### XII.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.

```

2852 \list@create{\inserts@list}
2853 %

```

`\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`.

```

2854 \newcommand*{\add@inserts}{%
2855   \global\let\add@inserts@next=\relax
2856   %

```

If `\inserts@list` is empty, there are not any more notes or insertions for this paragraph, and we need not waste our time.

```

2857   \ifx\inserts@list\empty \else
2858   %

```

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.

```

2859 \ifx\next@insert\empty
2860 \ifx\insertlines@list\empty
2861 \global\noteschanged@true
2862 \gdef\next@insert{100000}%
2863 \else
2864 \gl@p\insertlines@list\to\next@insert
2865 \fi
2866 \fi
2867 %

```

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.

```

2868 \ifnum\next@insert=\absline@num
2869 \gl@p\inserts@list\to\@insert
2870 \@insert
2871 \global\let\@insert=\undefined
2872 \global\let\next@insert=\empty
2873 \global\let\add@inserts@next=\add@inserts
2874 \fi
2875 \fi
2876 %

```

Make the recursive call, if necessary.

```

2877 \add@inserts@next}
2878
2879 %

```

**`\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, in order to be inserted later in one block. This insertion in one block is made by `\add@Xgroupbyline`.

```

2880 \newcommand{\add@Xgroupbyline}{%
2881 \unless\ifnocritical@%
2882 \def\do##1{%Looping on the series
2883 \let\olddo\do%Save the old \do macro, that is this macro itself!
2884 \def\do####1{%Looping on the ##1@forinserting command
2885 \ifcsdef{##1@forinserting@####1}{%
2886 \X@beforeinsertion{##1}%
2887 \if@ledgroup%
2888 \global\setbox\@nameuse{mp##1footins}=\vbox%
2889 \else%
2890 \insert\csize ##1footins\endcsize%
2891 \fi%
2892 }%
2893 \Xsetparindent{##1}%

```

```

2894 \ifcsdef{Xhsize\csuse{series@display##1}@##1}%
2895   {\hsize \csuse{Xhsize\csuse{series@display##1}@##1}}%
2896   }%
2897 \if@ledgroup%
2898   \unvbox\@nameuse{mp##1footins}%
2899 \fi%
2900 \X@atbegininsertion{##1}%
2901 \ifcsstring{series@display##1}{normal}%
2902   {%
2903     \Xledsetnormalparstuff{##1}%
2904     \rule{z@}\splittopskip%
2905   }%
2906   {%
2907     \csuse{##1@forinserting@###1}%
2908     \strut\par%
2909   }%
2910   \global\csundef{##1@forinserting@###1}%
2911   }%
2912   {%
2913   }%
2914 \ifcsdef{##1@forinserting}{%
2915   \dolistcslloop{##1@forinserting}%
2916 }{%
2917   \global\csundef{##1@forinserting}%
2918   \let\do\olddo%Restore old do
2919 }%
2920 \dolistloop{\@series}%
2921 \fi%
2922 }%
2923
2924
2925 %

```

## XII.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@dttempcnta 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. 168). Finally, the penalty is checked to see that it does not go below  $-10000$ .

```

2926 \newcommand*{\add@penalties}{\@l@dttempcnta=\ballast@count
2927   \ifnum\num@lines>\@ne
2928     \global\advance\par@line \@ne

```

```

2929 \ifnum\par@line=\@ne
2930 \advance\@l@dttempcnta \clubpenalty
2931 \fi
2932 \@l@dttempcntb=\par@line \advance\@l@dttempcntb \@ne
2933 \ifnum\@l@dttempcntb=\num@lines
2934 \advance\@l@dttempcnta \widowpenalty
2935 \fi
2936 \ifnum\par@line<\num@lines
2937 \advance\@l@dttempcnta \interlinepenalty
2938 \fi
2939 \fi
2940 \ifnum\@l@dttempcnta=\z@
2941 \relax
2942 \else
2943 \ifnum\@l@dttempcnta>-10000
2944 \penalty\@l@dttempcnta
2945 \else
2946 \penalty -10000
2947 \fi
2948 \fi}
2949
2950 %

```

### XII.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. For the first run, we do not flush the notes, as that means all the notes will be added at the end of numbered section, and so, very far of the expected position

```

2951 \newcommand*{\flush@notes}{%
2952 \iftoggle{notfirstrun@jobname.\extensionchars\the\section@num}{%
2953 \@xloop%
2954 \ifx\inserts@list\empty \else%
2955 \gl@p\inserts@list\to\@insert%
2956 \@insert%
2957 \global\let\@insert=\undefined%
2958 \repeat%
2959 }{}%
2960 }%
2961
2962
2963 %

```

`\@xloop` `\@xloop` is a variant of the PLAIN  $\TeX$  `\loop` macro, useful when it's hard to construct a positive test using the  $\TeX$  `\if` commands—as in `\flush@notes` above. One 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  $\TeX$  `\loop` is used, too, so we could just call it `\loop`; but it seems preferable not to change the definitions of any of the standard macros.)

This variant of `\loop` was introduced by Alois Kabsch in *TUGboat* 8 (1987), pp. 184–5.

```
2964 \def\@xloop#1\repeat{%
2965   \def\body{#1\expandafter\body\fi}%
2966   \body}
2967
2968 %
```

## XII.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 `\Xtextbeforenotes` will be inserted.

```
2969 \newcommand{\set@Xtxtbeforenotes}{%
2970   \unless\ifnocritical@%
2971     \def\do##1{%
2972       \nottoggle{Xtxtbeforenotesonlyonce@##1}{%
2973         \global\togglefalse{Xtxtbeforenotes@##1@typeset}%
2974       }{%
2975       }%
2976       \dolistloop{\@series}%
2977     \fi%
2978   }%
2979 %
```

`\set@txtbeforenotesX` The `\set@txtbeforenotesX` does the same for the `\textbeforenotesX`.

```
2980 \newcommand{\set@txtbeforenotesX}{%
2981   \unless\ifnofamiliar@%
2982     \def\do##1{%
2983       \nottoggle{txtbeforenotesonlyonceX@##1}{%
2984         \global\togglefalse{txtbeforenotesX@##1@typeset}%
2985       }{%
2986       }%
2987       \dolistloop{\@series}%
2988     \fi%
2989   }%
2990 %
```

`\insert@Xtxtbeforenotes` `\insert@Xtxtbeforenotes{⟨series⟩}`, called when inserting a familiar footnote, 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.

`\insert@txtbeforenotesX` is the same for familiar footnotes.

```

2991 \newcommand{\insert@txtbeforenotes}[1]{%
2992 \nottoggle{Xtxtbeforenotes@#1@typeset}{%
2993 \global\toggletrue{Xtxtbeforenotes@#1@typeset}%
2994 \ifcsvoid{Xtxtbeforenotes@#1}{}%
2995 \ifcsstring{series@display#1}{paragraph}%
2996 {\noindent\csuse{Xtxtbeforenotes@#1}}%
2997 {\expandafter\insert\cename#1footins\endcename%
2998 \bgroup%
2999 \noindent%
3000 \ifcsdef{\csuse{series@display#1}@begin@insert}{%
3001 \csuse{\csuse{series@display#1}@begin@insert}{#1}%
3002 }{}%
3003 \strut\csuse{Xnotefontsize@#1}%
3004 \csuse{Xtxtbeforenotes@#1}%
3005 \egroup%
3006 }%
3007 }%
3008 }%
3009 {}%
3010 }%
3011
3012
3013 \newcommand{\insert@txtbeforenotesX}[1]{%
3014 \nottoggle{txtbeforenotesX@#1@typeset}{%
3015 \global\toggletrue{txtbeforenotesX@#1@typeset}%
3016 \ifcsvoid{txtbeforenotesX@#1}{}%
3017 \ifcsstring{series@displayX#1}{paragraph}%
3018 {\noindent\csuse{txtbeforenotesX@#1}}%
3019 {\expandafter\insert\cename footins#1\endcename%
3020 \bgroup%
3021 \noindent%
3022 \ifcsdef{\csuse{series@displayX#1}@begin@insert}{%
3023 \csuse{\csuse{series@displayX#1}@begin@insert}{#1}%
3024 }{}%
3025 \strut\csuse{notefontsizeX@#1}\csuse{txtbeforenotesX@#1}%
3026 \egroup%
3027 }%
3028 }%
3029 }%
3030 {}%
3031 }%
3032
3033
3034 %

```

## XIII 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.

### XIII.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@lemmfont` `\select@lemmfont` 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.

```

3035 \def\select@lemmfont#1|#2|#3|#4|#5|#6|#7|{\select@lemmfont#7|}
3036 \def\select@lemmfont#1/#2/#3/#4|{%
3037   {\fontencoding{#1}\fontfamily{#2}\fontseries{#3}\fontshape{#4}%
3038   \selectfont}
3039
3040 %

```

### XIII.2 Individual note options

`\footnoteoptions@` The `\footnoteoption@` [*side*] [*options*] [*value*] changes the value of on options of Xfootnote, to switch between true and false.

```

3041 \newcommand*\footnoteoptions@[3]{%
3042   \def\do##1{%
3043     \ifstrequal{#1}{L}{% On the left side
3044       \xright@appenditem{\noexpand\setkeys[mac]{#3footnoteoption}{\
3045         unexpanded{##1}}}{\to\inserts@list%
3046       \global\advance\insert@count \@ne% Increment the left insert
3047       counter.
3048     }%
3049     \xright@appenditem{\noexpand\setkeys[mac]{#3footnoteoption}{\
3050       unexpanded{##1}}}{\to\inserts@listR%
3051       \global\advance\insert@countR \@ne% Increment the right insert
3052       counter insert.
3053     }%
3054     \notblank{#2}{\docsvlist{#2}}}% Parsing all options
3055   }
3056 %

```

### XIII.3 Notes language

`\footnotelang@lua` `\footnotelang@lua` is called to remember the information about the direction of a lemma when Lua<sup>1</sup>TeX is used.

```

3055 \newcommand*{\footnotelang@lua}[1][1=L,usedefault]{%
3056   \ifstrequal{#1}{L}{%
3057     \xright@appenditem{\csxdef{footnote@luatextextdir}{\the\textdir}}\to\
inserts@list%Know the dir of lemma
3058     \global\advance\insert@count \@ne%
3059     \xright@appenditem{\csxdef{footnote@luatexpardir}{\the\pardir}}\to\
inserts@list%Know the dir of lemma
3060     \global\advance\insert@count \@ne%
3061   }%
3062   {%
3063     \xright@appenditem{\csxdef{footnote@luatextextdir}{\the\textdir}}\to\
inserts@listR%Know the dir of lemma
3064     \global\advance\insert@countR \@ne%
3065     \xright@appenditem{\csxdef{footnote@luatexpardir}{\the\pardir}}\to\
inserts@listR%Know the dir of lemma
3066     \global\advance\insert@countR \@ne%
3067   }%
3068 }
3069 %

```

`\footnotelang@poly` `\footnotelang@poly` is called to remember the information about the language of a lemma when polyglossia is used.

```

3070 \newcommand*{\footnotelang@poly}[1][1=L,usedefault]{%
3071   \ifstrequal{#1}{L}{%
3072     \if@RTL%
3073       \xright@appenditem{\csxdef{footnote@dir}{@RTLtrue}}\to\
inserts@list%Know the language used in the lemma
3074       \global\advance\insert@count \@ne%
3075     \else
3076       \xright@appenditem{\csxdef{footnote@dir}{@RTLfalse}}\to\
inserts@list%Know the language of lemma
3077       \global\advance\insert@count \@ne%
3078     \fi%
3079     \xright@appenditem{\csxdef{footnote@lang}{\expandonce\language}}\
to\inserts@list%Know the language of lemma
3080     \global\advance\insert@count \@ne%
3081   }%
3082   {%
3083     \if@RTL
3084       \xright@appenditem{\csxdef{footnote@dir}{@RTLtrue}}\to\
inserts@listR%Know the language of lemma
3085       \global\advance\insert@countR \@ne%
3086     \else
3087       \xright@appenditem{\csxdef{footnote@dir}{@RTLfalse}}\to\
inserts@listR%Know the language of lemma

```

```

3088 \global\advance\insert@countR \@ne%
3089 \fi
3090 \xright@appenditem{{\csxdef{footnote@lang}{\expandonce\language}}}\
to\inserts@listR%Know the language of lemma
3091 \global\advance\insert@countR \@ne%
3092 }%
3093 }
3094 %

```

### XIII.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”.

### XIII.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`.

```

3095 \newcommand*{\footsplitskips}{%
3096 \interlinepenalty=\interfootnotelinepenalty
3097 \unless\ifl@dprintingpages%
3098 \floatingpenalty=\@MM%
3099 \fi%

```

```

3100 \splittopskip=\ht\strutbox \splitmaxdepth=\dp\strutbox
3101 \leftskip=\z@skip \rightskip=\z@skip
3102
3103 %

```

**\normalfootnoterule** \normalfootnoterule is a standard footnote-rule macro, for use by a footstart macro: just the same as the PLAIN T<sub>E</sub>X footnote rule.

```

3104 \let\normalfootnoterule=\footnoterule
3105 %

```

## XIII.6 Footnotes arrangement

### XIII.6.1 User level macro

**\Xarrangement** \Xarrangement[⟨s⟩]{⟨arrangement⟩} The command calls, for each series, a specific command which set many counters and commands in order to define specific arrangement.

```

3106 \newcommand{\Xarrangement}[2][1,usedefault]{%
3107   \def\do##1{%
3108     \csname Xarrangement@#2\endcsname{##1}%
3109   }%
3110   \ifstrempy{#1}%
3111   {%
3112     \dolistloop{\@series}%
3113   }%
3114   {
3115     \docsvlist{#1}%
3116   }%
3117 }%
3118 %

```

### XIII.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\vsizex
\let\vAfootnote=\normalvfootnote \let\Afootfmt=\normalfootfmt
\let\Afootstart=\normalfootstart \let\Afootgroup=\normalfootgroup
\let\Afootnoterule=\normalfootnoterule

```

(Read *The TeXbook* in order to understand what are the counter, skip and dimen associated to an insertion.)

Instead of repeating ourselves, we define a `\Xarrangement@normal` macro that makes all these assignments for us, for any given series letter. This command is called when people use `\Xarrangement[⟨series⟩]{normal}`

Now we set up the `\Xarrangement@normal` macro itself. It takes one argument: the footnote series letter.

```

3119 \newcommand*{\Xarrangement@normal}[1]{%
3120   \csgdef{series@display#1}{normal}
3121   \expandafter\let\csname #1footstart\endcsname=\normalfootstart
3122   \expandafter\let\csname v#1footnote\endcsname=\normalvfootnote
3123   \expandafter\let\csname #1footfmt\endcsname=\normalfootfmt
3124   \expandafter\let\csname #1footgroup\endcsname=\normalfootgroup
3125   \expandafter\let\csname #1footnoterule\endcsname=%
3126                                     \normalfootnoterule
3127   \count\csname #1footins\endcsname=1000
3128   \dimen\csname #1footins\endcsname=\csuse{Xmaxhnotes@#1}
3129   \skip\csname #1footins\endcsname=\csuse{Xbeforenotes@#1}%
3130   \advance\skip\csname #1footins\endcsname by\csuse{Xafterrule@#1}%
3131   %

```

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.

```

3132   \csxdef{default@#1footins}{1000}%Use this to confine the notes to one
3133   side only
3134   %

```

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).

```

3134   \ifnoledgroup@else%
3135     \expandafter\let\csname mpv#1footnote\endcsname=\mpnormalvfootnote
3136     \expandafter\let\csname mp#1footgroup\endcsname=\mpnormalfootgroup
3137     \count\csname mp#1footins\endcsname=1000
3138     \dimen\csname mp#1footins\endcsname=\csuse{Xmaxhnotes@#1}
3139     \skip\csname mp#1footins\endcsname=\csuse{Xbeforenotes@#1}%
3140     \advance\skip\csname mp#1footins\endcsname by\csuse{Xafterrule@#1}%
3141   \fi
3142 }
3143
3144 %

```

`\normalvfootnote` We now begin a series of commands that do 'normal' footnote formatting: a format much like that implemented in PLAIN  $\text{\TeX}$ , in which each footnote is a separate paragraph.

`\normalvfootnote` takes the series letter as #1 and the entire text of the footnote is #2. It does the `\insert` for this note, calling on the `\footfmt` macro for this note series to format the text of the note.

```

3145 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\normalvfootnote}[2]{%
3146 \iftoggle{Xgroupbyline@#1}{%In the case we use \Xgroupbyline, the
insertion is done later, in \add@Xgroupbyline.
3147 \prepare@Xgroupbyline{#1}{#2}{\normalvfootnote@inserted}%
3148 }{%In the case we don't use \Xgroupbyline, the insertion is made directly
3149 \X@beforeinsertion{#1}%
3150 \insert\csname #1footins\endcsname{%
3151 \X@atbegininsertion{#1}%
3152 \normalvfootnote@inserted{#1}{#2}%
3153 }%
3154 }%
3155 }%
3156 %

```

`\normalvfootnote@inserted` The `\normalvfootnote@inserted` macro is expanded to the content to be added to a `\insert` for normal critical footnote.

```

3157 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\normalvfootnote@inserted}[2]{%
3158 \nottoggle{Xgroupbyline@#1}{\noindent}{\csuse{Xhooknote@#1}%
3159 \csuse{Xnotefontsize@#1}%
3160 \iftoggle{Xgroupbyline@#1}{\strut}{}%
3161 \footssplitskips
3162 \ifl@dpairing\ifl@dpadding\else%
3163 \setXnoteswidthliketwocolumns@{#1}%
3164 \fi\fi%
3165 \setXnotespositionliketwocolumns@{#1}%
3166 \spaceskip=\z@skip \xspaceskip=\z@skip%
3167 \csname #1footfmt\endcsname #2{#1}%
3168 }%
3169 %

```

```

\X@beforeinsertion 70 \newcommand{\X@beforeinsertion}[1]{%
3171 \if@ledgroup\else%
3172 \insert@Xtxtbeforenotes{#1}%
3173 \fi%
3174 \csuse{Xbeforeinserting@#1}%
3175 }%
3176 %

```

```

\beforeinsertion@X 77 \newcommand{\beforeinsertion@X}[1]{%
3178 \if@ledgroup\else%
3179 \insert@txtbeforenotesX{#1}%
3180 \fi%
3181 \csuse{beforeinsertingX@#1}%
3182 }%
3183 %

```

```

\X@atbegininsertion84 \newcommand{\X@atbegininsertion}[1]{%
3185   \hsize=\expandafter\dimexpr\csuse{Xwidth@#1}\relax%
3186 }%
3187 %

```

And somewhat different versions of `\normalvfootnote` and `\normalvfootnote@inserted` for minipages.

```

\mpnormalvfootnote88 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\mpnormalvfootnote}[2]{%
3189   \iftoggle{Xgroupbyline@#1}{%
3190     \prepare@Xgroupbyline{#1}{#2}{\mpnormalvfootnote@inserted}%
3191   }%
3192   {%
3193     \global\setbox\@nameuse{mp#1footins}%
3194     \vbox{%
3195       \unvbox\@nameuse{mp#1footins}%
3196       \mpnormalvfootnote@inserted{#1}{#2}%
3197     }%
3198   }%
3199 }%
3200
3201 %

```

```

\mpnormalvfootnote@inserted02 \newcommand{\mpnormalvfootnote@inserted}[2]{%
3203   \noindent\csuse{Xhooknote@#1}%
3204   \csuse{Xnotefontsize@#1}%
3205   \hsize\columnwidth%
3206   \@parboxrestore%
3207   \color@begingroup%
3208   \csname #1footfmt\endcsname #2{#1}\color@endgroup%
3209 }%
3210 %

```

`\normalfootfmt` `\normalfootfmt` is a ‘normal’ macro to take the footnote line and page number information (see V.9 p. 111), 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; #4 is the note’s series. 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.

```

3211 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\normalfootfmt}[4]{%
3212   \Xstorelineinfo{#1}{#4}%
3213   \nottoggle{Xgroupbyline@#4}{\Xledsetnormalparstuff{#4}}{}%
3214   \Xsethangindent{#4}%
3215   \nottoggle{Xgroupbyline@#4}{\rule{z@}{\splittopskip}}{}%
3216   {\printlinefootnote{#1}{#4}}%
3217   \print@lemma{#1}{#2}{#4}%
3218   \csuse{Xwrapcontent@#4}{#3}%

```



```

3219 \nottoggle{Xgroupbyline@#4}{\strut\par}{}%
3220 }%
3221 %

```

**\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 `\skipXfootins` 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 `\skipXprenotes@` is greater than 0 pt, it is used instead of `\skipXfootins` 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`.

```

3222 \newcommand*{\normalfootstart}[1]{%
3223 %

```

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 XIX p. 248. Here is part of this algorithm, when the block of notes are ready to be printed.

```

3224 \ifdimequal{0pt}{\Xprenotes@}{}%
3225 {%
3226 \iftoggle{Xprenotes@}{%
3227 \togglefalse{Xprenotes@}%
3228 \skip\csname #1footins\endcsname=%
3229 \glueexpr\csuse{Xprenotes@}+\csuse{Xafterrule@#1}\relax%
3230 }%
3231 }%
3232 }%
3233 \vskip\skip\csname #1footins\endcsname%
3234 %

```

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 XVI p. 246 for the general description of the problem.

```

3235 \leftskip0pt \rightskip0pt
3236 \ifl@dpairing\else%
3237 \hsize=\old@hsize%
3238 \fi%
3239 \setXnoteswidthliketwocolumns@{#1}%

```

```

3240 \setXnotespositionliketwocolumns@{#1}%
3241 %

```

And now, print the footnote's rule to finish the footnote's introduction.

```

3242 \print@Xfootnoterule{#1}%
3243 }%
3244 %

```

**\normalfootgroup** \normalfootgroup is a standard footnote-grouping macro: it sends the contents of the footnote-insert box to the output page without alteration.

```

3245 \newcommand*{\normalfootgroup}[1]{%
3246 \csuse{Xhookgroup@#1}%
3247 \unvbox\csname #1footins\endcsname%
3248 \hsize=\old@hsize%
3249 }%
3250
3251 %

```

**\mpnormalfootgroup** A somewhat different version for minipages. Note that, in this case, we do not make distinctions between the \Xfootgroup and \Xfootstarts macros.

```

3252 \unless\ifnoledgroup@
3253 \newcommand*{\mpnormalfootgroup}[1]{%
3254 \vskip\skip\@nameuse{mp#1footins}
3255 \ifl@dpairing\ifparledgroup%
3256 \leavevmode\marks\parledgroup@{begin}%
3257 \marks\parledgroup@series{#1}%
3258 \marks\parledgroup@type{Xfootnote}%
3259 \fi\fi\normalcolor%
3260 \ifparledgroup%
3261 \ifl@dpairing%
3262 \else%
3263 \setXnoteswidthliketwocolumns@{#1}%
3264 \setXnotespositionliketwocolumns@{#1}%
3265 \print@Xfootnoterule{#1}%%
3266 \fi%
3267 \else%
3268 \setXnoteswidthliketwocolumns@{#1}%
3269 \setXnotespositionliketwocolumns@{#1}%
3270 \print@Xfootnoterule{#1}%%
3271 \fi%
3272 \setlength{\parindent}{0pt}
3273 \csuse{Xhookgroup@#1}%
3274 \unvbox\csname mp#1footins\endcsname}}
3275 \fi
3276 %

```

### XIII.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  $\TeX$  of ordinary size may not be able to handle more than about 100 notes of this kind on a page.

`\Xarrangement@paragraph` The `\Xarrangement@paragraph` macro sets up everything for one series of the footnotes so that they will be 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.

```

3277 \newcommand*{\Xarrangement@paragraph}[1]{%
3278   \csgdef{series@display#1}{\paragraph}
3279   \expandafter\let\csname #1footstart\endcsname=\parafootstart
3280   \expandafter\let\csname v#1footnote\endcsname=\paravfootnote
3281   \expandafter\let\csname #1footfmt\endcsname=\parafootfmt
3282   \expandafter\let\csname #1footgroup\endcsname=\parafootgroup
3283   \count\csname #1footins\endcsname=1000
3284   \csxdef{default@#1footins}{1000}%Use this to confine the notes to one
side only
3285   \dimen\csname #1footins\endcsname=\csuse{Xmaxhnotes@#1}
3286   \skip\csname #1footins\endcsname=\csuse{Xbeforenotes@#1}%
3287   \advance\skip\csname #1footins\endcsname by\csuse{Xafterrule@#1}%
3288   \para@footsetup{#1}
3289   %

```

And the extra setup for minipages.

```

3290 \ifnoledgroup@else
3291   \expandafter\let\csname mpv#1footnote\endcsname=\mpparavfootnote
3292   \expandafter\let\csname mp#1footgroup\endcsname=\mpparafootgroup
3293   \count\csname mp#1footins\endcsname=1000
3294   \dimen\csname mp#1footins\endcsname=\csuse{Xmaxhnotes@#1}
3295   \skip\csname mp#1footins\endcsname=\csuse{Xbeforenotes@#1}%
3296   \advance\skip\csname mp#1footins\endcsname by\csuse{Xafterrule@#1}%
3297   \fi
3298 }
3299 %

```

`\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.

```

3300 \providecommand{\footfudgefiddle}{64}
3301 %

```

`\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`.

```

3302 \newcommand*{\para@footsetup}[1]{\csuse{Xhookgroup@#1}\csuse{
Xnotefontsize@#1}
3303   \setXnoteswidthliketwocolumns@{#1}%
3304   \ifcempty{Xwidth@#1}%
3305     {}%
3306     {\columnwidth=\expandafter\dimexpr\csuse{Xwidth@#1}\relax}%
3307   \dimen0=\baselineskip
3308   \multiply\dimen0 by 1024
3309   \divide \dimen0 by \columnwidth \multiply\dimen0 by \footfudgefiddle\
relax
3310   \csxdef{#1footfudgefactor}{%
3311     \expandafter\strip@pt\dimen0 }}
3312
3313 %

```

`\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.

```

3314 \newcommand*{\parafootstart}[1]{%
3315   \rightskip=0pt \leftskip=0pt%
3316   \Xsetparindent{#1}%
3317   \ifdimequal{0pt}{\Xprenotes@}{}%
3318     {%
3319       \iftoggle{Xprenotes@}{%
3320         \togglefalse{Xprenotes@}%
3321         \skip\csname #1footins\endcsname=%
3322         \glueexpr\csuse{Xprenotes@}+\csuse{Xafterrule@#1}\relax%
3323       }%
3324     }%
3325   }%
3326   \vskip\skip\csname #1footins\endcsname%
3327   \setXnoteswidthliketwocolumns@{#1}%
3328   \setXnotespositionliketwocolumns@{#1}%
3329   \print@Xfootnoterule{#1}%
3330   \let\old@bidi@RTL@everypar\bidi@RTL@everypar%
3331   \let\bidi@RTL@everypar@empty%
3332   \noindent\leavevmode%

```

```

3333 \let\bidirTL@everypar\old@bidirTL@everypar%
3334 }%
3335 %

```

**\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>30</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>31</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>32</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>33</sup> 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

<sup>30</sup>Michael Downes, 'Line Breaking in \unhboxed Text', *TUGboat* 11 (1990), pp. 605–612.

<sup>31</sup>See *The TeXbook*, p. 455 (editions after January 1990).

<sup>32</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>33</sup>'Line Breaking', p. 610.

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. XIII.6.2 p. 193 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.

```

3336 \newcommand*{\paravfootnote}[2]{%
3337   \csuse{Xbeforeinserting@#1}%
3338   \insert\csname #1footins\endcsname
3339   \bgroup
3340     \csuse{Xnotefontsize@#1}
3341     \footplitskips
3342     \setbox0=\vbox{\hsize=\maxdimen%
3343       \let\bidir@RTL@everypar\@empty%
3344       \insert@Xtxtbeforenotes{#1}%
3345       \noindent\csuse{Xhooknote@#1}%
3346       \csname #1footfmt\endcsname #2{#1}}%
3347     \setbox0=\hbox{\Xunvxh{0}{#1}}%
3348     \dp0=0pt
3349     \ht0=\csname #1footfudgefactor\endcsname\wd0
3350   %

```

Here we produce the contents of the footnote from box 0, and add a penalty of 0 between boxes in this insert.

```

3351   \if@RTL\noindent \leavevmode\fi\box0%
3352   \penalty0
3353   \egroup}
3354
3355 %

```

The final penalty of 0 was added here at Wayne's suggestion to avoid a weird page-breaking problem, which occurs on those occasions when  $\TeX$  attempts to split foot paragraphs. After trying out such a split (see *The TeXbook*, p. 124),  $\TeX$  inserts a penalty of  $-10000$  here, which nearly always forces the break at the end of the whole footnote paragraph (since individual notes can't be split) even when this leads to an overfull vbox. The change above results in a penalty of 0 instead which allows, but does not force, such breaks. This penalty of 0 is later removed, after page breaks have been decided, by the `\unpenalty` macro in `\makehboxofhboxes`. So it does not affect how the footnote paragraphs are typeset (the notes still have a penalty of  $-10$  between them, which is added by `\parafootfmt`).

`\mpparavfootnote` This version is for minipages.

```

3356 \newcommand*{\mpparavfootnote}[2]{%
3357   \global\setbox\@nameuse{mp#1footins}\vbox{%
3358     \unvbox\@nameuse{mp#1footins}%
3359     \csuse{Xnotefontsize@#1}

```

```

3360 \footssplitskip
3361 \setbox0=\vbox{\hsize=\maxdimen%
3362   \let\bidir@RTL@everypar\@empty%
3363   \insert@Xtxtbeforenotes{#1}%
3364   \noindent\color@begingroup%
3365   \csuse{Xbhooknote@#1}%
3366   \csname #1footfmt\endcsname #2{#1}\color@endgroup}%
3367 \setbox0=\hbox{\Xunvxh{0}{#1}}%
3368 \dp0=\z@
3369 \ht0=\csname #1footfudgefactor\endcsname\wd0
3370 \box0
3371 \penalty0
3372 }}
3373
3374 %

```

**\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`.

```

3375 \newcommand*{\Xunvxh}[2]{%
3376   \setbox0=\vbox{\unvbox#1%
3377     \global\setbox1=\lastbox}%
3378   \unhbox1
3379   \unskip           % remove \rightskip,
3380   \unskip           % remove \parfillskip,
3381   \unpenalty        % remove \penalty of 10000,
3382   \hskip\csuse{Xafternote@#2}\relax}% add the glue to go between the notes
3383
3384 %

```

**\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).

```

3385 \newcommand*{\parafootfmt}[4]{%
3386   \Xstorelineinfo{#1}{#4}%
3387   \Xinsertparafootsep{#4}%
3388   \ledsetnormalparstuff@common%
3389   \printlinefootnote{#1}{#4}%
3390   \print@lemma{#1}{#2}{#4}%
3391   \csuse{Xwrapcontent@#4}{#3}%
3392   \penalty-10 }
3393 %

```

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.

```

3394 \newcommand*{\parafootgroup}[1]{%
3395   \hsize=\expandafter\dimexpr\csuse{Xwidth@#1}\relax%
3396   \unvbox\csname #1footins\endcsname
3397   \ifcsstring{Xragged@#1}{L}{\RaggedLeft}{}%
3398   \ifcsstring{Xragged@#1}{R}{\RaggedRight}{}%
3399   \makehboxofhboxes
3400   \setbox0=\hbox{\unhbox0 \removehboxes}%
3401   \csuse{Xbhookgroup@#1}%
3402   \csuse{Xnotefontsize@#1}%
3403   \unhbox0\par%
3404   \global\hsize=\old\hsize%
3405   }%
3406
3407 %

```

`\mpparafootgroup` The minipage version.

```

3408 \newcommand*{\mpparafootgroup}[1]{%
3409   \setXnoteswidthliketwocolumns@{#1}%
3410   \vskip\skip\@nameuse{mp#1footins}
3411   \ifl@dpairing\ifparledgroup%
3412     \leavevmode\marks\parledgroup@{begin}%
3413     \marks\parledgroup@series{#1}%
3414     \marks\parledgroup@type{Xfootnote}%
3415     \fi\fi\normalcolor
3416     \ifparledgroup%
3417       \ifl@dpairing%
3418         \else%
3419           \setXnoteswidthliketwocolumns@{#1}%
3420           \setXnotespositionliketwocolumns@{#1}%
3421           \print@Xfootnoterule{#1}%
3422         \fi%
3423       \else%
3424         \setXnoteswidthliketwocolumns@{#1}%
3425         \setXnotespositionliketwocolumns@{#1}%
3426         \print@Xfootnoterule{#1}%
3427       \fi%
3428     \unvbox\csname mp#1footins\endcsname
3429     \ifcsstring{Xragged@#1}{L}{\RaggedLeft}{}%

```



```

3430 \ifcsstring{Xragged@#1}{R}{\RaggedRight}{}%
3431 \makehboxofhboxes
3432 \setbox0=\hbox{\unhbox0 \removehboxes}%
3433 \csuse{Xbhookgroup@#1}%
3434 \csuse{Xnotefontsize@#1}%
3435 \Xsetparindent{#1}%
3436 \unhbox0\par}}
3437
3438 %

```

And finally, the two macros which are required to transform the long horizontal box stored in the insert' box to a printable text.

```

\makehboxofhboxes 39 \newcommand*{\makehboxofhboxes}{\setbox0=\hbox{}}%
\removehboxes 40 \loop
3441 \unpenalty
3442 \setbox2=\lastbox
3443 \ifhbox2
3444 \setbox0=\hbox{\box2\unhbox0}%
3445 \repeat}
3446
3447 \newcommand*{\removehboxes}{\setbox0=\lastbox
3448 \ifhbox0{\removehboxes}\unhbox0 \fi}
3449
3450 %

```

**Insertion of the footnotes separator** The command `\Xinsertparafootsep{<series>}` must be called at the beginning of `\parafootftm`.

```

\prevpage@num 51 \newcommand{\Xinsertparafootsep}[1]{%
\Xinsertparafootsep 52 \ifledRcol{%
3453 \ifnumequal{\csuse{#1prevpage@numR}}{\page@numR}%
3454 {\ifcsdef{prevline#1}% Be sur \prevline#1 exists.
3455 {\ifcsequal{prevline#1}{lineinfo@}%
3456 {\ifcsequal{Xsymlinenum@#1}{\csuse{Xparafootsep@#1}}{}}%
3457 {\csuse{Xparafootsep@#1}}%
3458 }%
3459 {\csuse{Xparafootsep@#1}}%
3460 }%
3461 {}}%
3462 \global\csname #1prevpage@numR\endcsname=\page@numR%
3463 \else%
3464 \ifnumequal{\csuse{#1prevpage@num}}{\page@num}%
3465 {\ifcsdef{prevline#1}% Be sur \prevline#1 exists.
3466 {\ifcsequal{prevline#1}{lineinfo@}%
3467 {\ifcsequal{Xsymlinenum@#1}{\csuse{Xparafootsep@#1}}{}}%
3468 {\csuse{Xparafootsep@#1}}%
3469 }%

```

```

3470     {\csuse{Xparafootsep@#1}}}%
3471   }%
3472   }%
3473   \global\csname #1prevpage@num\endcsname=\page@num%
3474   \fi%
3475 }
3476 %

```

### XIII.6.4 Columnar footnotes

#### Common tools

`\rigidbalance` We will now define macros for three-column notes and two-column notes. Both sets of macros will use `\rigidbalance`, which splits a box (#1) into into a number (#2) of columns, each with a space (#3) between the top baseline and the top of the `\vbox`. The `\Xrigidbalance` `\dosplits` `\splitoff` `\@h` `\@k` `\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`.

```

3477 \newcount\@k \newdimen\@h
3478 \newcommand*\Xrigidbalance}[3]{%
3479   \hsize=\expandafter\dimexpr\csuse{Xwidth@\@currentseries}\relax%
3480   \rigidbalance{#1}{#2}{#3}%
3481 }%
3482
3483 \newcommand*\rigidbalanceX}[3]{%
3484   \hsize=\expandafter\dimexpr\csuse{widthX@\@currentseries}\relax%
3485   \rigidbalance{#1}{#2}{#3}%
3486 }%
3487
3488 \newcommand*\rigidbalance}[3]{%
3489   \setbox0=\box#1 \@k=#2 \@h=#3%
3490   \@@line{\splittopskip=\@h \vbadness=\@M \hfilneg
3491     \valign{##\vfil\cr\dosplits}}}%
3492
3493 \newcommand*\dosplits{\ifnum\@k>0 \noalign{\hfil}\splitoff
3494   \global\advance\@k-1\cr\dosplits\fi}
3495
3496 \newcommand*\splitoff{\dimen0=\ht0
3497   \divide\dimen0 by\@k \advance\dimen0 by\@h
3498   \setbox2 \vsplit0 to \dimen0
3499   \unvbox2 }
3500
3501 %

```

### Three columns

```
\Xarrangement@threecol 3502 \newcommand*{\Xarrangement@threecol}[1]{%
3503 \csgdef{series@display#1}{threecol}
3504 \expandafter\let\csname v#1footnote\endcsname=\threecolvfootnote
3505 \expandafter\let\csname #1footfmt\endcsname=\threecolfootfmt
3506 \expandafter\let\csname #1footgroup\endcsname=\threecolfootgroup
3507 \dimen\csname #1footins\endcsname=\csuse{Xmaxhnotes@#1}%
3508 \skip\csname #1footins\endcsname=\csuse{Xbeforenotes@#1}%
3509 \advance\skip\csname #1footins\endcsname by\csuse{Xafterrule@#1}%
3510 \threecolfootsetup{#1}
3511 %
```

The additional setup for minipages.

```
3512 \ifnoledgroup@else
3513 \expandafter\let\csname mpv#1footnote\endcsname=\mpnormalvfootnote
3514 \expandafter\let\csname mp#1footgroup\endcsname=\mpthreecolfootgroup
3515 \skip\csname mp#1footins\endcsname=\csuse{Xbeforenotes@#1}%
3516 \advance\skip\csname mp#1footins\endcsname by\csuse{Xafterrule@#1}%
3517 \mpthreecolfootsetup{#1}
3518 \fi
3519 }
3520
3521 %
```

The `\footstart` and `\footnoterule` macros for these notes assume the normal values (XIII.6.2 p. 193 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  $\text{\TeX}$  is accumulating material for the page and checking that limit, it does not apply the `\count` scaling.

```
3522 \newcommand*{\threecolfootsetup}[1]{%
3523 \count\csname #1footins\endcsname 333
3524 \csxdef{default@#1footins}{333}%Use this to confine the notes to one
side only
3525 \multiply\dimen\csname #1footins\endcsname \thr@@}
3526 %
```

`\mpthreecolfootsetup` The setup for minipages.

```

3527 \newcommand*\mpthreecolfootsetup}[1]{%
3528   \count\csname mp#1footins\endcsname 333
3529   \multiply\dimen\csname mp#1footins\endcsname \thr@@}
3530
3531 %

```

`\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 #1 the full text of the note (including numbers, lemma and text).

```

3532 \notbool{parapparatus@}\newcommand*\newcommand*\threecolvfootnote}[2]{%
3533   \iftoggle{Xgroupbyline@#1}{%
3534     \prepare@Xgroupbyline{#1}{#2}\threecolvfootnote@inserted}%
3535   }%
3536   {%
3537     \let\bidir@RTL@everypar\relax%
3538     \X@beforeinsertion{#1}%
3539     \insert\csname #1footins\endcsname{%
3540       \threecolvfootnote@inserted{#1}{#2}%
3541     }%
3542   }%
3543 }%
3544 %

```

```

\threecolvfootnote@inserted 3545 \notbool{parapparatus@}\newcommand*\newcommand*\threecolvfootnote@inserted}[2]{%
3546   \hsize=\expandafter\dimexpr\csuse{Xwidth@#1}\relax%
3547   \noindent\csuse{Xhooknote@#1}%
3548   \csuse{Xnotefontsize@#1}%
3549   \footsplitskips%
3550   \csname #1footfmt\endcsname #2{#1}%
3551 }%
3552 %

```

`\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.

```

3553 \notbool{parapparatus@}\newcommand*\newcommand*\threecolfootfmt}[4]{%

```

```

3554 \xstorelineinfo{#1}{#4}%
3555 \threecol@begin@insert{#4}%
3556 \hspace{\parindent}%
3557 \printlinefootnote{#1}{#4}%
3558 \print@lemma{#1}{#2}{#4}%
3559 \cuse{Xwrapcontent@#4}{#3}%
3560 \nottoggle{Xgroupbyline@#4}%
3561   {\strut\par\allowbreak}%
3562   }%
3563 }%
3564 %

```

**\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`.

```

3565 \newcommand{\threecol@begin@insert}[1]{%
3566   \normal@pars%
3567   \nottoggle{Xgroupbyline@#1}%
3568     {\hsize \cuse{Xhsizethreecol@#1}}%
3569     }%
3570   \Xsetparindent{#1}%
3571   \tolerance=5000%
3572   \Xsethangindent{#1}%
3573   \@tempdima=\parindent%
3574   \cuse{Xcolalign@#1}%
3575   \parindent=\@tempdima%
3576   \strut%
3577 }%
3578 %

```

**\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.

```

3579 \newcommand*{\threecolfootgroup}[1]{%
3580   \begin@group%
3581     \cuse{Xhookgroup@#1}%
3582     \cuse{Xnotefontsize@#1}%
3583     \par%
3584     \splittopskip=\ht\strutbox%
3585     \expandafter%
3586     \Xrigidbalance\csname #1footins\endcsname \thr@@ \splittopskip%

```

```

3587 \endgroup%
3588 }%
3589 %

```

**\mpthreecolfootgroup** The setup for minipages.

```

3590 \newcommand*\mpthreecolfootgroup}[1]{%
3591 \vskip\skip\@nameuse{mp#1footins}
3592 \ifl@dpairing\ifparledgroup%
3593 \leavevmode\marks\parledgroup@{begin}%
3594 \marks\parledgroup@series{#1}%
3595 \marks\parledgroup@type{Xfootnote}%
3596 \fi\fi\normalcolor
3597 \ifparledgroup%
3598 \ifl@dpairing%
3599 \else%
3600 \setXnoteswidthliketwocolumns@{#1}%
3601 \setXnotespositionliketwocolumns@{#1}%
3602 \print@Xfootnoterule{#1}%
3603 \fi%
3604 \else%
3605 \setXnoteswidthliketwocolumns@{#1}%
3606 \setXnotespositionliketwocolumns@{#1}%
3607 \print@Xfootnoterule{#1}%
3608 \fi%
3609 \csuse{Xbhookgroup@#1}\par%
3610 \splittopskip=\ht\strutbox
3611 \expandafter
3612 \Xrigidbalance\csname mp#1footins\endcsname \thr@@ \splittopskip}}
3613 %
3614 %

```

## Two columns

```

\Xarrangement@twocol 3615 \newcommand*\Xarrangement@twocol}[1]{%
3616 \csgdef{series@display#1}{twocol}
3617 \expandafter\let\csname v#1footnote\endcsname=\twocolvfootnote
3618 \expandafter\let\csname #1footfmt\endcsname=\twocolfootfmt
3619 \expandafter\let\csname #1footgroup\endcsname=\twocolfootgroup
3620 \dimen\csname #1footins\endcsname=\csuse{Xmaxhnotes@#1}%
3621 \skip\csname #1footins\endcsname=\csuse{Xbeforenotes@#1}%
3622 \advance\skip\csname #1footins\endcsname by\csuse{Xafterrule@#1}%
3623 \twocolfootsetup{#1}
3624 %

```

The additional setup for minipages.

```

3625 \ifnoledgroup@else
3626 \expandafter\let\csname mpv#1footnote\endcsname=\mpnormalvfootnote
3627 \expandafter\let\csname mp#1footgroup\endcsname=\mptwocolfootgroup

```

```

3628 \skip\csname mp#1footins\endcsname=\csuse{Xbeforenotes@#1}%
3629 \advance\skip\csname mp#1footins\endcsname by\csuse{Xafterrule@#1}%
3630 \mptwocolfootsetup{#1}
3631 \fi
3632 }
3633
3634 %

```

`\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 `\hsize`.

```

3635 \twocolvfootnote \newcommand*{\twocolfootsetup}[1]{%
3636 \count\csname #1footins\endcsname 500
3637 \csxdef{default@#1footins}{500}%
3638 \multiply\dimen\csname #1footins\endcsname \tw@}
3639 %

```

```

3640 \notbool{parapparatus@}{\newcommand*{\newcommand}{\twocolvfootnote}[2]{%
3641 \iftoggle{Xgroupbyline@#1}{%
3642 \prepare@Xgroupbyline{#1}{#2}{\twocolvfootnote@inserted}%
3643 }{%
3644 \let\bidir@RTL@everypar\relax%
3645 \X@beforeinsertion{#1}%
3646 \insert\csname #1footins\endcsname{%
3647 \twocolvfootnote@inserted{#1}{#2}%
3648 }%
3649 }%
3650 }%
3651 \notbool{parapparatus@}{\newcommand*{\newcommand}{\
twocolvfootnote@inserted}[2]{%
3652 \hsize=\expandafter\dimexpr\csuse{Xwidth@#1}\relax%
3653 \noindent\csuse{Xhooknote@#1}%
3654 \csuse{Xnotefontsize@#1}%
3655 \footssplitsskip%
3656 \csname #1footfmt\endcsname #2{#1}%
3657 }%
3658 %

```

```

3659 \notbool{parapparatus@}{\newcommand*{\newcommand}{\twocolfootfmt}[4]{% 4th
arg is optional, for backward compatibility
3660 \Xstorelineinfo{#1}{#4}%
3661 \twocol@begin@insert{#4}%
3662 \hspace{\parindent}%
3663 \printlinefootnote{#1}{#4}%
3664 \print@lemma{#1}{#2}{#4}%
3665 \csuse{Xwrapcontent@#4}{#3}%
3666 \nottoggle{Xgroupbyline@#4}%
3667 {\strut\par\allowbreak}%
3668 {}%

```

```

3669 }%
3670 %

3671 \newcommand{\twocol@begin@insert}[1]{%
3672   \normal@pars%
3673   \nottoggle{Xgroupbyline@#1}%
3674   {\hsize \csuse{Xhsizetwocol@#1}}%
3675   {}%
3676   \Xsetparindent{#1}%
3677   \tolerance=5000%
3678   \Xsethangindent{#1}%
3679   \@tempdima=\parindent%
3680   \csuse{Xcolalign@#1}%
3681   \parindent=\@tempdima%
3682   \strut%
3683 }%
3684
3685 \newcommand*\twocolfootgroup[1]{%
3686   \begin@group%
3687   \csuse{Xbhookgroup@#1}%
3688   \csuse{Xnotefontsize@#1}%
3689   \par%
3690   \splittopskip=\ht\strutbox%
3691   \expandafter%
3692   \Xrigidbalance\csname #1footins\endcsname \tw@ \splittopskip%
3693   \endgroup%
3694 }%
3695
3696 %

```

`\mptwocolfootsetup` The versions for minipages.

`\mptwocolfootgroup`

```

3697 \newcommand*\mptwocolfootsetup[1]{%
3698   \count\csname mp#1footins\endcsname 500
3699   \multiply\dimen\csname mp#1footins\endcsname \tw@}
3700 %

3701 \newcommand*\mptwocolfootgroup[1]{%
3702   \vskip\skip\@nameuse{mp#1footins}
3703   \ifl@dpairing\ifparledgroup%
3704     \leavevmode\marks\parledgroup@{begin}%
3705     \marks\parledgroup@series{#1}%
3706     \marks\parledgroup@type{Xfootnote}%
3707   \fi\fi\normalcolor
3708   \ifparledgroup%
3709     \ifl@dpairing%
3710     \else%
3711       \setXnoteswidthliketwocolumns@{#1}%
3712       \setXnotespositionliketwocolumns@{#1}%
3713       \print@Xfootnoterule{#1}%

```



```

3714 \fi%
3715 \else%
3716 \setXnoteswidthliketwocolumns@{#1}%
3717 \setXnotespositionliketwocolumns@{#1}%
3718 \print@Xfootnoterule{#1}%
3719 \fi%
3720 \csuse{Xbhookgroup@#1}\par%
3721 \splittopskip=\ht\strutbox
3722 \expandafter
3723 \Xrigidbalance\csname mp#1footins\endcsname \tw@ \splittopskip}}
3724
3725 %

```

### XIII.7 Footnote paragraph indent

**\Xsetparindent** These two commands set the paragraph indentation of the footnotes, depending of the  
**\setparindentX** settings of the user.

```

3726 \newcommand{\Xsetparindent}[1]{%
3727 \nottoggle{Xparindent@#1}{%
3728 \parindent=\z@%
3729 }%
3730 {%
3731 \ifdef{\parindent@beforestanza}{\parindent=\parindent@beforestanza}{}%
3732 }%
3733 }%
3734 \newcommand{\setparindentX}[1]{%
3735 \nottoggle{parindentX@#1}{%
3736 \parindent=\z@%
3737 }%
3738 {%
3739 \ifdef{\parindent@beforestanza}{\parindent=\parindent@beforestanza}{}%
3740 }%
3741 }%
3742 %

```

### XIII.8 Footnote hanging indent

**\Xsethangindent** \Xsethangindent sets the hangindent for a critical footnote, while \sethangindentX  
**\sethangindentX** does it for a familiar footnote.

```

3743 \newcommand{\Xsethangindent}[1]{%
3744 \if@RTL%
3745 \hangindent=-\csuse{Xhangindent@#1}%
3746 \everypar{\hangindent=-\csuse{Xhangindent@#1}}%
3747 \else%
3748 \hangindent=\csuse{Xhangindent@#1}%
3749 \everypar{\hangindent=\csuse{Xhangindent@#1}}%
3750 \fi%

```

```

3751 }%
3752 \newcommand{\sethangindentX}[1]{%
3753   \if@RTL%
3754     \hangindent=-\csuse{hangindentX@#1}%
3755     \everypar{\hangindent=-\csuse{hangindentX@#1}}%
3756   \else%
3757     \hangindent=\csuse{hangindentX@#1}%
3758     \everypar{\hangindent=\csuse{hangindentX@#1}}%
3759   \fi%
3760 }%
3761 %

```

### XIII.9 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.

#### XIII.9.1 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.

```

3762 \def\endashchar{\textnormal{--}}
3763
3764 \newcommand*{\fullstop}{\textnormal{.}}
3765 \def\Xsublinesep@side{\fullstop}
3766
3767 \newcommand*{\rbracket}{\textnormal{%
3768   \csuse{text\csuse{footnote@lang}}}%
3769   \ifluatex%
3770     \ifdefstring{\footnote@luatextextdir}{TRT}{\thinspace[]\thinspace
3771     ]}%
3772   \else%

```

```

3772     \thinspace]%
3773     \fi}%
3774 }%
3775 }
3776
3777 %

```

### XIII.9.2 Pstart number in footnote

`\printpstart` The `\printpstart` macro prints the pstart number for a note.

```

3778 \newcommand{\printpstart}[0]{%
3779   \ifboolexpr{bool{l@dpairing} or bool{l@dprintingpages} or bool{
l@dprintingcolumns}}{%
3780     \ifledRcol%
3781       \thepstartR%
3782     \else%
3783       \thepstartL%
3784     \fi%
3785   }{%
3786     \thepstart%
3787   }%
3788 }
3789 %

```

### XIII.9.3 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).

```

3790 %
3791 \newcommand{\print@lemma}[3]{%
3792   \bgroup%
3793   \nottoggle{Xlemmadisablefontselection@#3}%
3794     {\select@lemmafont#1|}%
3795   }%
3796   \bgroup%
3797     \csuse{Xlemmafont@#3}%Deprecated
3798     \csuse{Xwraplemma@#3}{#2}%
3799   \egroup%
3800   \egroup%
3801   \iftoggle{nosep@}{%
3802     \hskip\csuse{Xinplaceoflemmaseparator@#3}%
3803     \relax%
3804   }%
3805   {\ifcsemtty{Xlemmaseparator@#3}%
3806     {%
3807       \hskip\csuse{Xinplaceoflemmaseparator@#3}%
3808       \relax%

```

```

3809 }%
3810 {%
3811   \nobreak%
3812   \hskip\csuse{Xbeforelemmaseparator@#3}%
3813   \csuse{Xlemmaseparator@#3}%
3814   \hskip\csuse{Xafterlemmaseparator@#3}%
3815   \relax%
3816 }%
3817 }%
3818 }%
3819 %

```

### XIII.9.4 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.

```

3820 \newcommand{\Xstorelineinfo}[2]{%
3821   \l@dp@rsefootspec#1|}%
3822   \iftoggle{Xnumberonlyfirstintwolines@#2}{%
3823     \xdef\lineinfo@{\l@dparsedstartline - \l@dparsedstartsub - \
3824     \l@dparsedendline - \l@dparsedendsub}%
3825   }%
3826   {%
3827     \xdef\lineinfo@{\l@dparsedstartline - \l@dparsedstartsub}%
3828   }%
3829 %

```

**\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`.

```

3830 \newcommand{\printlinefootnote}[2]{%
3831   \xdef\@currentseries{#2}%
3832   \iftoggle{nonum@}{%Try if the line number must printed for this specific
3833   not (by default, yes)
3834     \hspace{\csuse{Xinplaceofnumber@#2}}}%
3835   }%
3836   {%
3837     \iftoggle{Xnonumber@#2}{%Try if the line number must printed (by
3838     default, yes)
3839     }%
3840     \hspace{\csuse{Xinplaceofnumber@#2}}}%
3841   }%

```

```

3841      {%
3842      {\iftoggle{Xnumberonlyfirstinline@#2}% If for this series the
line number must be printed only in the first time.
3843      {%
3844      \ifcsdef{prevline#2}%
3845      {%Be sure the \prevline exists.
3846      \ifcsequal{prevline#2}{\lineinfo@}%Try it
3847      {%
3848      \ifcseempty{Xsymlinenum@#2}%
3849      {%
3850      \hspace{\csuse{Xinplaceofnumber@#2}}%
3851      }%
3852      {\printsymlinefootnotearea{#2}}%
3853      }%
3854      {%
3855      \printlinefootnotearea{#1}{#2}%
3856      }%
3857      }%
3858      {%
3859      \printlinefootnotearea{#1}{#2}%
3860      }%
3861      }%
3862      {%
3863      \printlinefootnotearea{#1}{#2}%
3864      }%
3865      \csxdef{prevline#2}{\lineinfo@}%
3866      }%
3867      }%
3868      }%
3869      }%
3870      }
3871      %

```

**\printsymlinefootnotearea** This macro prints the space before the line symbol, changes the font, when prints the line symbol and the space after it.

```

3872 \newcommand{\printsymlinefootnotearea}[1]{%
3873 \hspace{\csuse{Xbeforeosymlinenum@#1}}%
3874 \csuse{Xnotenumfont@#1}%
3875 \ifdimequal{\csuse{Xboxsymlinenum@#1}}{\z@}%
3876 {\csuse{Xsymlinenum@#1}}%
3877 {\hbox to \csuse{Xboxsymlinenum@#1}%
3878 {\csuse{Xsymlinenum@#1}\hfill}%
3879 }%
3880 \hspace{\csuse{Xaftersymlinenum@#1}}%
3881 }%
3882 %

```

**\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.)

```

3883 \newcommand{\printlinefootnotearea}[2]{%
3884   \printXbeforenumber{#2}%
3885   \csuse{Xnotenumfont@#2}%
3886   \boxfootnotenumbers{#1}{#2}%
3887   \printXafternumber{#2}%
3888 }%
3889 %

```

**\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.

```

3890 \newcommand{\boxfootnotenumbers}[2]{%
3891   \ifdimequal{\csuse{Xboxlinenum@#2}}{0pt}{%
3892     \printlinefootnotenumbers{#1}{#2}%
3893   }%
3894   {%
3895     \hbox to \csuse{Xboxlinenum@#2}%
3896     {%
3897       \IfSubStr{RC}{\csuse{Xboxlinenumalign@#2}}{\hfill}{}%
3898       \printlinefootnotenumbers{#1}{#2}%
3899       \IfSubStr{LC}{\csuse{Xboxlinenumalign@#2}}{\hfill}{}%
3900     }%
3901   }%
3902 }%
3903 %

```

**\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.

```

3904 \newcommand{\printlinefootnotenumbers}[2]{%
3905   \get@pstartcounter%
3906   \ifboolexpr{%
3907     (togl{Xpstart@#2} and bool{numberpstart})%
3908     or togl{Xpstarteverytime@#2}%
3909     and not (%
3910       togl{Xpstartonlyfirst@#2}%
3911       and test {\ifnumequal{\csuse{Xprevpstart@#2}}{\current@pstart}}%
3912     )%
3913   }%
3914   {%
3915     \printpstart%
3916     \csuse{Xpstartseparator@#2}%
3917   }{}%
3918   \global\cslet{Xprevpstart@#2}{\current@pstart}%
3919   \get@stanzacounter%

```

```

3920 \ifboolexpr{%
3921   togl{Xstanza@#2}%
3922   and bool {numberstanza}%
3923   and not (%
3924     togl{Xpstartonlyfirst@#2}%
3925     and test {\ifnumequal{\csuse{Xprevstanza@#2}}{\current@stanza}}%
3926   )%
3927 }{%
3928   \printstanza%
3929   \csuse{Xstanzaseparator@#2}%
3930 }{%
3931   \global\cslet{Xprevstanza@#2}{\current@stanza}%
3932   \iftoggle{Xonlypstart@#2}{%
3933     \csuse{Xtxtbeforenumber@#2}%
3934     \printlines#1|\ifledRcol@{\Rlineflag\fi}}%
3935 }%
3936 %

```

**\get@pstartcounter** A single macro to get the current pstart counter.

```

3937 \newcommand{\get@pstartcounter}[0]{%
3938   \ifboolexpr{bool{l@dprintingpages} or bool{l@dprintingcolumns}}%
3939   {%
3940     \ifledRcol%
3941       \global\numdef\current@pstart{\the\c@pstartR}%
3942     \else%
3943       \global\numdef\current@pstart{\the\c@pstartL}%
3944     \fi%
3945   }%
3946   {%
3947     \global\numdef\current@pstart{\the\c@pstart}%
3948   }%
3949 }%
3950 %

```

**\get@stanzacounter** The same to get the current stanza counter.

```

3951 \newcommand{\get@stanzacounter}[0]{%
3952   \ifboolexpr{bool{l@dprintingpages} or bool{l@dprintingcolumns}}%
3953   {%
3954     \ifledRcol%
3955       \global\numdef\current@stanza{\the\c@stanzaR}%
3956     \else%
3957       \global\numdef\current@stanza{\the\c@stanzaL}%
3958     \fi%
3959   }%
3960   {%
3961     \global\numdef\current@stanza{\the\c@stanza}%
3962   }%
3963 }%

```

```
3964 %
```

`\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.)

```
3965 \newcommand{\printXbeforenumber}[1]{%
3966   \hspace{\csuse{Xbeforenumber@#1}}%
3967 }%
3968 %
```

`\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

```
3969 \newcommand{\printXafternumber}[1]{%
3970   \iftoggle{Xnonbreakableafternumber@#1}{\nobreak}{}%
3971   \hspace{\csuse{Xafternumber@#1}}%
3972 }%
3973 %
```

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. 111: 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.

Maïeul 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. Is also defines `\@annot@start@print` and `\@annot@end@print` which define annotations associated with the starting and ending line numbers in critical footnotes.

```
\ifl@d@pnum74 \newif\ifl@d@pnum
\ifl@d@ssub75 \newif\ifl@d@ssub
\ifl@d@elin76 \newif\ifl@d@elin
\ifl@d@esl77 \newif\ifl@d@esl
\ifl@d@dash78 \newif\ifl@d@dash

\ifl@d@Xtwolines
\ifl@d@Xmorethantwolines
\@annot@start@print
\@annot@end@print
```



```

3979 \newif\ifl@d@Xtwolines%
3980 \newif\ifl@d@Xmorethantwolines%
3981 \let\@annot@start@print\relax%
3982 \let\@annot@end@print\relax%
3983 %

```

`\l@dparsefootspec` `\l@dp@rsefootspec{<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@dp@rsefootspec` which defines macros holding the numeric values. In many cases, this last command is called directly. Just a reminder of the arguments:

<code>\l@dparseendpage</code>	<code>\printlines</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@dparsestartpage</code>	<code>\printlines</code>	start-page   line   subline   end-page   line   subline   fontflag												
<code>\l@dparsestartline</code>														
<code>\l@dparsestartsub</code>														
<code>\l@dparseendpage</code>														
<code>\l@dparseendline</code>														
<code>\l@dparseendsub</code>														

```

3984 \newcommand*\l@dparsefootspec}[3]{\l@dp@rsefootspec#1|}
3985 \def\l@dp@rsefootspec#1|#2|#3|#4|#5|#6|#7|{%
3986   \gdef\l@dparsestartpage{#1}%
3987   \gdef\l@dparsestartline{#2}%
3988   \gdef\l@dparsestartsub{#3}%
3989   \gdef\l@dparseendpage{#4}%
3990   \gdef\l@dparseendline{#5}%
3991   \gdef\l@dparseendsub{#6}%
3992 }
3993 %

```

Initialise the several number value macros.

```

3994 \def\l@dparsestartpage{0}%
3995 \def\l@dparsestartline{0}%
3996 \def\l@dparsestartsub{0}%
3997 \def\l@dparseendpage{0}%
3998 \def\l@dparseendline{0}%
3999 \def\l@dparseendsub{0}%
4000
4001 %

```

`\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`.

```

4002 \newcommand*\setprintlines}[6]{%
4003   \let\@annot@start@print\relax%
4004   \let\@annot@end@print\relax%
4005   \l@d@pnumfalse%
4006   \l@d@dashfalse%
4007   \l@d@elinfalse%
4008 %

```

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

```

4009 \ifbypage@
4010   \ifnum#4=#1 \else
4011     \l@d@pnumtrue
4012     \l@d@dashtrue
4013   \fi
4014 \fi
4015 %

```

We print the ending line number if: (1) we are printing the ending page number, or (2) the ending line number is different from the starting line number. If either of these conditions is true, we also print the annotation linked to the ending line number annotations.

```

4016 \ifboolexpr{%
4017   bool{\l@d@pnum}%
4018   or not test{\ifnumequal{#2}{#5}}}%
4019 }{%
4020   \l@d@elintrue%
4021   \l@d@dashtrue%
4022   \unless\ifx\relax\annot@end%
4023     \def\@annot@end@print{%
4024       \l@wrapcs@ifnotemptybox{Xwraplinenumannotation@\@currentseries}{\
annot@end}%
4025     }%
4026   \fi%
4027 }{%
4028 %

```

We define the starting line number annotation as a merge of the starting annotation and ending annotation if we don't print the ending line number. Otherwise, it is only the starting annotation.

```

4029 \ifl@d@elin%
4030   \def\@annot@start@print{%
4031     \l@wrapcs@ifnotemptybox{Xwraplinenumannotation@\@currentseries}{\
annot@start}%
4032   }%
4033 \else%
4034   \ifx\annot@start\annot@end%
4035     \unless\ifx\@annot@start\relax%
4036       \def\@annot@start@print{%
4037         \l@wrapcs@ifnotemptybox{Xwraplinenumannotation@\@currentseries}{\
annot@start}%
4038       }%
4039     \fi%
4040   \else%
4041     \ifx\@annot@end@print\relax%
4042       \def\@annot@start@print{%
4043         \l@wrapcs@ifnotemptybox{Xwraplinenumannotation@\@currentseries}{%
4044           \ifx\annot@start\empty\else%
4045             \annot@start%

```

```

4046         \ifdefined\linangesep@%
4047             \linangesep@%
4048         \else%
4049             \csuse{Xlinangeseparator@}\@currentseries}%
4050         \fi%
4051     \fi%
4052     \annot@end%
4053 }%
4054 }%
4055 \else%
4056     \let\@annot@start@print\@annot@end@print%
4057     \let\@annot@end@print\relax%
4058 \fi%
4059 \fi%
4060 \fi%
4061 %

```

We print the starting sub-line if it is nonzero.

```

4062 \l@ds@subfalse
4063 \ifnum#3=0 \else
4064     \l@ds@subtrue
4065 \fi
4066 %

```

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.

```

4067 \l@ds@eslfalse
4068 \ifnum#6=0 \else
4069     \ifnum#6=#3
4070         \ifl@ds@elin \l@ds@esltrue \else \l@ds@eslfalse \fi
4071     \else
4072         \l@ds@esltrue
4073         \l@ds@dashtrue
4074     \fi
4075 \fi%
4076 %

```

However, if the `\Xtwolines` is set for the current series, we do not print the last line number.

```

4077 \ifl@ds@dash%
4078     \ifboolexpr{togl{fulllines@} or test{\ifcempty{Xtwolines@}\@currentseries}}}%
4079     {%
4080     {%
4081         \setistwofollowinglines{#1}{#2}{#4}{#5}%
4082         \ifboolexpr{%
4083             (%
4084                 togl {Xtwolinesbutnotmore@\@currentseries}%
4085                 and not%

```

```

4086         (%)
4087         bool {istwofollowinglines@}%
4088     )%
4089 }%
4090 or%
4091 (%)
4092     (not test{\ifnumequal{#1}{#4}})%
4093     and togl{Xtwolinesonlyinsamepage@\@currentseries}%
4094 )%
4095 }%
4096 {}%
4097 {%
4098     \l@d@dashfalse%
4099     \l@d@Xtwolinestrue%
4100     \l@d@elinfalse%
4101     \l@d@eslfalse%
4102     \ifcempty{Xmorethantwolines@\@currentseries}%
4103     {%
4104         {\ifistwofollowinglines@\else%
4105             \l@d@Xmorethantwolinestrue%
4106             \fi%
4107         }%
4108     }%
4109 }%
4110 \fi%
4111 %

```

If the `\Xnoidenticallinenumannotation` is set for this series, we check if the ending annotation is identical to the starting. If true, we don't print the ending annotation

```

4112 \iftoggle{Xnoidenticallinenumannotation@\@currentseries}{%
4113     \ifx\annot@start\annot@end%
4114         \let\annot@end\print\relax%
4115         \ifx\linenumrep\@gobble%Don't print the dash if we're not printing
the line number
4116         \l@d@dashfalse%
4117         \fi%
4118     \fi%
4119 }{%
4120 %

```

Finally, we check for `\Xlinenumannotationonlyfirst` and `\Xlinenumannotationonlyfirstintwo`, and we redefine, if required, `\@annot@start@print` and `\annot@end@print`. We also store the current line number annotations.

```

4121 \iftoggle{Xlinenumannotationonlyfirst@\@currentseries}{%
4122     \ifboolexpr{%
4123         (%)
4124         togl{Xlinenumannotationonlyfirstintwo@\@currentseries}%
4125         and test{\ifcsequal{annot@start}{prevannot@start@\@currentseries}}%
4126         and test{\ifcsequal{annot@end}{prevannot@end@\@currentseries}}%

```

```

4127     )%
4128     or%
4129     (%
4130         not togl{Xlinenumannotationonlyfirstintwo@\@currentseries}%
4131         test{\ifcsequal{annot@start}{prevannot@start@\@currentseries}}}%
4132     )%
4133 }{%
4134     \def\@annot@start@print{%
4135         \l@wrapcs@ifnotemptybox{Xwraplinenumannotation@\@currentseries}{\
csuse{Xsymlinenumannotation@\@currentseries}}}%
4136     }%
4137     \let\@annot@end@print\relax%
4138     \ifx\linenumrep\@gobble%Don't print the dash if we're not printing
the line number
4139         \l@d@d@dashfalse%
4140     \fi%
4141 }{%}%
4142 \global\cslet{prevannot@start@\@currentseries}{\annot@start}%
4143 \global\cslet{prevannot@end@\@currentseries}{\annot@end}%
4144 }%
4145 {}%
4146 %

```

End of \setprintlines.

```

4147 }%
4148 %

```

**\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.

```

4149 \newif\ifistwofollowinglines@%
4150 \newcommand{\setistwofollowinglines}[4]{%
4151   \ifcsdef{lastlinenumberon@#1}%
4152     {\numdef{tmp}{\csuse{lastlinenumberon@#1}}}%
4153     {\numdef{tmp}{0}}%
4154   \istwofollowinglines@false%
4155   \ifnumequal{#4-#2}{1}%
4156     {\istwofollowinglines@true}%
4157   {\ifbypage@%
4158     \ifnumequal{#3-#1}{1}%
4159     {%
4160       \ifnumequal{#2}{tmp}%
4161       {\ifnumequal{#4}{1}{\istwofollowinglines@true}{}}%
4162       {}%
4163     }%
4164     {}%
4165   \fi%
4166 }%
4167 }%
4168 %

```

`\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

```

4169 \def\printlines#1|#2|#3|#4|#5|#6|#7|#8|{%
4170   \begingroup%
4171   %

```

Decide which part of line number components we will print.

```

4172   \setprintlines{#1}{#2}{#3}{#4}{#5}{#6}%
4173   %

```

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.

```

4174   \ifdimequal{\csuse{Xboxstartlinenum@\@currentseries}}{0pt}%
4175     {\bgroup}%
4176     {\leavevmode\hbox to \csuse{Xboxstartlinenum@\@currentseries}\bgroup\
hfill}%
4177   \ifcsstring{Xlinenumannotationposition@\@currentseries}{before}%
4178     {\@annot@start@print}%
4179     {}%
4180   \ifl@d@pnum%
4181     \wrap@edcrossref{\@this@crossref@start}{#1}%
4182     \csuse{Xpagelinesep@\@currentseries}%
4183   \fi%

```

```

4184 \wrap@edcrossref{\@this@crossref@start}{%
4185 \linenumrep{#2}%
4186 \iftoggle{Xlineflag@\@currentseries}{#8}{}%
4187 }%
4188 \ifl@d@ssub%
4189 \csuse{Xsublinesep@\@currentseries}%
4190 \wrap@edcrossref{\@this@crossref@start}{\sublinenumrep{#3}}%
4191 \fi
4192 \ifcsstring{Xlinenumannotationposition@\@currentseries}{after}%
4193 {\@annot@start@print}%
4194 {}%
4195 \egroup%
4196 %

```

Then print the dash + end line number, or the range symbol.

```

4197 \ifdimequal{\csuse{Xboxendlinenum@\@currentseries}}{0pt}%
4198 {\bgroup}%
4199 {\hbox to \csuse{Xboxendlinenum@\@currentseries}\bgroup}%
4200 \ifl@d@Xtwolines%
4201 \ifl@d@Xmorethantwolines%
4202 \csuse{Xmorethantwolines@\@currentseries}%
4203 \else%
4204 \csuse{Xtwolines@\@currentseries}%
4205 \fi%
4206 \else%
4207 \ifl@d@dash%
4208 \ifdefined\linerangesep%
4209 \linerangesep%
4210 \else%
4211 \csuse{Xlinerangeseparator@\@currentseries}%
4212 \fi%
4213 \fi%
4214 \ifcsstring{Xlinenumannotationposition@\@currentseries}{before}%
4215 {\@annot@end@print}%
4216 {}%
4217 \ifl@d@pnum%
4218 \wrap@edcrossref{\@this@crossref@end}{#4}%
4219 \csuse{Xpagelinesep@\@currentseries}%
4220 \fi%
4221 \ifl@d@elin%
4222 \wrap@edcrossref{\@this@crossref@end}{%
4223 \linenumrep{#5}%
4224 \iftoggle{Xlineflag@\@currentseries}{#8}{}%
4225 }%
4226 \fi%
4227 \ifl@d@esl%
4228 \ifl@d@elin%
4229 \csuse{Xsublinesep@\@currentseries}%
4230 \fi%
4231 \wrap@edcrossref{\@this@crossref@end}{\sublinenumrep{#6}}%

```

```

4232 \fi%
4233 \ifcsstring{Xlinenumannotationposition@ \@currentseries}{after}%
4234 {\@annot@end@print}%
4235 }%
4236 \fi%
4237 \ifdimequal{\csuse{Xboxendlinenum@ \@currentseries}}{0pt}%
4238 {}%
4239 {\hfill}%Prevent underfull hbox
4240 \egroup%
4241 \endgroup%
4242 }%
4243 %

```

### XIII.9.5 Footnote grouped by line

`\prepare@Xgroupbyline` `\prepare@Xgroupbyline` is a macro called on the `\metaXXXvfootnote` if `\Xgroupbyline` is set to true, instead of calling `\insert` directly. #1 The series  
#2 The content of the footnote, which is also the line number indication  
#3 This macro, which prepares the contents of the insertion

```

4244 \newcommand{\prepare@Xgroupbyline}[3]{%
4245 \iftoggle{Xgroupbylineseparetwolines@#1}{%
4246 \l@dparsfootspec#2%
4247 \ifcsdef{#1@forinserting@ \l@dparsedendpage- \l@dparsedendline- \l@dparsedendsub}%
4248 {%
4249 \csgappto{#1@forinserting@ \l@dparsedendpage- \l@dparsedendline- \l@dparsedendsub}{%
4250 \hskip\csuse{Xafternote@#1}\relax%
4251 }%
4252 }%
4253 }%
4254 \add@hooktoggle@specific@to@cs{#1@forinserting@ \l@dparsedendpage- \l@dparsedendline- \l@dparsedendsub}%
4255 \add@hookarg@specific@to@cs{#1@forinserting@ \l@dparsedendpage- \l@dparsedendline- \l@dparsedendsub}%
4256 \csxappto%
4257 {#1@forinserting@ \l@dparsedendpage- \l@dparsedendline- \l@dparsedendsub}%
4258 }%
4259 {%
4260 \keep@this@crossref@forinserting%
4261 \unexpanded{%
4262 \ifcseempty{Xsymlinenum@#1}%
4263 {\csuse{Xparafootsep@#1}}%
4264 }%
4265 }%
4266 }%

```



```

4267 \listcsxadd{#1@forinserting}{\l@dparsedendpage-\l@dparsedendline-\
l@dparsedendsub}%
4268 }{%
4269 \ifcsdef{#1@forinserting@all}{%
4270 \csgappto%
4271 {#1@forinserting@all}%
4272 {\hskip\csuse{Xafternote@#1}\relax}%
4273 }{}%
4274 \add@hooktoggle@specific@to@cs{#1@forinserting@all}%
4275 \add@hookarg@specific@to@cs{#1@forinserting@all}%
4276 \csxappto%
4277 {#1@forinserting@all}%
4278 {%
4279 \keep@this@crossref@forinserting%
4280 \unexpanded{%
4281 \ifcseempty{Xsymlinenum@#1}%
4282 {\csuse{Xparafootsep@#1}}%
4283 }{}%
4284 #3{#1}{#2}%
4285 }%
4286 }%
4287 }%
4288 \listcsgadd{#1@forinserting}{all}%
4289 }%
4290 %

```

`\this@crossref@forinserting` `\keep@this@crossref@forinserting` is called by `\prepare@Xgroupbyline`. It just ensures that `\@this@crossref@start` and `\this@crossref@end` value are kept in memory.

```

4291 \newcommand{\keep@this@crossref@forinserting}{%
4292 \unexpanded{\gdef\@this@crossref@start}{\@this@crossref@start}%
4293 \unexpanded{\gdef\@this@crossref@end}{\@this@crossref@end}%
4294 }%
4295 %

```

## XIV Familiar footnotes

### XIV.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` 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`.

```
4296 \providecommand*\multiplefootnotemarker}{3sp}
4297 \providecommand*\multfootsep{\textsuperscript{\normalfont,}}
4298
4299 %
```

`\m@mmf@prepare` A pair of self-cancelling kerns. This may have been defined in the `memoir` class.

```
4300 \providecommand*\m@mmf@prepare}{%
4301   \kern-\multiplefootnotemarker
4302   \kern\multiplefootnotemarker\relax}
4303 %
```

`\m@mmf@check` This may have been defined in the `memoir` class. If it recognises the last kern as `\multiplefootnotemarker` it typesets `\multfootsep`.

```
4304 \providecommand*\m@mmf@check}{%
4305   \ifdim\lastkern=\multiplefootnotemarker\relax
4306     \edef\@x@sf{\the\spacefactor}%
4307     \unkern
4308     \multfootsep
4309     \spacefactor\@x@sf\relax
4310   \fi}
4311
4312 %
```

We have to modify `\@footnotetext` and `\@footnotemark`. However, if `memoir` is used the modifications have already been made.

```
4313 \@ifclassloaded{memoir}{\%
4314 %
```

`\@footnotetext` Add `\m@mmf@prepare` at the end of `\@footnotetext`.

```
4315 \apptocmd{\@footnotetext}{\m@mmf@prepare}{\%
4316 %
```

`\@footnotemark` Modify `\@footnotemark` to cater for adjacent `\footnotes`.

```
4317
4318 \patchcmd{\@footnotemark}
4319   {\nobreak}
4320   {\m@mmf@check
```

```

4321 \nobreak
4322 }
4323 {}{}
4324 \patchcmd{\@footnotemark}
4325 {\@makefnmark}
4326 {\@makefnmark
4327 \m@mmf@prepare
4328 }
4329 {}{}
4330 %

```

Finished the modifications for the non-memoir case.

```

4331 }
4332
4333 %

```

## XIV.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.

```

4334 \pretocmd{\@footnotetext}{%
4335 \ifnumberedpar@
4336 \edtext{}{\l@dbfnote{#1}}}%
4337 \else
4338 }{}{}
4339 \apptocmd{\@footnotetext}{\fi}{}{}%
4340 %

```

`\l@dbfnote` `\l@dbfnote` adds the footnote to the insert list, and `\vl@dbfnote` calls the original `\@footnotetext`. We also patch `\footnote` in order to get the correct footnote numbers when typesetting parallel texts. This is moved into a `\get@fnmark` command.

```

\footnote
\get@fnmark
\get@thisfootnote
4341
4342 \patchcmd%
4343 {\footnote}%
4344 {\stepcounter\@mpfn}%
4345 {%
4346 \ifboolexpr{bool{l@dpairing} or bool{l@dprintingpages} or bool{
l@dprintingcolumns}}{%
4347 \global\advance\footnote@reading by \@ne%
4348 \get@thisfootnote%
4349 \get@fnmark{\thisc@footnote}%
4350 \ifcsdef{footnotereading\the\footnote@reading=typeset}%
4351 {\setcounter{\@mpfn}{\csuse{footnotereading\the\footnote@reading=
typeset}}}%

```

```

4352     {\setcounter{\@mpfn}{\footnote@reading}}}%
4353     }{%
4354     \stepcounter{\@mpfn}%
4355     }%
4356   }%
4357   {}
4358   {}
4359
4360 \newcommand{\get@thisfootnote}{%
4361   \ifboolexpr{bool{l@dpairing} or bool{l@dprintingpages} or bool{
4362     l@dprintingcolumns}}{%
4363     \protected@xdef\thisc@footnote{\the\footnote@reading}%
4364   }{%
4365     \protected@xdef\thisc@footnote{\the\c@footnote}%
4366   }%
4367 }%
4368
4369 \newcommand{\l@dbfnote}[1]{%
4370   \get@thisfootnote%
4371   \gdef\@tag{#1\relax}%
4372   \ifledRcol%
4373     \xright@appenditem{%
4374       \ifdefined\Hy@footnote@currentHref%
4375         \noexpand\def\noexpand\Hy@footnote@currentHref{\
4376         Hy@footnote@currentHref}%
4377       \fi%
4378       \noexpand\vl@dbfnote{{\expandonce\@tag}}{\thisc@footnote}%
4379     }%
4380     \to\inserts@listR
4381     \global\advance\insert@countR \@ne%
4382   \else%
4383     \xright@appenditem{%
4384       \ifdefined\Hy@footnote@currentHref%
4385         \noexpand\def\noexpand\Hy@footnote@currentHref{\
4386         Hy@footnote@currentHref}%
4387       \fi%
4388       \noexpand\vl@dbfnote{{\expandonce\@tag}}{\thisc@footnote}%
4389     }%
4390     \to\inserts@list
4391     \global\advance\insert@count \@ne%
4392   \fi
4393   \ignorespaces%
4394 }%
4395
4396 \newcommand{\get@fnmark}[1]{%
4397   \ifboolexpr{bool{l@dpairing} or bool{l@dprintingpages} or bool{
4398     l@dprintingcolumns}}{%
4399     {%
4400       \stepcounter{footnote@typeset}%
4401       \setcounter{footnote}{\c@footnote@typeset}%

```

```

4398 \immediate\write\@mainaux{%
4399 \csgdef{footnotereading#1=typeset}{\the\c@footnote@typeset}%
4400 }%
4401 \def\@thefnmark{\thefootnote}%
4402 }%
4403 {%
4404 \setcounter{footnote}{#1}%
4405 \def\@thefnmark{\thefootnote}%
4406 }%
4407 }%
4408
4409 \newcommand{\v1@dbfnote}[2]{%
4410 \get@fnmark{#2}%
4411 \@footnotetext{#1}%
4412 }%
4413 %

```

### XIV.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
4414 \newcommand*{\prebodyfootmark}{%
4415 \leavevmode
4416 \ifhmode
4417 \edef\x@sf{\the\spacefactor}%
4418 \m@mmf@check
4419 \nobreak
4420 \fi}
4421 \newcommand{\postbodyfootmark}{%
4422 \m@mmf@prepare
4423 \ifhmode\spacefactor\x@sf\fi\relax}
4424
4425 %

```

### XIV.4 Footnote arrangement

#### XIV.4.1 User level macro

`\arrangementX` `\arrangementX[⟨s⟩]{⟨arrangement⟩}` command calls, for each series, a specific command which set many counters and commands in order to define specific arrangement.

```

4426 \newcommandx{\arrangementX}[2][1,usedefault]{%
4427 \def\do##1{%
4428 \csname arrangementX@#2\endcsname{##1}%
4429 }%
4430 \ifstrempy{#1}%

```

```

4431    {%
4432    \dolistloop{\@series}%
4433    }%
4434    {
4435    \docsvlist{#1}%
4436    }%
4437  }%
4438  %

```

#### XIV.4.2 Normal footnotes

`\normal@footnotemarkX` `\normal@footnotemarkX{<series>}` sets up the typesetting of the marker at the point where the footnote is called for.

```

4439 \newcommand*{\normal@footnotemarkX}[1]{%
4440   \prebodyfootmark
4441   \wrapped@bodyfootmarkX{#1}%
4442   \postbodyfootmark}
4443
4444 %

```

`\normalbodyfootmarkX` The `\normalbodyfootmarkX{<series>}` *really* typesets the in-text marker. The style is the normal superscript.

```

4445 \newcommand*{\normalbodyfootmarkX}[1]{%
4446   \hbox{\textsuperscript{\normalfont\@nameuse{@thefnmark#1}}}}
4447 %

```

`\normalvfootnoteX` `\normalvfootnoteX{<series>}{<text>}` does the `\insert` for the `<series>` and calls the series' `\footfmt...` to format the `<text>`.

```

4448 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\normalvfootnoteX}[2]{%
4449   \beforeinsertionX{#1}%
4450   \insert\@nameuse{footins#1}\bgroup
4451     \fontseries{\seriesdefault}%
4452     \fontshape{\shapedefault}%
4453     \hsize=\expandafter\dimexpr\csuse{widthX@#1}\relax%
4454     \noindent\csuse{bhooknoteX@#1}%
4455     \csuse{notefontsizeX@#1}%
4456     \footsplitskips
4457     \ifl@dpairing\ifl@dpaging\else%
4458       \setnoteswidthliketwocolumnsX@{#1}%
4459     \fi\fi%
4460     \setnotesXpositionliketwocolumns@{#1}%
4461     \spaceskip=\z@skip \xspaceskip=\z@skip
4462     \csuse{\csuse{footnote@dir}}{\@nameuse{footfmt#1}{#1}{#2}\egroup}
4463
4464 %

```

`\mpnormalvfootnoteX` The minipage version.

```

4465 \newcommand*{\mpnormalvfootnoteX}[3]{%
4466   \get@thisfootnoteX{#1}%
4467   \get@fnmarkX{#1}{\thisc@footnote}{#3}%
4468   \ifstrepty{#3}{%
4469     \edef\this@footnoteX@reading{\the\c@name footnote#1@reading\endcsname}%
4470   }{%
4471     \edef\this@footnoteX@reading{###3}%
4472   }%
4473   \global\setbox\@nameuse{mpfootins#1}\vbox{%
4474     \unvbox\@nameuse{mpfootins#1}
4475     \noindent\csuse{bhooknoteX@#1}%
4476     \csuse{notefontsizeX@#1}%
4477     \hsize\columnwidth
4478     \@parboxrestore
4479     \color@begingroup
4480     \@nameuse{footfmt#1}{#1}{#2}\color@endgroup}}
4481
4482 %

```

`\normalfootfmtX` `\normalfootfmtX{<series>}{<text>}` typesets the footnote text, prepended by the marker.

```

4483 \notbool{parapparatus@}{\newcommand*{\newcommand}{\normalfootfmtX}[2]{%
4484   \ifluatex%
4485     \texdir\footnote@luatextextdir%
4486     \pardir\footnote@luatexpardir%
4487   \fi%
4488   \protected@edef\@currentlabel{%
4489     \@nameuse{@thefnmark#1}%
4490   }%
4491   \ledsetnormalparstuffX{#1}%
4492   \sethangindentX{#1}%
4493   \rule{z@}{splittopskip}%
4494   {\csuse{notenumfontX@#1}\wrapped@footfootmarkX{#1}}%
4495   \csuse{wrapcontentX@#1}{#2}%
4496   \strut\par}}
4497
4498 %

```

`\normalfootfootmarkX` `\normalfootfootmarkX{<series>}` is called by `\normalfootfmtX` to typeset the footnote marker in the footer before the footnote text.

```

4499 \newcommand*{\normalfootfootmarkX}[1]{%
4500   \textsuperscript{\@nameuse{@thefnmark#1}}}
4501
4502 %

```

`\normalfootstartX` `\normalfootstartX{<series>}` is the `<series>` footnote starting macro used in the output routine.

```

4503 \newcommand*{\normalfootstartX}[1]{%
4504   \ifdimequal{Opt}{\prenotesX@}{}%
4505   {%
4506     \iftoggle{prenotesX@}{%
4507       \togglefalse{prenotesX@}%
4508       \skip\csname footins#1\endcsname=%
4509       \glueexpr\csuse{prenotesX@}+\csuse{afterruleX@#1}\relax%
4510     }%
4511   }%
4512 }%
4513 \vskip\skip\csname footins#1\endcsname%
4514 \leftskip=\z@
4515 \rightskip=\z@
4516 \ifl@dpairing\else%
4517   \hsize=\old@hsize%
4518 \fi%
4519 \setnoteswidthliketwocolumnsX@{#1}%
4520 \setnotesXpositionliketwocolumns@{#1}%
4521 \print@footnoterule{#1}%
4522 }%
4523
4524 %

```

`\normalfootnoteruleX` The rule drawn before the footnote series group.

```

4525 \let\normalfootnoteruleX=\footnoterule
4526
4527 %

```

`\normalfootgroupX` `\normalfootgroupX{<series>}` sends the contents of the `<series>` insert box to the output page without alteration.

```

4528 \newcommand*{\normalfootgroupX}[1]{%
4529   \csuse{bhookgroupX@#1}%
4530   \unvbox\@nameuse{footins#1}%
4531   \hsize=\old@hsize%
4532 }%
4533
4534 %

```

`\mpnormalfootgroupX` The minipage version.

```

4535 \newcommand*{\mpnormalfootgroupX}[1]{%
4536   \vskip\skip\@nameuse{mpfootins#1}
4537   \ifl@dpairing\ifparledgroup%
4538     \leavevmode\marks\parledgroup@{begin}%
4539     \marks\parledgroup@series{#1}%
4540     \marks\parledgroup@type{footnoteX}%
4541   \fi\fi\normalcolor
4542   \ifparledgroup%

```



```

4543 \ifl@dpairing%
4544 \else%
4545 \setnoteswidthliketwocolumnsX@{#1}%
4546 \setnotesXpositionliketwocolumns@{#1}%
4547 \print@footnoteXrule{#1}%
4548 \fi%
4549 \else%
4550 \setnoteswidthliketwocolumnsX@{#1}%
4551 \setnotesXpositionliketwocolumns@{#1}%
4552 \print@footnoteXrule{#1}%
4553 \fi%
4554 \csuse{bhookgroupX@{#1}}%
4555 \unvbox\@nameuse{mpfootins#1}}
4556
4557 %

```

**\normalbfnoteX** #1 = footnote series, #2 = footnote content, #3 manual footnote number

```

4558
4559 \newcommand{\normalbfnoteX}[3]{%
4560 \get@thisfootnoteX{#1}%
4561 \ifledRcol%
4562 \ifluatex
4563 \footnotelang@lua[R]%
4564 \fi
4565 \@ifundefined{xpg@main@language}%if polyglossia
4566 {}%
4567 {\footnotelang@poly[R]}%
4568 \xright@appenditem{%
4569 \noexpand\led@set@index@fornote{#1}%
4570 \noexpand\prepare@edindex@fornote{\led@nums}%
4571 \unexpanded{\gdef\@currentnotetype{familiar}}%
4572 \unexpanded{\def\this@footnoteX@reading{\the\csname footnote#1
@reading\endcsname}}%
4573 \noexpand\vbfnoteX{#1}{#2}{\thisc@footnote}{#3}%
4574 \noexpand\led@reinit@index@fornote%
4575 \unexpanded{\advance\@edindex@fornote@m@ne}%
4576 }%
4577 \to\inserts@listR
4578 \global\advance\insert@countR \@ne%
4579 \else%
4580 \ifluatex
4581 \footnotelang@lua%
4582 \fi
4583 \@ifundefined{xpg@main@language}%if polyglossia
4584 {}%
4585 {\footnotelang@poly}%
4586 \xright@appenditem{%
4587 \noexpand\led@set@index@fornote{#1}%
4588 \noexpand\prepare@edindex@fornote{\led@nums}%

```

```

4589     \unexpanded{\gdef\@currentnotetype{familiar}}}%
4590     \unexpanded{\def\this@footnoteX@reading{\the\csname footnote#1
@reading\endcsname}}%
4591     \noexpand\vbfnoteX{#1}{#2}{\thisc@footnote}{#3}%
4592     \noexpand\led@reinit@index@fornote%
4593     \unexpanded{\advance\@edindex@fornote@\m@ne}%
4594   }%
4595       \to\inserts@list
4596   \global\advance\insert@count \@ne%
4597   \fi
4598   \ignorespaces}
4599
4600 %

```

**\get@thisfootnoteX** The macro `\get@thisfootnote` command just saves the footnote number in the `\thisfootnote` macro, depending on the use of pairing environments.

```

4601 \newcommand{\get@thisfootnoteX}[1]{%
4602   \ifboolexpr{bool{l@dpairing} or bool{l@dprintingpages} or bool{
l@dprintingcolumns}}{%
4603     \protected@xdef\thisc@footnote{\the\csname footnote#1@reading\
endcsname}}%
4604   }{%
4605     \protected@xdef\thisc@footnote{\the\csname c@footnote#1\endcsname}}%
4606   }%
4607 }%
4608 %

```

**\vbfnoteX** This command calls the correct footnote-inserting commands. #1 footnote series, #2 footnote content, #3 footnote counter, #4 manual footnote mark

```

4609 \newcommand{\vbfnoteX}[4]{%
4610   \get@fnmarkX{#1}{#3}{#4}\relax%
4611   \@nameuse{regvfootnote#1}{#1}{#2}%
4612 }%
4613 %

```

**\get@fnmarkX** This command gets the correct footnote number when typesetting parallel texts. #1 footnote series, #2 footnote counter, #3 manual footnote number

```

4614 \newcommand{\get@fnmarkX}[3]{%
4615   \ifstrempy{#3}{%
4616     \ifboolexpr{bool{l@dpairing} or bool{l@dprintingpages} or bool{
l@dprintingcolumns}}{%
4617       {%
4618         \stepcounter{footnote#1@typeset}%
4619         \setcounter{footnote#1}{\value{footnote#1@typeset}}%
4620         \@namedef{@thefnmark#1}{\csuse{thefootnote#1}}%
4621         \immediate\write\@mainaux{%

```

```

4622 \csgdef{footnote#1reading#2=typeset}{\the\csname c@footnote#1
@typeset\endcsname}%
4623 }%
4624 }%
4625 {%
4626 \setcounter{footnote#1}{#2}%
4627 \@namedef{@thefnmark#1}{\csuse{thefootnote#1}}%
4628 }%
4629 }%
4630 {%
4631 \csdef{@thefnmark#1}{#3}%
4632 }%
4633 }%
4634 %
4635 %

```

**\vnumfootnoteX** #1 footnote series, #2 footnote content, #3 manual footnote mark

```

4636 \newcommand{\vnumfootnoteX}[3]{%
4637 \ifnumberedpar@
4638 \edtext{}{\normalbfnoteX{#1}{#2}{#3}}%
4639 \else
4640 \def\this@footnoteX@reading{\the\csname footnote#1@reading\endcsname}%
4641 \get@thisfootnoteX{#1}%
4642 \get@fnmarkX{#1}{\expandonce\thisc@footnote}{#3}%
4643 \@nameuse{regvfootnote#1}{#1}{#2}%
4644 \fi}
4645 %
4646 %

```

**arrangementX@normal** \arrangementX@normal{<series>} initialises the settings for the <series> footnotes. This should always be called for each series.

```

4647 \newcommand*{\arrangementX@normal}[1]{%
4648 \csgdef{series@displayX#1}{normal}
4649 \expandafter\let\csname footstart#1\endcsname=\normalfootstartX
4650 \@namedef{@footnotemark#1}{\normal@footnotemarkX{#1}}
4651 \@namedef{bodyfootmark#1}{\normalbodyfootmarkX{#1}}
4652 \expandafter\let\csname regvfootnote#1\endcsname=\normalvfootnoteX
4653 \expandafter\let\csname vfootnote#1\endcsname=\vnumfootnoteX
4654 \expandafter\let\csname footfmt#1\endcsname=\normalfootfmtX
4655 \@namedef{footfootmark#1}{\normalfootfootmarkX{#1}}
4656 \expandafter\let\csname footgroup#1\endcsname=\normalfootgroupX
4657 \expandafter\let\csname footnoterule#1\endcsname=\normalfootnoteruleX
4658 \count\csname footins#1\endcsname=1000
4659 \csxdef{default@footins#1}{1000}%Use to have note only for one side
4660 \dimen\csname footins#1\endcsname=\csuse{maxhnotesX@#1}
4661 \skip\csname footins#1\endcsname=\csuse{beforenotesX@#1}%
4662 \advance\skip\csname footins#1\endcsname by\csuse{afterruleX@#1}%
4663 %

```

Additions for minipages.

```

4664 \ifnoledgroup@else%
4665 \expandafter\let\csname mpvfootnote#1\endcsname=\mpnormalvfootnoteX
4666 \expandafter\let\csname mpfootgroup#1\endcsname=\mpnormalfootgroupX
4667 \count\csname mpfootins#1\endcsname=1000
4668 \dimen\csname mpfootins#1\endcsname=\csuse{maxhnotesX@#1}
4669 \skip\csname mpfootins#1\endcsname=\csuse{beforenotesX@#1}%
4670 \advance\skip\csname mpfootins#1\endcsname by\csuse{afterruleX@#1}%
4671 \fi
4672 }
4673
4674 %

```

#### XIV.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 4675 \newcommand*{\arrangementX@twocol}[1]{%
4676 \csgdef{series@displayX#1}{twocol}
4677 \expandafter\let\csname regvfootnote#1\endcsname=\twocolvfootnoteX
4678 \expandafter\let\csname footfmt#1\endcsname=\twocolfootfmtX
4679 \expandafter\let\csname footgroup#1\endcsname=\twocolfootgroupX
4680 \dimen\csname footins#1\endcsname=\csuse{maxhnotesX@#1}%
4681 \skip\csname footins#1\endcsname=\csuse{beforenotesX@#1}%
4682 \advance\skip\csname footins#1\endcsname by \csuse{afterruleX@#1}\relax%
4683 \twocolfootsetupX{#1}
4684 \ifnoledgroup@else%
4685 \expandafter\let\csname mpvfootnote#1\endcsname=\mpnormalvfootnoteX
4686 \expandafter\let\csname mpfootgroup#1\endcsname=\mptwocolfootgroupX
4687 \skip\csname mpfootins#1\endcsname=\csuse{beforenotesX@#1}%
4688 \advance\skip\csname mpfootins#1\endcsname by\csuse{afterruleX@#1}
4689 \mptwocolfootsetupX{#1}
4690 \fi%
4691 }
4692
4693 %

```

```

\twocolfootsetupX \twocolfootsetupX{<series>}
\mptwocolfootsetupX
4694 \newcommand*{\twocolfootsetupX}[1]{%
4695 \count\csname footins#1\endcsname 500
4696 \csxdef{default@footins#1}{500}%Use this to confine the notes to one
side only
4697 \multiply\dimen\csname footins#1\endcsname by \tw@}
4698 \newcommand*{\mptwocolfootsetupX}[1]{%
4699 \count\csname mpfootins#1\endcsname 500
4700 \multiply\dimen\csname mpfootins#1\endcsname by \tw@}

```

```

4701
4702 %
\twocolvfootnoteX \twocolvfootnoteX{<series>}
4703 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\twocolvfootnoteX}[2]{%
4704 \beforeinsertion@X{#1}%
4705 \let\bidir@RTL@everypar\relax%
4706 \insert\csname footins#1\endcsname\bgroup%
4707 \hspace=\expandafter\dimexpr\csuse{widthX@#1}\relax%
4708 \noindent\csuse{bhooknoteX@#1}%
4709 \csuse{notefontsizeX@#1}%
4710 \footssplitsskip%
4711 \spaceskip=\z@skip \xspaceskip=\z@skip%
4712 \@nameuse{footfmt#1}{#1}{#2}\egroup}
4713
4714 %

```

```

\twocolfootfmtX \twocolfootfmtX{<series>}
4715 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\twocolfootfmtX}[2]{%
4716 \protected@edef\@currentlabel{%
4717 \nameuse{@thefnmark#1}%
4718 }%
4719 \normal@pars%
4720 \sethangindentX{#1}%
4721 \hspace \csuse{hsizetwocolX@#1}%
4722 \setparindentX{#1}%
4723 \tolerance=5000\relax%
4724 \par%
4725 \@tempdima=\parindent%
4726 \csuse{colalignX@#1}%
4727 \parindent=\@tempdima%
4728 {\hspace{\parindent}%
4729 \csuse{notenumbfontX@#1}\wrapped@footfootmarkX{#1}\strut%
4730 \csuse{wrapcontentX@#1}{#2}%
4731 \strut\par}%
4732 \allowbreak%
4733 }%
4734
4735 %

```

```

\twocolfootgroupX \twocolfootgroupX{<series>}
\mptwocolfootgroupX
4736 \newcommand*{\twocolfootgroupX}[1]{\csuse{bhookgroupX@#1}\csuse{
notefontsizeX@#1}
4737 \splittopskip=\ht\strutbox
4738 \expandafter
4739 \rigidbalanceX\csname footins#1\endcsname \tw@ \splittopskip}}
4740

```

```

4741 \newcommand*\mptwocolfootgroupX}[1]{%
4742   \vskip\skip\@nameuse{mpfootins#1}
4743   \ifl@dpairing\ifparledgroup%
4744     \leavevmode\marks\parledgroup@{begin}%
4745     \marks\parledgroup@series{#1}%
4746     \marks\parledgroup@type{footnoteX}%
4747   \fi\fi\normalcolor
4748   \ifparledgroup%
4749     \ifl@dpairing%
4750     \else%
4751       \setnoteswidthliketwocolumnsX@{#1}%
4752       \setnotesXpositionliketwocolumns@{#1}%
4753       \print@footnoteXrule{#1}%
4754     \fi%
4755   \else%
4756     \setnoteswidthliketwocolumnsX@{#1}%
4757     \setnotesXpositionliketwocolumns@{#1}%
4758     \print@footnoteXrule{#1}%
4759   \fi%
4760   \csuse{bhookgroupX@#1}%
4761   \splittopskip=\ht\strutbox
4762   \expandafter
4763   \rigidbalanceX\csname mpfootins#1\endcsname \tw@ \splittopskip}}
4764
4765 %

```

#### XIV.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 4766 \newcommand*\arrangementX@threecol}[1]{%
4767   \csgdef{series@displayX#1}{threecol}
4768   \expandafter\let\csname regvfootnote#1\endcsname=\threecolvfootnoteX
4769   \expandafter\let\csname footfmt#1\endcsname=\threecolfootfmtX
4770   \expandafter\let\csname footgroup#1\endcsname=\threecolfootgroupX
4771   \dimen\csname footins#1\endcsname=\csuse{maxhnotesX@#1}%
4772   \skip\csname footins#1\endcsname=\csuse{beforenotesX@#1}%
4773   \advance\skip\csname footins#1\endcsname by \csuse{afterruleX@#1}\relax%
4774   \threecolfootsetupX{#1}
4775   \ifnoledgroup@else%
4776     \expandafter\let\csname mpvfootnote#1\endcsname=\mpnormalvfootnoteX
4777     \expandafter\let\csname mpfootgroup#1\endcsname=\mpthreecolfootgroupX
4778     \skip\csname mpfootins#1\endcsname=\csuse{beforenotesX@#1}%
4779     \advance\skip\csname mpfootins#1\endcsname by \csuse{afterruleX@#1}
4780     \mpthreecolfootsetupX{#1}
4781   \fi%
4782 }
4783

```

```

4784 %

\threecolfootsetupX \threecolfootsetupX{<series>}
\mpthreecolfootsetupX
4785 \newcommand*{\threecolfootsetupX}[1]{%
4786 \count\csname footins#1\endcsname 333
4787 \csxdef{default@footins#1}{333}%Use this to confine the notes to one
side only
4788 \multiply\dimen\csname footins#1\endcsname by \thr@@}
4789 \newcommand*{\mpthreecolfootsetupX}[1]{%
4790 \count\csname mpfootins#1\endcsname 333
4791 \multiply\dimen\csname mpfootins#1\endcsname by \thr@@}
4792 %
4793 %

\threecolvfootnoteX \threecolvfootnoteX{<series>}{<text>}

4794 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\threecolvfootnoteX}[2]{
%
4795 \let\bidir@RTL@everypar\relax%
4796 \beforeinsertion@X{#1}%
4797 \insert\csname footins#1\endcsname\bgroup%
4798 \hspace=\expandafter\dimexpr\csuse{widthX@#1}\relax%
4799 \noindent\csuse{hooknoteX@#1}%
4800 \csuse{notefontsizeX@#1}%
4801 \footssplitsskip%
4802 \@nameuse{footfmt#1}{#1}{#2}\egroup}
4803 %
4804 %

\threecolfootfmtX \threecolfootfmtX{<series>}

4805 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\threecolfootfmtX}[2]{%
4806 \protected@edef\@currentlabel{%
4807 \@nameuse{@thefnmark#1}%
4808 }%
4809 \sethangindentX{#1}%
4810 \normal@pars%
4811 \hspace \csuse{hsizethreecolX@#1}%
4812 \setparindentX{#1}%
4813 \tolerance=5000\relax%
4814 \@tempdima=\parindent%
4815 \csuse{colalignX@#1}%
4816 \parindent=\@tempdima%
4817 {\hspace{\parindent}%
4818 \csuse{notenumfontX@#1}\wrapped@footfootmarkX{#1}\strut%
4819 \csuse{wrapcontentX@#1}{#2}%
4820 \strut\par}\allowbreak}
4821 %
4822 %

```

```

\threecolfootgroupX \threecolfootgroupX{<series>}
\mpthreecolfootgroupX
4823 \newcommand*{\threecolfootgroupX}[1]{\csuse{bhookgroupX@#1}\csuse{
notefontsizeX@#1}
4824 \splittopskip=\ht\strutbox
4825 \expandafter
4826 \rigidbalanceX\csname footins#1\endcsname \thr@@ \splittopskip}}
4827
4828 \newcommand*{\mpthreecolfootgroupX}[1]{\%
4829 \vskip\skip\@nameuse{mpfootins#1}
4830 \ifl@dpairing\ifparledgroup
4831 \leavevmode\marks\parledgroup@{begin}%
4832 \marks\parledgroup@series{#1}%
4833 \marks\parledgroup@type{footnoteX}%
4834 \fi\fi\normalcolor
4835 \ifparledgroup%
4836 \ifl@dpairing%
4837 \else%
4838 \setnoteswidthliketwocolumnsX@{#1}%
4839 \setnotesXpositionliketwocolumns@{#1}%
4840 \print@footnoteXrule{#1}%
4841 \fi%
4842 \else%
4843 \setnoteswidthliketwocolumnsX@{#1}%
4844 \setnotesXpositionliketwocolumns@{#1}%
4845 \print@footnoteXrule{#1}%
4846 \fi%
4847 \csuse{bhookgroupX@#1}%
4848 \splittopskip=\ht\strutbox
4849 \expandafter
4850 \rigidbalanceX\csname mpfootins#1\endcsname \thr@@ \splittopskip}}
4851
4852 %

```

#### XIV.4.5 Paragraphed footnotes

The following macros set footnotes as one paragraph.

```

\arrangementX@paragraph \footparagraphX{<series>}
4853 \newcommand*{\arrangementX@paragraph}[1]{\%
4854 \csgdef{series@displayX#1}{paragraph}%
4855 \expandafter\let\csname footstart#1\endcsname=\parafootstartX
4856 \expandafter\let\csname regvfootnote#1\endcsname=\para@vfootnoteX
4857 \expandafter\let\csname footfmt#1\endcsname=\parafootfmtX
4858 \expandafter\let\csname footgroup#1\endcsname=\para@footgroupX
4859 \expandafter\let\csname footnoterule#1\endcsname=\normalfootnoteruleX
4860 \count\csname footins#1\endcsname=1000
4861 \csxdef{default@footins#1}{1000}%Use this to confine the notes to one
side only

```



```

4862 \dimen\csname footins#1\endcsname=\csuse{maxhnotesX@#1}
4863 \skip\csname footins#1\endcsname=\csuse{beforenotesX@#1}%
4864 \advance\skip\csname footins#1\endcsname by\csuse{afterruleX@#1}%
4865 \para@footsetupX{#1}
4866 \ifnoledgroup@else
4867   \expandafter\let\csname mpvfootnote#1\endcsname=\mppara@vfootnoteX
4868   \expandafter\let\csname mpfootgroup#1\endcsname=\mppara@footgroupX
4869   \count\csname mpfootins#1\endcsname=1000
4870   \dimen\csname mpfootins#1\endcsname=\csuse{maxhnotesX@#1}
4871   \skip\csname mpfootins#1\endcsname=\csuse{beforenotesX@#1}%
4872   \advance\skip\csname mpfootins#1\endcsname by\csuse{afterruleX@#1}%
4873 \fi
4874 }
4875
4876 %

```

`\para@footsetupX` `\para@footsetupX{<series>}`

```

4877 \newcommand*{\para@footsetupX}[1]{\csuse{bhookgroupX@#1}\csuse{
notefontsizeX@#1}
4878 \setnoteswidthliketwocolumnsX@#1}%
4879 \ifcsempy{widthX@#1}%
4880 {}%
4881 {\columnwidth=\expandafter\dimexpr\csuse{widthX@#1}\relax}%
4882 \dimen0=\baselineskip
4883 \multiply\dimen0 by 1024
4884 \divide\dimen0 by \columnwidth \multiply\dimen0 by \footfudgefiddle\relax
4885 %
4886 \expandafter
4887 \xdef\csname footfudgefactor#1\endcsname{%
4888   \expandafter\strip@pt\dimen0 }}
4889 %

```

`\parafootstartX` `\parafootstartX{<series>}`

```

4890 \newcommand*{\parafootstartX}[1]{%
4891   \ifdimequal{Opt}{\prenotesX@}{}%
4892   {%
4893     \iftoggle{prenotesX@}{%
4894       \togglefalse{prenotesX@}%
4895       \skip\csname footins#1\endcsname=%
4896       \glueexpr\csuse{prenotesX@}+\csuse{afterruleX@#1}\relax%
4897     }%
4898     {}%
4899   }%
4900 \leftskip=\z@
4901 \rightskip=\z@
4902 \setparindentX{#1}%
4903 \vskip\skip\@nameuse{footins#1}%

```

```

4904 \setnoteswidthliketwocolumnsX@{#1}%
4905 \setnotesXpositionliketwocolumns@{#1}%
4906 \print@footnoteXrule{#1}%
4907 }
4908
4909 %

```

```

\para@vfootnoteX \para@vfootnoteX{<series>}{<text>}
\mppara@vfootnoteX
4910 \newcommand*{\para@vfootnoteX}[2]{%
4911 \csuse{beforeinsertingX@#1}%
4912 \insert\csname footins#1\endcsname%
4913 \bgroup
4914 \csuse{notefontsizeX@#1}
4915 \footsplitskips
4916 \setbox0=\vbox{\hsize=\maxdimen%
4917 \let\bidir@RTL@everypar\@empty%
4918 \insert@txtbeforenotesX{#1}%
4919 \noindent\csuse{bhooknoteX@#1}%
4920 \@nameuse{footfmt#1}{#1}{#2}}%
4921 \setbox0=\hbox{\unvXH{0}{#1}}%
4922 \dp0=\z@
4923 \ht0=\csname footfudgefactor#1\endcsname\wd0
4924 \box0
4925 \penalty0
4926 \egroup}
4927 \newcommand*{\mppara@vfootnoteX}[3]{%
4928 \get@thisfootnoteX{#1}%
4929 \get@fnmarkX{#1}{\thisc@footnote}{#3}%
4930 \edef\this@footnoteX@reading{\the\csname footnote#1@reading\endcsname}%
4931 \global\setbox\@nameuse{mpfootins#1}\vbox{%
4932 \unvbox\@nameuse{mpfootins#1}
4933 \csuse{notefontsizeX@#1}
4934 \footsplitskips
4935 \setbox0=\vbox{\hsize=\maxdimen%
4936 \let\bidir@RTL@everypar\@empty%
4937 \noindent\color@begingroup%
4938 \csuse{bhooknoteX@#1}%
4939 \@nameuse{footfmt#1}{#1}{#2}\color@endgroup}%
4940 \setbox0=\hbox{\unvXH{0}{#1}}%
4941 \dp0=\z@
4942 \ht0=\csname footfudgefactor#1\endcsname\wd0
4943 \box0
4944 \penalty0}}
4945
4946 %

```

```

\unvXH_{497 \newcommand*{\unvXH}[2]{% 2th is optional for retro-compatibility
4948 \setbox0=\vbox{\unvbox#1%

```

```

4949 \global\setbox1=\lastbox}%
4950 \unhbox1
4951 \unskip           % remove \rightskip,
4952 \unskip           % remove \parfillskip,
4953 \unpenalty        % remove \penalty of 10000,
4954 \hskip\csuse{afternoteX@#2}%
4955 \relax}% but add the glue to go between the notes
4956
4957 %

```

`\parafootfmtX` `\parafootfmtX{<series>}`

```

4958 \newcommand*{\parafootfmtX}[2]{%
4959   \protected@edef\@currentlabel{%
4960     \@nameuse{@thefnmark#1}%
4961   }%
4962   \insertparafootsepX{#1}%
4963   \ledsetnormalparstuff@common%
4964   {\csuse{notenumfontX@#1}%
4965    \csuse{notenumfontX@#1}%
4966    \wrapped@footfootmarkX{#1}%
4967    \strut%
4968    \csuse{wrapcontentX@#1}{#2}%
4969    \penalty-10}}
4970
4971 %

```

`\para@footgroupX` `\para@footgroupX{<series>}`  
`\mppara@footgroupX`

```

4972 \newcommand*{\para@footgroupX}[1]{%
4973   \hsize=\expandafter\dimexpr\csuse{widthX@#1}\relax%
4974   \unvbox\csname footins#1\endcsname
4975   \ifcsstring{raggedX@#1}{L}{\RaggedLeft}{}%
4976   \ifcsstring{raggedX@#1}{R}{\RaggedRight}{}%
4977   \makehboxofhboxes
4978   \setbox0=\hbox{\unhbox0 \removehboxes}%
4979   \csuse{bhookgroupX@#1}
4980   \csuse{notefontsizeX@#1}
4981   \unhbox0\par}
4982
4983 \newcommand*{\mppara@footgroupX}[1]{%
4984   \setnoteswidthliketwocolumnsX@{#1}%
4985   \vskip\skip\@nameuse{mpfootins#1}
4986   \ifl@dpairing\ifparledgroup
4987     \leavevmode%
4988     \leavevmode\marks\parledgroup@{begin}%
4989     \marks\parledgroup@series{#1}%
4990     \marks\parledgroup@type{footnoteX}%
4991     \fi\fi\normalcolor
4992   \ifparledgroup%

```

```

4993 \ifl@pairing%
4994 \else%
4995 \setnoteswidthliketwocolumnsX@{#1}%
4996 \setnotesXpositionliketwocolumns@{#1}%
4997 \print@footnoteXrule{#1}%
4998 \fi%
4999 \else%
5000 \setnoteswidthliketwocolumnsX@{#1}%
5001 \setnotesXpositionliketwocolumns@{#1}%
5002 \print@footnoteXrule{#1}%
5003 \fi%
5004 \unvbox\csname mpfootins#1\endcsname
5005 \ifcsstring{raggedX@#1}{L}{\RaggedLeft}{}%
5006 \ifcsstring{raggedX@#1}{R}{\RaggedRight}{}%
5007 \makehboxofhboxes
5008 \setbox0=\hbox{\unhbox0 \removehboxes}%
5009 \csuse{bhookgroupX@#1}%
5010 \csuse{notefontsizeX@#1}%
5011 \setparindentX{#1}%
5012 \unhbox0\par}}
5013
5014 %

```

**Insertion of the footnotes separator** The command `\insertparafootsepX{<series>}` must be called at the beginning of `\parafootftmX`.

```

\insertparafootsepX15 \newcommand{\insertparafootsepX}[1]{%
5016 \ifl@Rcol@%
5017 \ifnumequal{\csuse{prevpage#1@numR}}{\page@numR}%
5018 {\csuse{parafootsepX@#1}}%
5019 {}%
5020 \global\csname prevpage#1@numR\endcsname=\page@numR%
5021 \else%
5022 \ifnumbering%
5023 \ifnumequal{\csuse{prevpage#1@num}}{\page@num}%
5024 {\csuse{parafootsepX@#1}}%
5025 {}%
5026 \global\csname prevpage#1@num\endcsname=\page@num%
5027 \else%We can't use directly the page counter, see https://tex.
stackexchange.com/a/239457/7712
5028 \global\expandafter\advance\csname footnote#1@parafootsepX\endcsname
by 1%
5029 \edef\@tmpa{parafootsepX@#1@the\csname footnote#1@parafootsepX\
endcsname}%
5030 \expandafter\edlabel\expandafter{\@tmpa}%
5031 \edef\@tmpb{\expandafter\pageref\expandafter{\@tmpa}}%
5032 \IfStrEq{\csuse{prevpage#1@num}}{\@tmpb}%
5033 {\csuse{parafootsepX@#1}}%
5034 {}%

```

```

5035 \global\cslet{prevpage#1@num}{\@tmpb}%
5036 \fi%
5037 \fi%
5038 }
5039 %

```

## XIV.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.

```

5040 \newcommand{\wrapped@footfootmarkX}[1]{%
5041 \ifdefined\hypertarget%
5042 \hyperlink%
5043 { @bodyfootmark#1@this@footnoteX@reading}%
5044 { \@nameuse{footfootmark#1}}%
5045 \Hy@raisedlink{%
5046 \hypertarget%
5047 { @footnotemark#1@this@footnoteX@reading}%
5048 {}%
5049 }%
5050 \else%
5051 \@nameuse{footfootmark#1}%
5052 \fi%
5053 }%
5054 %

```

**\wrapped@bodyfootmarkX** \wrapped@bodyfootmarkX prints the footnote mark of the text body, wrapped in hyperref package's commands, if needed.

```

5055 \newcommand{\wrapped@bodyfootmarkX}[1]{%
5056 \ifdefined\hypertarget%
5057 \hyperlink%
5058 { @footnotemark#1@expandafter\the\csname footnote#1@reading\
5059 endcsname}%
5060 { \@nameuse{bodyfootmark#1}}%
5061 \Hy@raisedlink{%
5062 \hypertarget%
5063 { @bodyfootmark#1@expandafter\the\csname footnote#1@reading\
5064 endcsname}%
5065 {}%
5066 }%
5067 \else%
5068 \@nameuse{bodyfootmark#1}%
5069 \fi%
5070 }%
5071 %

```

## XV 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 Lua<sup>®</sup>TeX.

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%70 \newcommand*{\ledsetnormalparstuff@common}{%
\Xledsetnormalparstuff%71 \ifluatex%
\ledsetnormalparstuffX%72 \ifdefstring{\footnote@luatextextdir}{TLT}{}%
5073 {\textdir\footnote@luatextextdir}%
5074 \pardir\footnote@luatexpardir%
5075 \fi%
5076 \csuse{\csuse{footnote@dir}}%
5077 \normal@pars%
5078 \parfillskip \z@ \@plus 1fil}%
5079
5080 \newcommand*{\Xledsetnormalparstuff}[1]{%
5081 \ledsetnormalparstuff@common%
5082 \Xsetparindent{#1}%
5083 \iftoggle{Xparindent@#1}{\hspace{\parindent}}{}%
5084 }%
5085
5086 \newcommand*{\ledsetnormalparstuffX}[1]{%
5087 \ledsetnormalparstuff@common%
5088 \setparindentX{#1}%
5089 \iftoggle{parindentX@#1}{\hspace{\parindent}}{}%
5090 }%
5091 %

```

## XVI Footnotes' width for two columns

We define here some commands which make sense only with `reledpar`, but must be called when defining notes parameters. These commands change the width of block notes to allow them to have the same size than two parallel columns.

`\old@hsize` These two commands are called at the beginning of critical or familiar notes groups.  
`\setXnoteswidthliketwocolumns@` They set, if the option is enabled, the `\hsize`. They are also called at the on the setup  
`\setnoteswidthliketwocolumnsX@` for paragraphed notes.

```

5092
5093 \newdimen\old@hsize%

```

```

5094 \AtBeginDocument{\old@hsize=\hsize}%
5095
5096 \newcommand{\setXnoteswidthliketwocolumns@}[1]{%
5097   \global\let\hsize@fornote=\hsize%
5098   \global\old@hsize=\hsize%
5099   \let\old@columnwidth=\columnwidth%
5100   \iftoggle{Xnoteswidthliketwocolumns@#1}{%
5101     {%
5102       \setwidthliketwocolumns%
5103       \global\let\hsize@fornote=\hsize%
5104     }%
5105     {}%
5106   \let\hsize=\hsize@fornote%
5107   \let\columnwidth=\old@columnwidth%
5108 }%
5109
5110 \newcommand{\setnoteswidthliketwocolumnsX@}[1]{%
5111   \global\let\hsize@fornote=\hsize%
5112   \global\old@hsize=\hsize%
5113   \let\old@columnwidth=\columnwidth%
5114   \iftoggle{noteswidthliketwocolumnsX@#1}{%
5115     {%
5116       \setwidthliketwocolumns%
5117       \global\let\hsize@fornote=\hsize%
5118     }%
5119     {}%
5120   \let\hsize=\hsize@fornote%
5121   \let\columnwidth=\old@columnwidth%
5122 }%
5123
5124 %

```

`\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`.

```

5125 \newcommand{\setXnotespositionliketwocolumns@}[1]{%
5126   \iftoggle{Xnoteswidthliketwocolumns@#1}{%
5127     \csuse{setnotespositionliketwocolumns@\columns@position}%
5128   }{}%
5129 }%
5130
5131 \newcommand{\setnotesXpositionliketwocolumns@}[1]{%
5132   \iftoggle{noteswidthliketwocolumnsX@#1}{%
5133     \csuse{setnotespositionliketwocolumns@\columns@position}%
5134   }{}%
5135 }%
5136
5137 %

```

## XVII Footnotes' order

`\fnpos` The `\fnpos` and `\mpfnpos` simply place their arguments in `\@fnpos` and `\@mpfnpos`, which will be used later in the output routine.

```

5138 \def\fnpos{familiar-critical}
5139 \def\mpfnpos{critical-familiar}
5140 \newcommand{\fnpos}[1]{\xdef\@fnpos{#1}}
5141 \newcommand{\mpfnpos}[1]{\xdef\@mpfnpos{#1}}
5142 %

```

## XVIII 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@Xfootnoterule43 \newcommand{\print@Xfootnoterule}[1]{%
\print@footnoterule44 \vskip-\csuse{Xafterrule@#1}%Because count in \dimen\csuse{#1footins}
5145 \nointerlineskip%
5146 \moveleft-\leftskip\vbox{\csuse{#1footnoterule}}%
5147 \nointerlineskip%
5148 \vskip\csuse{Xafterrule@#1}%
5149 }%
5150
5151 \newcommand{\print@footnoterule}[1]{%
5152 \vskip-\csuse{afterruleX@#1}%Because count in \dimen\csuse{footins#1}
5153 \nointerlineskip%
5154 \moveleft-\leftskip\vbox{\csuse{footnoterule#1}}%
5155 \nointerlineskip%
5156 \vskip\csuse{afterruleX@#1}%
5157 }%
5158
5159 %

```

## XIX Specific skip for first series of footnotes

### XIX.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 !

## XIX.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.

```

5160 \newtoggle{Xprenotes@}%
5161 \toggletrue{Xprenotes@}%
5162 \newcommand{Xprenotes@}{Opt}%
5163 \newcommand*{\Xprenotes}[1]{\renewcommand{\Xprenotes@}{#1}}%
5164 \newcommand{\preXnotes}[1]{\led@warning@preXnotes@deprecated\Xprenotes{#1}}
    %For compatibility
5165 %

```

The same, but for familiar footnotes.

```

\Xprenotes 66 \newtoggle{prenotesX@}
\Xprenotes 67 \toggletrue{prenotesX@}
5168 \newcommand{prenotesX@}{Opt}
5169 \newcommand*{\prenotesX}[1]{\renewcommand{\prenotesX@}{#1}}
5170 %

```

## XIX.3 Internal commands

```

firstXseries@ 71 \gdef\firstXseries@{}
prepare@Xprenotes 72 \newcommand{\prepare@Xprenotes}[1]{%
5173   \ifdimequal{Opt}{\Xprenotes@}%

```

```

5174 {}%
5175 {%
5176   \IfStrEq{\firstXseries@}{}{}%
5177   \global\skip\csuse{#1footins}=\Xprenotes@%
5178   \global\advance\skip\csname #1footins\endcsname by\csuse{Xafterterrule@
#1}%
5179   \gdef\firstXseries@{#1}%
5180 }%
5181 {%
5182   \ifseriesbefore{#1}{\firstXseries@}%
5183   {%
5184     \global\skip\csuse{#1footins}=\csuse{Xbeforenotes@\firstXseries@}%
5185     \global\advance\skip\csname #1footins\endcsname by\csuse{Xafterterrule@
#1}%
5186     \gdef\firstXseries@{#1}%
5187   }%
5188   {}%
5189 }%
5190 }%
5191 }
5192 %

```

The same thing is required for familiar notes and \prenotesX.

```

firstseriesX@93 \gdef\firstseriesX@{}
prepare@prenotesX94 \newcommand{\prepare@prenotesX}[1]{%
5195   \ifdimequal{0pt}{\prenotesX@}%
5196   {}%
5197   {%
5198     \IfStrEq{\firstseriesX@}{}{}%
5199     \global\skip\csuse{footins#1}=\prenotesX@%
5200     \global\advance\skip\csname footins#1\endcsname by\csuse{afterterruleX@
#1}%
5201     \gdef\firstseriesX@{#1}%
5202   }%
5203   {%
5204     \ifseriesbefore{#1}{\firstseriesX@}%
5205     {%
5206       \global\skip\csuse{footins#1}=\csuse{beforenotesX@\firstseriesX@}%
5207       \global\advance\skip\csname footins#1\endcsname by\csuse{afterterruleX@
#1}%
5208       \gdef\firstXseries@{#1}%
5209     }%
5210     {}%
5211   }%
5212 }%
5213 }
5214 %

```

## XX Endnotes

First, check the noend option.

```
5215 \ifbool{noend@}{}{%Used instead of \ifnoend@ to prevent expansion problem
5216 %
```

### XX.1 Internal commands

`\l@dend@open` `\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.

```
5217 \newcommand{\l@dend@open}[1]{%
5218   \global\booltrue{l@dend@#1}%
5219   \expandafter\immediate%
5220   \expandafter\openout%
5221   \csname l@d@#1end\endcsname%
5222   =\l@auxdir\jobname.#1end\relax%
5223 }%
5224 \newcommand{\l@dend@close}[1]{%
5225   \global\boolfalse{l@dend@#1}%
5226   \expandafter\immediate%
5227   \expandafter\closeout\csname l@d@#1end\endcsname%
5228 }%
5229
5230 %
```

`\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.

```
5231 \newcommand{\l@dend@stuff}{%
5232   \def\do##1{%
5233     \ifbool{l@dend@##1}{}%
5234     {\l@dend@open{##1}}%
5235     \expandafter\immediate\expandafter\write\csname l@d@##1end\endcsname{\
string\l@d@section{\the\section@num}\@percentchar}%
5236   }%
5237   \dolistloop{@series}%
5238 }%
5239
5240 %
```

`\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:

Ar-

- #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.

```

5241 \global\newbool{parapparatus@#4}\def\endprint#1#2#3#4#5#6#7{
5242 \iftoggle{Xendfirstnote@#4}{%
5243 \global\togglefalse{Xendfirstnote@#4}%
5244 \csuse{Xendtxtbeforenotes@#4}%
5245 }{}%
5246 \csuse{Xendbhooknote@#4}%
5247 \csuse{Xendnotefontsize@#4}%
5248 \hangindent=\csuse{Xendhangindent@#4}%
5249 \Xendstorelineinfo{#1}{#4}%
5250 \ifXendinsertsep%
5251 \hskip\csuse{Xendafternote@#4}\relax%
5252 \ifcsdef{prevendline#4}%
5253 {\ifcsequal{prevendline#4}{lineinfo@}%
5254 {\ifcsequal{Xendsynlinenum@#4}{\csuse{Xendsep@#4}}{}}%
5255 {\csuse{Xendsep@#4}}%
5256 }%
5257 {\csuse{Xendsep@#4}}%
5258 \else%
5259 \iftoggle{Xendparagraph@#4}%
5260 {\global\Xendinsertsep@true}%
5261 {}%
5262 \fi%
5263 \xdef\@currentseries{#4}%
5264 \def\do##1{%
5265 \setkeys{mac}{truefootnoteoption}{##1}%
5266 }%
5267 \notblank{#5}{\docsvlist{#5}}{}%
5268 \IfStrEq{#6}{R}{\ledRcol@true}{}%
5269 \def\@this@crossref@start{#7:start}%
5270 \def\@this@crossref@end{#7:end}%
5271 \printlineendnote{#1}{#4}%
5272 \IfStrEq{#6}{R}{\ledRcol@false}{}%
5273 \nottoggle{Xendlemmadisablefontselection@#4}%
5274 {\select@lemmafont#1}%

```

```

5275 {}%
5276 \bgroup%
5277 \csuse{Xendlemmafont@#4}%
5278 \csuse{Xendwraplemma@#4}{#2}%
5279 \egroup%
5280 \ifboolexpr{%
5281   togl {nosep@}%
5282   or test{\ifcsemtyp{Xendlemmaseparator@#4}}%
5283 }%
5284 {\hskip\csuse{Xendinplaceoflemmaseparator@#4}\relax}%
5285 {\nobreak%
5286 \hskip\csuse{Xendbeforelemmaseparator@#4}%
5287 \csuse{Xendlemmaseparator@#4}%
5288 \hskip\csuse{Xendafterlemmaseparator@#4}%
5289 \relax%
5290 }%
5291 \csuse{Xendwrapcontent@#4}{#3}%
5292 \nottoggle{Xendparagraph@#4}{\par}{}%
5293 \def\do##1{%
5294   \setkeys[mac]{falsefootnoteoption}{##1}%
5295 }%
5296 \notblank{#5}{\docsvlist{#5}}{%}%
5297 }}%
5298
5299 \let\l@d@section=\@gobble
5300
5301 %

```

**\Xendstorelineinfo** The `\Xendstorelineinfo` macro is used to store some data about line number of the current critical endnote, data which will be reused later for the `\Xnumberonlyfirstinline` and related setting.

#1 endnote specification for the current endnote; #2 endnote series.

```

5302 \newcommand{\Xendstorelineinfo}[2]{%
5303   \l@dp@rsefootspec#1|%
5304   \iftoggle{Xendnumberonlyfirstintwolines@#2}{%
5305     \xdef\lineinfo@{\l@dparsedstartpage - \l@dparsedstartline - \
5306       \l@dparsedstartsub - \l@dparsedendpage - \l@dparsedendline - \
5307       \l@dparsedendsub}%
5308   }%
5309   {\
5310     \xdef\lineinfo@{\l@dparsedstartpage - \l@dparsedstartline - \
5311       \l@dparsedstartsub}%
5312   }%
5313 }%
5314 %

```

**\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.

```

5312 \newcommand{\printlineendnote}[2]{%
5313   \ifbool{expr{%
5314     togl {nonum@}%
5315     or togl {Xendnonumber@#2}%
5316   }}%
5317   {\hspace{\csuse{Xendinplaceofnumber@#2}}}%
5318   {%
5319     \iftoggle{Xendnumberonlyfirstinline@#2}%
5320     {\ifcndef{prevendline#2}%
5321       {\ifcsequal{prevendline#2}{lineinfo@}%
5322         {%
5323           \csuse{Xendbhookinplaceofnumber@#2}%
5324           \ifcempty{Xendsymmlinenum@#2}%
5325             {\hspace{\csuse{Xendinplaceofnumber@#2}}}%
5326             {\printsymlineendnotearea{#2}}%
5327           \csuse{Xendahookinplaceofnumber@#2}%
5328         }%
5329         {\printlineendnotearea{#1}{#2}}}%
5330       {\printlineendnotearea{#1}{#2}}}%
5331     }%
5332     {\printlineendnotearea{#1}{#2}}}%We keep every time line
5333   \csxdef{prevendline#2}{\lineinfo@}%
5334   }%
5335 }%
5336 %

```

```

\printsymlineendnotearea \newcommand{\printsymlineendnotearea}[1]{%
5338   \hspace{\csuse{Xendbeforemsymmlinenum@#1}}%
5339   \csuse{Xendnotenumfont@#1}%
5340   \ifdimequal{\csuse{Xendboxsymmlinenum@#1}}{\z@}%
5341     {\csuse{Xendsymmlinenum@#1}}%
5342     {\hbox to \csuse{Xendboxsymmlinenum@#1}%
5343       {\csuse{Xendsymmlinenum@#1}\hfill}%
5344     }%
5345   \hspace{\csuse{Xendaftersymmlinenum@#1}}%
5346 }%
5347 %

```

**\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.)

```

5348 \newcommand{\printlineendnotearea}[2]{%
5349   \csuse{Xendbhooklinenumber@#2}%
5350   \hspace{\csuse{Xendbeforenumber@#2}}%
5351   \bgroup%
5352   \csuse{Xendnotenumfont@#2}%
5353   \ifdimequal{\csuse{Xendboxlinenumber@#2}}{\Opt}%

```

```

5354     {\printendlines#1||\ifledRcol@\@Rlineflag\fi}%
5355     {\leavevmode%
5356       \hbox to \csuse{Xendboxlinenum@#2}%
5357       {%
5358         \IfSubStr{RC}{\csuse{Xendboxlinenumalign@#2}}{\hfill}{}}%
5359         \printendlines#1||\ifledRcol@\@Rlineflag\fi%
5360         \IfSubStr{LC}{\csuse{Xendboxlinenumalign@#2}}{\hfill}{}}%
5361       }%
5362     \egroup%
5363     \hspace{\csuse{Xendafternumber@#2}}%
5364     \csuse{Xendahooklinenum@#2}%
5365   }%
5366   %

```

## XX.2 User level commands

### XX.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.

```

\toendnotes67 \newcommandx{\toendnotes}[2][1,usedefault]{%
\toendnotes*68 \ifboolexpr{bool{numbering} or bool{numberingR}}{%
5369   \def\do##1{%
5370     \expandafter\immediate\expandafter\write\csname l@d@##1end\endcsname%
5371     {\unexpanded{#2}\@percentchar}%
5372   }%
5373   \ifstrempy{#1}%
5374     {\dolistloop{\@series}}%
5375     {\docsvlist{#1}}%
5376   }\led@err@toendnotes@outsidenumbering}%
5377 }%
5378 \WithSuffix\newcommandx\toendnotes*[2][1,usedefault]{%
5379 \ifboolexpr{bool{numbering} or bool{numberingR}}{%
5380   \def\do##1{%
5381     \expandafter\immediate\expandafter\write\csname l@d@##1end\endcsname%
5382     {#2\@percentchar}%
5383   }%
5384   \ifstrempy{#1}%
5385     {\dolistloop{\@series}}%
5386     {\docsvlist{#1}}%
5387   }\led@err@toendnotes@outsidenumbering}%
5388 }%
5389 %

```

### XX.2.2 Printing endnotes

`\doendnotes` `\doendnotes` is the command you use to print one series of endnotes; it takes one argument: the series letter of the note series you want to print. `\Xendinsertsep@` is set to true at the first note of the series, and to false at the last one.

```

5390 \newif\ifXendinsertsep@%
5391 \newcommand*{\doendnotes}[1]{%
5392   \l@dend@close{#1}%
5393   \begingroup
5394     \csxdef{prevpagenum@#1}{}%
5395     \csxdef{prevpagerange@#1}{}%
5396     \global\toggletrue{Xendfirstnote@#1}%
5397     \makeatletter
5398     \expandafter\let\csname #1end\endcsname=\endprint
5399     \input\l@auxdir\jobname.#1end%
5400     \global\Xendinsertsep@false%
5401   \endgroup}
5402 %

```

`\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.

```

5403 \newcommand*{\doendnotesbysection}[1]{%
5404   \l@dend@close{#1}%
5405   \csxdef{prevpagenum@#1}{}%
5406   \csxdef{prevpagerange@#1}{}%
5407   \global\expandafter\advance\csname #1end@bysection\endcsname by 1%
5408   \begingroup%
5409     \makeatletter%
5410     \def\l@d@section##1{%
5411       \global\toggletrue{Xendfirstnote@#1}%
5412       \ifnumequal{##1}{\csname #1end@bysection\endcsname}%
5413         {\cslet{#1end}{\endprint}}%
5414         {\cslet{#1end}{\@gobbleseven}}%
5415     }%
5416     \input\l@auxdir\jobname.#1end%
5417     \global\Xendinsertsep@false%
5418   \endgroup%
5419 }%
5420 %

```

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.

```

5421 }%
5422 %

```



`\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`.

from the starting page number.

```

5423 \newcommand*{\setprintendlines}[6]{%
5424   \let\@annot@start@print\relax%
5425   \let\@annot@end@print\relax%
5426   \l@dpnumfalse%
5427   \l@ddashfalse%
5428   \l@d@elinfalse%
5429   %

```

First of all, we print the second page number only if the ending page number is different

```

5430   \ifnum#4=#1 \else
5431     \l@dpnumtrue
5432     \l@ddashtrue
5433   \fi
5434   %

```

We print the ending line number if: (1) we are printing the ending page number, or (2) the ending line is different from the starting line number.

```

5435   \ifboolexpr{%
5436     bool{\l@dpnum}%
5437     or not test{\ifnumequal{#2}{#5}}%
5438   }{%
5439     \l@d@elintrue%
5440     \l@ddashtrue%
5441     \l@d@elintrue
5442     \l@ddashtrue
5443     \unless\ifx\relax\annot@end%
5444       \def\@annot@end@print{%
5445         \l@wrapcs@ifnotemptybox{Xendwraplinenumannotation@\@currentseries
5446       }\@annot@end}%
5447     \fi%
5448   }%
5449   {%
5450   %

```

We define the starting line number annotation as a merge of the starting annotation and ending annotation if we don't print the ending line number. Otherwise, it is only the starting annotation.

```

5451   \ifl@d@elin%
5452   \def\@annot@start@print{%

```

```

5453 \l@wrapcs@ifnotemptybox{Xendwraplinenumannotation@ \@currentseries}{\
annot@start}%
5454 }%
5455 \else%
5456 \ifx\annot@start\annot@end%
5457 \unless\ifx\@annot@start\relax%
5458 \def\@annot@start@print{%
5459 \l@wrapcs@ifnotemptybox{Xendwraplinenumannotation@ \@currentseries
}{\annot@start}%
5460 }%
5461 \fi%
5462 \else%
5463 \ifx\@annot@end@print\relax%
5464 \def\@annot@start@print{%
5465 \l@wrapcs@ifnotemptybox{Xendwraplinenumannotation@ \@currentseries
}{%
5466 \ifx\annot@start\empty\else%
5467 \annot@start%
5468 \ifdefined\linangesep%
5469 \linangesep%
5470 \else%
5471 \csuse{Xendlinangeseparator@ \@currentseries}%
5472 \fi%
5473 \fi%
5474 \annot@end%
5475 }%
5476 }%
5477 \else%
5478 \let\@annot@start@print\@annot@end@print%
5479 \let\@annot@end@print\relax%
5480 \fi%
5481 \fi%
5482 \fi%
5483 %

```

We print the starting sub-line if it is nonzero.

```

5484 \l@d@ssubfalse
5485 \ifnum#3=0 \else
5486 \l@d@ssubtrue
5487 \fi
5488 %

```

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.

```

5489 \l@d@eslfalse
5490 \ifnum#6=0 \else
5491 \ifnum#6=#3
5492 \ifl@d@elin \l@d@esltrue \else \l@d@eslfalse \fi
5493 \else
5494 \l@d@esltrue

```

```

5495     \l@d@dashtrue
5496     \fi
5497     \fi%
5498 %

5499     \ifl@d@dash%
5500     \ifbool{expr{togl{fulllines@} or test{\ifcempty{Xendtwolines@}
5501     @currentseries}}}%
5502     {%
5503     \setistwofollowinglines{#1}{#2}{#4}{#5}%
5504     \ifbool{expr{
5505         (%
5506             togl {Xendtwolinesbutnotmore@}\@currentseries}%
5507             and not%
5508             (%
5509                 bool {istwofollowinglines@}%
5510             )%
5511         )%
5512         or%
5513         (%
5514             (not test{\ifnumequal{#1}{#4}})%
5515             and togl{Xendtwolinesonlyinsamepage@}\@currentseries}%
5516         )%
5517     }%
5518     {%
5519     \l@d@dashfalse%
5520     \l@d@Xtwolinestrue%
5521     \l@d@elinfalse%
5522     \l@d@eslfalse%
5523     \ifcempty{Xendmorethantwolines@}\@currentseries}%
5524     {%
5525     \ifistwofollowinglines@%
5526     \l@d@Xmorethantwolinestrue%
5527     \fi%
5528     }%
5529     }%
5530     }%
5531     \fi%
5532 %
5533 %

```

If the `\Xendnoidenticallylinenumannotation` is set for this series, we check if the ending annotation is identical to the starting. If true, we don't print the ending annotation

```

5534     \iftoggle{Xendnoidenticallylinenumannotation@}\@currentseries}{%
5535     \ifx\annot@start\annot@end%
5536     \let\annot@end@print\relax%
5537     \ifx\linenumrep\gobble%Don't print the dash if we're not printing
the line number

```

```

5538 \l@d@dashfalse%
5539 \fi%
5540 \fi%
5541 }{}%
5542 %

```

Finally, we check for `\Xendlinenumannotationonlyfirst` and `\Xendlinenumannotationonlyfirsti` and we redefine, if required, `\@annot@start@print` and `\annot@end@print`. We also store the current line number annotations.

```

5543 \iftoggle{Xendlinenumannotationonlyfirst@{\@currentseries}{%
5544 \ifboolexpr{%
5545 (%
5546 togl{Xendlinenumannotationonlyfirstintwo@{\@currentseries}%
5547 and test{\ifcsequal{annot@start}{prevannot@start@{\@currentseries}}}%
5548 and test{\ifcsequal{annot@end}{prevannot@end@{\@currentseries}}}%
5549 )%
5550 or%
5551 (%
5552 not togl{Xendlinenumannotationonlyfirstintwo@{\@currentseries}%
5553 test{\ifcsequal{annot@start}{prevannot@start@{\@currentseries}}}%
5554 )%
5555 }{}%
5556 \def\@annot@start@print{%
5557 \l@wrapcs@ifnotemptybox{Xendwraplinenumannotation@{\@currentseries
5558 }{\csuse{Xendsymlinenumannotation@{\@currentseries}}}%
5559 \let\@annot@end@print\relax%
5560 \ifx\linenumrep\@gobble%Don't print the dash if we're not printing
the line number
5561 \l@d@dashfalse%
5562 \fi%
5563 }{}%
5564 \global\cslet{prevannot@start@{\@currentseries}}{\annot@start}%
5565 \global\cslet{prevannot@end@{\@currentseries}}{\annot@end}%
5566 }%
5567 }{}%
5568 %

```

End of `\setprintendlines`.

```

5569 }%
5570 %

```

`\printendlines` Now we are ready to print it all.

```

5571 \def\printendlines#1|#2|#3|#4|#5|#6|#7|#8|{%
5572 \begingroup
5573 \setprintendlines{#1}{#2}{#3}{#4}{#5}{#6}%
5574 %

```

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.

```

5575 \ifdimequal{\csuse{Xendboxstartlinenum@\@currentseries}}{0pt}%
5576 {\bgroup}%
5577 {\leavevmode\hbox to \csuse{Xendboxstartlinenum@\@currentseries}\bgroup
\hfill}}%
5578 %

```

Then, print the starting page number-

```

5579 \ifboolexpr{%
5580 (%
5581 test{\ifcsstring{prevpagenum@\@currentseries}{#1}}%
5582 and not%
5583 (togl{Xendpagenumberonlyfirstifsingle@\@currentseries} and bool{
1@d@pnum}})%
5584 )%
5585 or%
5586 (%
5587 test {\ifcsstring{prevpagerange@\@currentseries}{#1-#4}}%
5588 )%
5589 }%
5590 {%
5591 \ifcsemtyp{Xendsympagenum@\@currentseries}%
5592 {\hspace{\csuse{Xendinplaceofpagenumber@\@currentseries}}}%
5593 {\csuse{Xendsympagenum@\@currentseries}}%
5594 }%
5595 {%
5596 \wrap@edcrossref{\@this@crossref@start}{\printnpnum{#1}}%
5597 }%
5598 %

```

Then, determine what must be printed before the start line.

```

5599 \ifl@d@dash%
5600 \ifl@d@pnum%
5601 \csuse{Xendlineprefixsingle@\@currentseries}%
5602 \else%
5603 \ifcsemtyp{Xendlineprefixmore@\@currentseries}%
5604 {\csuse{Xendlineprefixsingle@\@currentseries}}%
5605 {\csuse{Xendlineprefixmore@\@currentseries}}%
5606 \fi%
5607 \else%
5608 \csuse{Xendlineprefixsingle@\@currentseries}%
5609 \fi%
5610 %

```

Then print the starting line, followed, if needed, by the side flag and the starting sub line number, then the line number annotation.

```

5611 \ifcsstring{Xendlinenumannotationposition@ \@currentseries}{before}%
5612 {\@annot@start@print}%
5613 {}%
5614 \wrap@edcrossref{\@this@crossref@start}{%
5615 \ifledRcol@%
5616 \linenumrepR{#2}%
5617 \else%
5618 \linenumrep{#2}%
5619 \fi%
5620 }%
5621 \iftoggle{Xendlineflag@ \@currentseries}{\ifledRcol@ \@Rlineflag\fi}{}%
5622 \ifl@d@ssub%
5623 \csuse{Xendsublinesep@ \@currentseries}%
5624 \wrap@edcrossref{\@this@crossref@start}{%
5625 \ifledRcol@%
5626 \sublinenumrepR{#3}%
5627 \else%
5628 \sublinenumrep{#3}%
5629 \fi%
5630 }%
5631 \fi%
5632 \ifcsstring{Xendlinenumannotationposition@ \@currentseries}{after}%
5633 {\@annot@start@print}%
5634 {}%
5635 %

```

Close the box.

```

5636 \egroup%
5637 %

```

Open the box for the ending line number.

```

5638 \ifdimequal{\csuse{Xendboxendlinenum@ \@currentseries}}{0pt}%
5639 {\bgroup}%
5640 {\hbox to \csuse{Xendboxendlinenum@ \@currentseries}\bgroup}%
5641 %

```

Print the dash + the ending line number, or the line number range symbol.

```

5642 \ifl@d@Xtwolines%
5643 \ifl@d@Xmorethantwolines%
5644 \csuse{Xendmorethantwolines@ \@currentseries}%
5645 \else%
5646 \csuse{Xendtwolines@ \@currentseries}%
5647 \fi%
5648 \else%
5649 \ifl@d@dash%
5650 \ifdefined\linerangesep@%
5651 \linerangesep@%
5652 \else%
5653 \csuse{Xendlinerrangeseparator@ \@currentseries}%

```

```

5654 \fi%
5655 \fi%
5656 %

```

Print the ending page number.

```

5657 \ifl@d@pnum%
5658 \ifcsstring{prevpagerange@\@currentseries}{#1-#4}%
5659 {%
5660 \ifcsemtyp{Xendsympagenum@\@currentseries}%
5661 {\hspace{\csuse{Xendinplaceofpagenumber@\@currentseries}}}%
5662 {\csuse{Xendsympagenum@\@currentseries}}%
5663 }%
5664 {%
5665 \wrap@edcrossref{\@this@crossref@end}\printnpnum{#4}%
5666 }%
5667 \fi%
5668 %

```

Print the ending line number, with, if needed, the line prefix, and followed by the side flag, the subline number, and line number annotation.

```

5669 \ifcsstring{Xendlinenumannotationposition@\@currentseries}{before}%
5670 {\@annot@end@print}%
5671 }%
5672 \ifl@d@elin%
5673 \ifl@d@pnum\csuse{Xendlineprefixsingle@\@currentseries}\fi%
5674 \wrap@edcrossref{\@this@crossref@end}{%
5675 \ifledRcol%
5676 \linenumrepR{#5}%
5677 \else%
5678 \linenumrep{#5}%
5679 \fi%
5680 }%
5681 \iftoggle{Xendlineflag@\@currentseries}{\ifledRcol@\@Rlineflag\fi}{}%
5682 \fi%
5683 \ifl@d@esl%
5684 \ifl@d@elin%
5685 \csuse{Xendsublinesep@\@currentseries}%
5686 \fi%
5687 \wrap@edcrossref{\@this@crossref@end}{%
5688 \ifledRcol%
5689 \sublinenumrepR{#6}%
5690 \else%
5691 \sublinenumrep{#6}%
5692 \fi%
5693 }%
5694 \fi%
5695 \ifcsstring{Xendlinenumannotationposition@\@currentseries}{after}%
5696 {\@annot@end@print}%
5697 }%
5698 \fi%

```

```
5699 %
```

Close the ending line box.

```
5700 \ifdimequal{\csuse{Xendboxendlinenum@\@currentseries}}{0pt}%
5701 {}%
5702 {\hfill}%Prevent underfull hbox
5703 \egroup%
5704 %
```

And, finally, save, if needed, the current page number for the Xendpagenumberonlyfirst hooks.

```
5705 \iftoggle{Xendpagenumberonlyfirst@\@currentseries}%
5706 {\iftoggle{Xendpagenumberonlyfirstintwo@\@currentseries}%
5707 {\csxdef{prevpagerange@\@currentseries}{#1-#4}}%
5708 {\csxdef{prevpagenum@\@currentseries}{#4}}%
5709 }%
5710 {}%
5711 %
```

Now, the end of \printendlines macro.

```
5712 \endgroup%
5713 }%
5714
5715 %
```

**\printnpnum** A macro to print a page number in an endnote. Should not be override anymore

```
5716 \newcommand*\printnpnum[1]{\csuse{Xendbeforepagenumber@\@currentseries}
5717 }#1\csuse{Xendafterpagenumber@\@currentseries}}
5718 %
```

## XXI 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@.

```
5719 \newcommand{\newseries}[1]{%
5720 \def\do##1{\newseries@{##1}}%
5721 \docsvlist{#1}
5722 }
5723 %
```

**\@series** The \series@ macro is an etoolbox list, which contains the name of all series.

```
5724 \newcommand{\@series}{}
5725 %
```



The command `\newseries@series` creates a new series of the footnote.

```
\newseries@26 \newcommand{\newseries@}[1]{
5727 %
```

### XXI.1 Test if series is still existing

```
5728 \xifinlist{#1}{\@series}{\led@warn@SeriesStillExist{#1}}%
5729 {%
5730 %
```

### XXI.2 Init specific to reledpar

When calling `\newseries@` after having loaded `reledpar`, we need to load specific setting.

```
5731 \ifdefined\newseries@par%
5732 \newseries@par{#1}%
5733 \fi%
5734 %
```

### XXI.3 For critical footnotes

Critical footnotes are those which start with letters. We look for the `\nocritical` option of `reledmac`.

```
5735 \unless\ifnocritical@
5736 %
```

#### XXI.3.1 Options

```
5737 \newtoggle{Xlineflag@#1}
5738 \newtoggle{Xparindent@#1}
5739 \newtoggle{Xlemmadisablefontselection@#1}
5740 \csgdef{Xwrapcontent@#1}{}%
5741 \csgdef{Xbeforeinserting@#1}{}%
5742 \csgdef{Xhangindent@#1}{Opt}%
5743 \csgdef{Xragged@#1}{}%
5744 \csgdef{Xhsizetwocol@#1}{0.45 \hsize}%
5745 \csgdef{Xhsizethreecol@#1}{.3 \hsize}%
5746 \csgdef{Xcolalign@#1}{\raggedright}%
5747 \csgdef{Xnotenumfont@#1}{\normalfont}%
5748 \csgdef{Xnotefontsize@#1}{\footnotesize}%
5749 \csgdef{Xbhooknote@#1}{}%
5750 \csgdef{Xbhookgroup@#1}{}%
5751
5752 \csgdef{Xboxlinenum@#1}{Opt}%
5753 \csgdef{Xboxlinenumalign@#1}{L}%
```

```

5754 \csgdef{Xboxstartlinenum@#1}{Opt}%
5755 \csgdef{Xboxendlinenum@#1}{Opt}%
5756
5757
5758 \csgdef{Xboxsymlinenum@#1}{Opt}%
5759 \newtoggle{Xgroupbyline@#1}%
5760 \newtoggle{Xgroupbylineseparetwelines@#1}%
5761
5762 \newtoggle{Xnumberonlyfirstinline@#1}%
5763 \newtoggle{Xnumberonlyfirstintwelines@#1}%
5764
5765 \newtoggle{Xlinenumannotationonlyfirst@#1}%
5766 \newtoggle{Xlinenumannotationonlyfirstintwo@#1}%
5767
5768 \csgdef{Xtwelines@#1}{}%
5769 \csgdef{Xmorethantwelines@#1}{}%
5770 \csgdef{Xsublinesep@#1}{\fullstop}%
5771 \csgdef{Xpagelinesep@#1}{\csname Xsublinesep@#1\endcsname}%for
backward compatibility, call Xsublinesep@#1
5772 \newtoggle{Xtwelinesbutnotmore@#1}%
5773 \newtoggle{Xtwelinesonlyinsamepage@#1}%
5774 \newtoggle{Xonlypstart@#1}%
5775 \newtoggle{Xpstarteverytime@#1}%
5776
5777 \newtoggle{Xpstart@#1}%
5778 \newtoggle{Xpstartonlyfirst@#1}%
5779 \csdef{Xprevpstart@#1}{0}%Not a hook, but a tool for the hook
Xpstartonlyfirst
5780 \csgdef{Xpstartseparator@#1}{}%
5781
5782 \newtoggle{Xstanza@#1}%
5783 \newtoggle{Xstanzaonlyfirst@#1}%
5784 \csdef{Xprevstanza@#1}{0}%Not a hook, but a tool for the hook
Xstanzaonlyfirst
5785
5786 \csgdef{Xstanzaseparator@#1}{}%
5787
5788 \csgdef{Xsymlinenum@#1}{}%
5789 \csgdef{Xsymlinenumannotation@#1}{}%
5790
5791 \newtoggle{Xnonumber@#1}%
5792 \csgdef{Xbeforenumber@#1}{Opt}%
5793 \csgdef{Xtxtbeforenumber@#1}{}%
5794 \csgdef{Xafternumber@#1}{0.5em}%
5795 \newtoggle{Xnonbreakableafternumber@#1}%
5796 \csgdef{Xbeforesymlinenum@#1}{\csuse{Xbeforenumber@#1}}%
5797 \csgdef{Xaftersymlinenum@#1}{\csuse{Xafternumber@#1}}%
5798 \csgdef{Xinplaceofnumber@#1}{1em}%
5799 \global\cslet{Xlemmaseparator@#1}{\rbracket}%
5800 \csgdef{Xbeforelemmaseparator@#1}{0em}%

```

```

5801 \csgdef{Xafterlemmaseparator@#1}{0.5em}%
5802 \csgdef{Xinplaceoflemmaseparator@#1}{1em}%
5803 \csgdef{Xbeforenotes@#1}{1.2em \@plus .6em \@minus .6em}
5804 \csgdef{Xafterrule@#1}{0pt}
5805
5806 \csgdef{Xtxtbeforenotes@#1}{}
5807 \newtoggle{Xtxtbeforenotes@#1@typeset}%Not directly used by user,
but internal
5808 \newtoggle{Xtxtbeforenotesonlyonce@#1}%
5809
5810 \csgdef{Xmaxhnotes@#1}{0.8\vsizel}
5811 \newtoggle{Xnoteswidthliketwocolumns@#1}%
5812 \csgdef{Xparafootsep@#1}{}%
5813 \csgdef{Xafternote@#1}{1em plus.4em minus.4em}
5814 \csgdef{Xlinrangeseparator@#1}{\endashchar}%
5815
5816 \csgdef{Xlemmafont@#1}{}%
5817 \csgdef{Xwraplemma@#1}{}
5818 \csgdef{Xwidth@#1}{\hsizel}
5819 \csgdef{Xwraplinenumannotation@#1}{\textsuperscript}%
5820 \csgdef{Xlinenumannotationposition@#1}{after}%
5821 \expandafter\csgdef{Xinnotemark@#1}##1{\ledinnotemark{##1}}%
5822 \newtoggle{Xnoidenticallinenumannotation@#1}%
5823 %

```

### XXI.3.2 Create inserts, needed to add notes in foot

As regards inserts, see chapter 15 of *The TeXbook* by D. Knuth.

```

5824 \expandafter\newinsert\csname #1footins\endcsname%
5825 \unless\ifnoledgroup%
5826 \expandafter\newinsert\csname mp#1footins\endcsname%
5827 \fi%
5828 %

```

### XXI.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.

```

5829 \global\newbool{parapparatus@}{\expandafter\newcommand\expandafter
*}{\expandafter\newcommand}\csname #1footnote\endcsname[2]{}%
5830 \if@edtext@secondarg%
5831 \ifledRcol%
5832 \ifcsstring{Xonlyside@#1}{L}{\
led@error@note@called@onrightside{#1footnote}}{}%
5833 \else%
5834 \ifcsstring{Xonlyside@#1}{R}{\
led@error@note@called@onleftside{#1footnote}}{}%
5835 \fi%
5836 \begingroup%

```

```

5837 \newcommand{\content}{##2}%
5838 \ifnumberedpar%
5839 \ifledRcol%
5840 \ifluatex%
5841 \footnotelang@lua[R]%
5842 \fi%
5843 \@ifundefined{xpg@main@language}%if polyglossia
5844 {}%
5845 {\footnotelang@poly[R]}%
5846 \footnoteoptions@{R}{##1}{true}%
5847 \xright@appenditem{%
5848 \ifbool{indtl@innote}%
5849 {\unexpanded{\let\index\nindex}}%
5850 {}%
5851 \ifbool{indtl@notenumber}%
5852 {\unexpanded{\let\index\nindex}}%There is no note
number... so
5853 {}%
5854 \noexpand\Xnote@true%
5855 \unexpanded{\gdef\@currentnotetype{critical}}%
5856 \noexpand\prepare@Xprenotes{#1}%
5857 \noexpand\prepare@edindex@fornote{\l@d@nums}%
5858 \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.
5859 \noexpand\setcounter{stanzaR}{\the\c@stanzaR}%Save
stanzaR counter for footnote
5860 \unexpanded{\def\@this@crossref@start}{\theedtext:
start}%
5861 \unexpanded{\def\@this@crossref@end}{\theedtext:end}%
5862 \expandonce{\@beforeinsertofthisedtext}% Internal for
now, no reason to make it public
5863 \noexpand\parse@annot\l@current@annotR|%
5864 \noexpand\csuse{v#1footnote}{#1}%
5865 {\l@d@nums}{\expandonce\@tag}{\expandonce\content}}
%
5866 \noexpand\Xnote@false%
5867 \unexpanded{\advance\@edindex@fornote@m@one}%
5868 \ifbool{indtl@innote}%
5869 {\unexpanded{\let\index\orig@@index}}%
5870 {}%
5871 \ifbool{indtl@notenumber}%
5872 {\unexpanded{\let\index\orig@@index}}%
5873 {}%
5874 }\to\inserts@listR
5875 \footnoteoptions@{R}{##1}{false}%
5876 \global\advance\insert@countR \@one%
5877 \else%
5878 \ifluatex%

```

```

5879         \footnotelang@lua%
5880         \fi%
5881     \ifundefined{xpg@main@language}%if polyglossia
5882         {}%
5883         {\footnotelang@poly}%
5884     \footnoteoptions@{L}{##1}{true}%
5885     \xright@appenditem{%
5886         \ifbool{indtl@innote}%
5887             {\unexpanded{\let\index\nindex}}%
5888             {}%
5889     \ifbool{indtl@notenumber}%
5890         {\unexpanded{\let\index\nindex}}%There is no note
number... so
5891         {}%
5892         \noexpand\Xnote@true%
5893         \unexpanded{\gdef\@currentnotetype{critical}}%
5894         \noexpand\prepare@Xprenotes{#1}%
5895         \noexpand\prepare@edindex@fornote{\l@d@nums}%
5896         \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.
5897         \ifl@pairing%
5898         \noexpand\setcounter{stanzaL}{\the\c@stanzaL}%Save
stanzaR counter for footnote
5899         \fi%
5900         \unexpanded{\def\@this@crossref@start}{\theedtext:
start}}%
5901         \unexpanded{\def\@this@crossref@end}{\theedtext:end}%
5902         \expandonce{\@beforeinsertofthisedtext}%Internal for
now, no reason to make it public
5903         \noexpand\parse@annot\l@current@annot|%
5904         \noexpand\csuse{v#1footnote}%
5905             {#1}%
5906             {\l@d@nums}{\expandonce\@tag}{\expandonce\content
}}%
5907         \noexpand\Xnote@false%
5908         \unexpanded{\advance\@edindex@fornote@m@ne}%
5909         \ifbool{indtl@innote}%
5910             {\unexpanded{\let\index\orig@@index}}%
5911             {}%
5912         \ifbool{indtl@notenumber}%
5913             {\unexpanded{\let\index\orig@@index}}%
5914             {}%
5915         }\to\inserts@list
5916         \global\advance\insert@count \@ne%
5917         \footnoteoptions@{L}{##1}{false}%
5918     \fi
5919 \else
5920     \csuse{v#1footnote}{#1}{\{0|0|0|0|0|0\}}{##1}%

```

```

5921         \fi%
5922         \endgroup%
5923     \else%
5924         \led@err@FootnoteNotInSecondArgEdtext{#1}%
5925     \fi%
5926 \ignorespaces%
5927 }
5928 %

```

Create counter used to determine on which page the previous note was called.

```

5929     \expandafter\newcount\csname #1prevpage@num\endcsname%
5930     \expandafter\newcount\csname #1prevpage@numR\endcsname%
5931 %

```

We need to be able to modify `reledmac`'s footnote macros and restore their

```

5932 \global\csletcs{#1@@footnote}{#1footnote}
5933 %

```

### XXI.3.4 Set standard display

```

5934 \Xarrangement@normal{#1}%
5935 %

```

End of for critical footnotes.

```

5936 \fi
5937 %

```

## XXI.4 For familiar footnotes

Familiar footnotes are those which end with letters. We look for the `nofamiliar` option of `reledmac`.

```

5938 \unless\ifnofamiliar@
5939 %

```

### XXI.4.1 Options

```

5940 \newtoggle{parindentX@#1}
5941 \csgdef{wrapcontentX@#1}{}%
5942 \csgdef{hangindentX@#1}{Opt}%
5943 \csgdef{beforeinsertingX@#1}{}%
5944 \csgdef{raggedX@#1}{}%
5945 \csgdef{hsizetwocolX@#1}{0.45 \hsize}%
5946 \csgdef{hsizethreecolX@#1}{.3 \hsize}%
5947 \csgdef{colalignX@#1}{\raggedright}%
5948 \csgdef{notenumfontX@#1}{\normalfont}%
5949 \csgdef{notefontsizeX@#1}{\footnotesize}%
5950 \csgdef{bhooknoteX@#1}{}%

```

```

5951 \csgdef{bhookgroupX@#1}{}%
5952 \csgdef{afterruleX@#1}{Opt}
5953 \csgdef{beforenotesX@#1}{1.2em \@plus .6em \@minus .6em}
5954 \csgdef{maxhnotesX@#1}{0.8\size}%
5955 \newtoggle{noteswidthliketwocolumnsX@#1}%
5956 \csgdef{parafootsepX@#1}{}%
5957 \csgdef{afternoteX@#1}{1em plus.4em minus.4em}
5958 \csgdef{widthX@#1}{\hspace}%
5959 \expandafter\csgdef\innotemarkX@#1##1{\ledinnotemark{##1}}%
5960 \csgdef{txtbeforenotesX@#1}{}%
5961 \newtoggle{txtbeforenotesX@#1@typeset}%Not directly used by user,
but internal
5962 \newtoggle{txtbeforenotesonlyonceX@#1}%
5963 % End of for familiar footnotes.
5964 % \subsubsection{Create inserts, needed to add notes in foot}
5965 % As regards inserts, see chapter 15 of the TeXBook by D. Knuth.
5966 % \begin{macrocode}
5967 \expandafter\newinsert\csname footins#1\endcsname%
5968 \unless\ifnoledgroup%
5969 \expandafter\newinsert\csname mpfootins#1\endcsname%
5970 \fi%
5971 %

```

#### XXI.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.

```

5972 \global\expandafter\newcommand\csname footnote#1\endcsname[2][]{%
5973 \begingroup%
5974 \prepare@prenotesX{#1}%
5975 \newcommand{\content}{##2}%
5976 %
5977 %

```

If we are using the `\csquotes` package, we reset the quotation level.

```

5978 \ifdefined\csq@qllevel%
5979 \csq@qllevel=0\relax%
5980 \fi%
5981 %

```

If we are preparing parallel typesetting, we cannot just increase the footnote counter. Read `reledpar`'s handbook about that (V.1.2 p. 54). If we have a manual footnote mark, use it.

```

5982 \global\expandafter\advance\csname footnote#1@reading\
endcsname by \@ne%
5983 \ifstreamp{##1}{%
5984 \ifboolexpr{bool{l@dpairing} or bool{l@dprintingpages} or
bool{l@dprintingcolumns}}{%
5985 \ifcsdef{footnote#1@reading\the\csname footnote#1@reading
\endcsname=typeset}%

```

```

5986         {\setcounter{footnote#1}{\csuse{footnote#1reading\the\
csname footnote#1@reading\endcsname=typeset}}}%
5987         {\setcounter{footnote#1}{\the\csname footnote#1@reading\
endcsname}}}%
5988     }{%
5989         \stepcounter{footnote#1}%
5990     }%
5991 }{}%
5992 %

```

We also have to check consistency with `\onlysideX` setting.

```

5993         \ifledRcol%
5994         \ifcsstring{onlysideX@#1}{L}{\
led@error@note@called@onrightside{footnote#1}}}%
5995         \else%
5996         \ifcsstring{onlysideX@#1}{R}{\
led@error@note@called@onleftside{footnote#1}}}%
5997         \fi%
5998 %

```

And now, the feature not depending of whether we are preparing parallel typesetting

```

5999         \ifstrempy{##1}%
6000         {\protected@csxdef{@thefnmark#1}{\csuse{thefootnote#1}}}%
6001         {\protected@csxdef{@thefnmark#1}{##1}}}%
6002         \nottoggle{nomk@}%Nomk is set to true when using \
footnoteXnomk with \parpackage
6003         {\csuse{@footnotemark#1}}%
6004         {}%
6005         \ifluatex%
6006         \xdef\footnote@luatextextdir{\the\textdir}%
6007         \xdef\footnote@luatexpardir{\the\pardir}%
6008         \fi%
6009         \if@ledgroup%
6010         \led@set@index@for@note{#1}%
6011         \fi%
6012         \csuse{vfootnote#1}{#1}{\expandonce\content}{##1}\
m@mmf@prepare%
6013         \ifbool{indtl@innote}%
6014         {\let\index\orig@@index}%
6015         {}%
6016         \ifbool{indtl@notenumber}%
6017         {\let\index\orig@@index}%
6018         {}%
6019         \endgroup%
6020     }
6021 %

```

Then define the counters. The  $\LaTeX$  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. 54).

```

6022 \newcounter{footnote#1}
6023 \global\expandafter\renewcommand\csname thefootnote#1\endcsname{\
arabic{footnote#1}}
6024 \expandafter\newcount\csname footnote#1@reading\endcsname%
6025 %

```

Create counter used to determine on which page the previous note was called.

```

6026 \expandafter\newcount\csname prevpage#1@num\endcsname%
6027 \expandafter\newcount\csname prevpage#1@numR\endcsname%
6028 \expandafter\newcount\csname footnote#1@parafootsepX\endcsname%For
the case we use \parafootsepX outside of numbered section
6029 %

```

Add `\let\footnoteX\@gobble` to `\no@expands`.

```

6030 \expandafter\gappto\expandafter\no@expands\expandafter{\expandafter\
let\csname footnote#1\endcsname\@gobble}%
6031 %

```

And now, define `\footnoteXmark` and `\footnoteXtext`, equivalent to classical `\footnotemark` and `\footnotetext`.

```

6032 \expandafter\newcommand\csname footnote#1mark\endcsname[1] []{%
6033 \begingroup%
6034 \prepare@prenotesX{#1}%
6035 \ifstrepty{##1}{%
6036 \stepcounter{footnote#1}%
6037 }{%
6038 \setcounter{footnote#1}{##1}%
6039 }%
6040 \protected@csxdef{@thefnmark#1}{\csuse{thefootnote#1}}%
6041 \csuse{@footnotemark#1}%
6042 \m@mmf@prepare%
6043 \endgroup%
6044 }%
6045 \expandafter\newcommand\csname footnote#1text\endcsname[2] []{%
6046 \begingroup%
6047 \csuse{vfootnote#1}{#1}{\expandonce{##2}}{##1}%
6048 \endgroup%
6049 }%
6050 %

```

Do not forget to initialize the series.

```

6051 \arrangementX@normal{#1}%
6052 \fi
6053 %

```

## XXI.5 The endnotes

Endnotes are commands like `\Xendnote`, where X is a series letter. First, we check for the `noend` options.

```
6054 \unless\ifnoend@
6055 %
```

### XXI.5.1 The auxiliary file

`\l@d@Xend` Endnotes of all varieties are saved up in a file, one by series, typically named `⟨jobname⟩.Xend`.  
`\ifl@dend@X` `\l@d@end` is the output stream number for this file, and `\ifl@dend@X` is a flag that is  
`\l@dend@Xtrue` true when the file is open.  
`\l@dend@Xfalse`

```
6056 \expandafter\newwrite\csname l@d@#1end\endcsname%
6057 \expandafter\newif\csname ifl@dend@#1\endcsname%
6058 %
```

### XXI.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.

```
6059 \global\expandafter\newcommandx\csname #1endnote\endcsname[2][1,
6060 usedefault]{%
6061 \bgroup%
6062 \newlinechar='40%
6063 \global\@noneed@Footnotetrue%
6064 \newcommand{\content}{##2}%
6065 \stepcounter{labidx}%
6066 \expandafter\immediate\expandafter\write\csname l@d@#1end\endcsname{%
6067 \unexpanded{\def\sw@list@inedtext}%
6068 {\expandafter\unexpanded\expandafter{\sw@inthisedtext}}%
6069 \@percentchar\space%Explicit space, to add a linebreak in the
output file
6070 \noexpand\parse@annot\l@current@annot|\@percentchar\space%
6071 \expandafter\string\csname #1end\endcsname%
6072 {\ifnumberedpar@l@d@nums\fi}%
6073 {\ifnumberedpar@\expandonce\@tag\fi}%
6074 {\expandonce\content}%
6075 {\#1}%
6076 {\unexpanded{##1}}%
6077 {\ifledRcol R\else L\fi}%
6078 {\theedtext}%
6079 \@percentchar%
6080 }%
```

```

6081     \egroup%
6082     \ignorespaces%
6083 }%
6084 %

```

### XXI.5.3 Tools

The `\Xtoendnotes` command inserts any arbitrary content into the endnote file. It is an alias of the more generalist `\addtoendnotes`

```

6085
6086     \global\expandafter\newcommand\csname #1toendnotes\endcsname[1]{%
6087     \toendnotes[#1]{##1}%
6088 }%
6089
6090     \expandafter\WithSuffix\expandafter\newcommand\csname #1toendnotes\
6091     endcsname*[1]{%
6092     \toendnotes*[#1]{##1}%
6093 }%
6094 %

```

### XXI.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.

```

6095
6096     \global\cslet{#1end}{\@gobbleseven}
6097 %

```

We need to store the number of times `\doendnotesbysection` is called for one series.

```

6098     \global\expandafter\newcount\csname #1end@bysection\endcsname%
6099 %

```

### XXI.5.5 The options

```

6100     \csgdef{Xendwraplemma@#1}{%
6101     \csgdef{Xendwrapcontent@#1}{}%
6102     \csgdef{Xendtwolines@#1}{}%
6103     \csgdef{Xendmoreethantwolines@#1}{}%
6104     \newtoggle{Xendtwolinesbutnotmore@#1}{}%
6105     \newtoggle{Xendtwolinesonlyinsamepage@#1}{}%
6106     \newtoggle{Xendlemmadisablefontselection@#1}%
6107     \csgdef{Xendnotenumfont@#1}{\normalfont}%

```

```

6108 \csgdef{Xendnotefontsize@#1}{\footnotesize}%
6109 \csgdef{Xendbhooknote@#1}{}%
6110
6111 \csgdef{Xendsublinesep@#1}{\fullstop}%
6112
6113 \csgdef{Xendbeforenumber@#1}{Opt}%
6114 \csgdef{Xendafternumber@#1}{0.5em}%
6115
6116 \csgdef{Xendboxlinenum@#1}{Opt}%
6117 \csgdef{Xendboxlinenumalign@#1}{L}%
6118
6119 \csgdef{Xendboxstartlinenum@#1}{Opt}%
6120 \csgdef{Xendboxendlinenum@#1}{Opt}%
6121
6122 \csgdef{Xendlemmaseparator@#1}{}%
6123 \csgdef{Xendbeforelemmaseparator@#1}{0em}%
6124 \csgdef{Xendafterlemmaseparator@#1}{0.5em}%
6125 \csgdef{Xendinplaceoflemmaseparator@#1}{0.5em}%
6126
6127 \newtoggle{Xendparagraph@#1}%
6128 \csgdef{Xendafternote@#1}{1em plus.4em minus.4em}%
6129 \csgdef{Xendsep@#1}{}%
6130
6131 \csgdef{Xendinplaceofnumber@#1}{Opt}%
6132 \newtoggle{Xendnonumber@#1}%
6133
6134 \csgdef{Xendhangindent@#1}{Opt}%
6135
6136 \newtoggle{Xendnumberonlyfirstinline@#1}%
6137 \newtoggle{Xendnumberonlyfirstintwolines@#1}%
6138
6139 \newtoggle{Xendlinenumannotationonlyfirst@#1}%
6140 \newtoggle{Xendlinenumannotationonlyfirstintwo@#1}%
6141
6142 \csgdef{Xendbeforesymlinenum@#1}{\csuse{Xendbeforenumber@#1}}%
6143 \csgdef{Xendaftersymlinenum@#1}{\csuse{Xendafternumber@#1}}%
6144
6145 \csgdef{Xendsymlinenum@#1}{}%
6146 \csgdef{Xendsymlinenumannotation@#1}{}%
6147
6148 \csgdef{Xendboxsymlinenum@#1}{Opt}%
6149
6150 \csgdef{Xendbhooklinenumber@#1}{}%
6151 \csgdef{Xendehooklinenumber@#1}{}%
6152 \csgdef{Xendbhookinplaceofnumber@#1}{}%
6153 \csgdef{Xendehookinplaceofnumber@#1}{}%
6154
6155 \csgdef{Xendlinerangeseparator@#1}{\endashchar}%
6156
6157 \csgdef{Xendbeforepagenumber@#1}{p.}%

```

```

6158 \csgdef{Xendafterpagenumber@#1}{ }%
6159 \csgdef{Xendlineprefixsingle@#1}{}%
6160 \csgdef{Xendlineprefixmore@#1}{}%
6161
6162 \newtoggle{Xendlineflag@#1}
6163
6164 \csgdef{Xendlemmafont@#1}{}%
6165
6166 \csgdef{Xendlinenumannotationposition@#1}{after}%
6167 \csgdef{Xendwraplinenumannotation@#1}{\textsuperscript}%
6168 \newtoggle{Xendnoidenticallinenumannotation@#1}%
6169
6170 \newtoggle{Xendpagenumberonlyfirst@#1}%
6171 \newtoggle{Xendpagenumberonlyfirstifsingle@#1}%
6172 \newtoggle{Xendpagenumberonlyfirstintwo@#1}%
6173 \csgdef{Xendsympagenum@#1}{}%
6174 \csgdef{Xendinplaceofpagenumber@#1}{Opt}%
6175
6176 \csgdef{Xendtxtbeforenotes@#1}{}%
6177 \newtoggle{Xendfirstnote@#1}%Not a hook, but used to apply
Xendtxtbeforenotes
6178
6179 %

End of endnotes declaration

6180 \fi%
6181 %

Dump series in \@series

6182 \listxadd{\@series}{#1}
6183 }
6184 }% End of \newseries
6185 %

```

## XXI.6 Init standards series (A,B,C,D,E)

```

6186 \expandafter\newseries\expandafter{\default@series}
6187 %

```

## XXII Setting series display

### XXII.1 Change series order

`\seriesatbegin` `\seriesatbegin{⟨s⟩}` changes the order of series, to put the series  $\langle s \rangle$  at the beginning of the list. The series can be the result of a command.

```

6188 \newcommand{\seriesatbegin}[1]{%
6189   \StrDel{\@series}{#1}[\@series]%
6190   \edef\@new{ }%

```

```

6191 \listeadadd{\@new}{#1}%
6192 \listeadadd{\@new}{\@series}%
6193 \xdef\@series{\@new}%
6194 }
6195 %

```

`\seriesatend` And `\seriesatend` moves the series to the end of the list.

```

6196 \newcommand{\seriesatend}[1]{%
6197   \StrDel{\@series}{#1}[\@series]%
6198   \edef\@new{}%
6199   \listeadadd{\@new}{\@series}%
6200   \listeadadd{\@new}{#1}%
6201   \xdef\@series{\@new}%
6202 }
6203 %

```

## XXII.2 Test series order

`\ifseriesbefore` `\ifseriesbefore{<seriesA>}{<seriesB>}{<true>}{<false>}` expands to `<true>` if `<seriesA>` is printed before `<seriesB>`, or to `<false>` otherwise.

```

6204 \newcommand{\ifseriesbefore}[4]{%
6205   \StrPosition{\@series}{#1}[\@first]%
6206   \StrPosition{\@series}{#2}[\@second]%
6207   \ifnumgreater{\@second}{\@first}{#3}{#4}%
6208 }
6209 %

```

### XXII.2.1 Get the first series

In some specific case, we need to know the first series of the list of series.

```

\@getfirstseries:10 \newcommand{\@getfirstseries}{%
6211   \ifdefempty{\@series}%
6212   {\xdef\@firstseries{}}%
6213   {\StrChar{\@series}{1}[\@firstseries]}%
6214 }%
6215 %

```

## XXII.3 Series setting

### XXII.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.

### XXII.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`.

```

6216 \newcommandx{\settoggle@series}[5][4,5,usedefault]{%
6217   \def\do##1{%
6218     \ifcsdef{etb@tgl@#2@##1}{%
6219       \global\settoggle{#2@##1}{#3}%
6220     }{%
6221       \led@warn@BadHookSeries{##1}{#2}%
6222     }%
6223     \ifstrequal{#4}{critical}{
6224       \csuse{Xarrangement@}\csuse{series@display##1}{##1}%
6225     }{
6226       \ifstrequal{#4}{familiar}{%
6227         \csuse{arrangementX@}\csuse{series@displayX##1}{##1}%
6228       }{%
6229       }%
6230     }%
6231     \ifstreempty{#1}{%
6232       \dolistloop{\@series}%
6233       \ifstreempty{#5}{}%

```

```

6233         \docsvlist{#5}%
6234     }
6235 }%
6236 {%
6237     \docsvlist{#1}%
6238 }%
6239 }
6240 %

```

`\setcommand@series` `\setcommand@series{<series>}{<command>}{<value>}` is a generic command to store a 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`.
- #6 (optional): the number of arguments of the hook.

```

6241 \newcommandx{\setcommand@series}[6][4,5,6=0,usedefault]{%
6242     \def\do##1{
6243         \ifcsdef{#2@##1}{%
6244             \expandafter\global\expandafter\renewcommand\csname#2@##1\endcsname
6245             [#6]{#3}%
6246         }{%
6247             \led@warn@BadHookSeries{##1}{#2}%
6248         }%
6249         \ifstrequal{#4}{critical}{%
6250             \csuse{Xarrangement@}\csuse{series@display##1}{##1}%
6251         }{
6252             \ifstrequal{#4}{familiar}{%
6253                 \csuse{arrangementX@}\csuse{series@displayX##1}{##1}%
6254             }{
6255                 \ifstrequal{#5}{%
6256                     \dolistloop{\@series}%
6257                     \ifstrequal{#5}{%
6258                         \docsvlist{#5}
6259                     }
6260                 }%
6261                 {%
6262                     \docsvlist{#1}%

```



```

6263 }%
6264 }%
6265 %

```

### XXII.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`. The third argument

```

6266 \newcommandx{\newhookcommand@series}[3][2,3=0,usedefault]{%
6267 \global\expandafter\newcommand\expandafter*\csname #1\endcsname[2][]{%
6268 \setcommand@series{##1}{#1}{##2}[][#2][#3]%
6269 }%
6270 \ifstrempy{#2}{-}{%
6271 \def\do##1{%
6272 \global\expandafter\newcommand\expandafter*\csname #1##1\endcsname
6273 [1]{%
6274 \csuse{#1}[##1]{####1}%
6275 }%
6276 }%
6277 \docsvlist{#2}%
6278 }%
6279 %

```

**\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`.

```

6280 \newcommandx{\newhooktoggle@series}[2][2,usedefault]{%
6281 \global\expandafter\newcommandx\expandafter*\csname #1\endcsname[2][1,2={
6282 true},usedefault]{%
6283 \settoggle@series{##1}{#1}{##2}[][#2]%
6284 }%
6285 \ifstrempy{#2}{-}{%
6286 \def\do##1{%
6287 \global\expandafter\newcommand\expandafter*\csname #1##1\endcsname{%
6288 \csuse{#1}[##1]%
6289 }%
6290 }%
6291 \docsvlist{#2}%
6292 }%
6293 %

```

`\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 of notes

```

6294 \newcommand{\newhooktoggle@series@reload}[2]{%
6295   \global\expandafter\newcommandx\expandafter*\csname #1\endcsname[2][1,2={
true},usedefault]{%
6296     \settoggle@series{##1}{#1}{##2}[#2]%
6297   }%
6298 }%
6299 %

```

`\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.

```

6300 \newcommand{\newhookcommand@series@reload}[2]{%
6301   \global\expandafter\newcommand\expandafter*\csname #1\endcsname[2][1]{%
6302     \setcommand@series{##1}{#1}{##2}[#2]%
6303   }%
6304 }
6305 %

```

### XXII.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.

```

6306 \unless\ifnocritical@
6307   \newhookcommand@series{Xwrapcontent}%
6308   \newhookcommand@series{Xbeforeinserting}%
6309   \newhookcommand@series{Xlemmafont}%
6310   \newhookcommand@series{Xwraplemma}%
6311   \newhooktoggle@series{Xparindent}
6312   \newhookcommand@series{Xhangindent}
6313   \newhookcommand@series{Xragged}
6314   \newhookcommand@series{Xhsizetwocol}
6315   \newhookcommand@series{Xhsizethreecol}
6316   \newhookcommand@series{Xcolalign}%
6317   \newhookcommand@series{Xnotenumfont}
6318   \newhookcommand@series{Xbhooknote}
6319   \newhookcommand@series@reload{Xbhookgroup}{critical}
6320   \newhookcommand@series{Xboxsymlinenum}%
6321   \newhookcommand@series{Xsymlinenum}
6322   \newhookcommand@series{Xsymlinenumannotation}%
6323   \newhookcommand@series{Xbeforenumber}
6324   \newhookcommand@series{Xtxtbeforenumber}
6325   \newhookcommand@series{Xafternumber}
6326   \newhookcommand@series{Xbeforesymlinenum}
6327   \newhookcommand@series{Xaftersymlinenum}

```

```

6328 \newhookcommand@series{Xinplaceofnumber}
6329 \newhookcommand@series{Xlemmaseparator}
6330 \newhookcommand@series{Xbeforelemmaseparator}
6331 \newhookcommand@series{Xafterlemmaseparator}
6332 \newhookcommand@series{Xinplaceoflemmaseparator}
6333 \newhookcommand@series{Xtxtbeforenotes}
6334 \newhooktoggle@series{Xtxtbeforenotesonlyonce}%
6335 \newhookcommand@series@reload{Xafterrule}{critical}
6336 \newhooktoggle@series{Xnumberonlyfirstinline}
6337 \newhooktoggle@series{Xnumberonlyfirstintwolines}
6338 \newhooktoggle@series{Xlinenumannotationonlyfirst}%
6339 \newhooktoggle@series{Xlinenumannotationonlyfirstintwo}%
6340 \newhooktoggle@series{Xgroupbyline}%
6341 \newhooktoggle@series{Xgroupbylineseparetwolines}%
6342 \newhooktoggle@series{Xnonumber}
6343
6344 \newhooktoggle@series{Xpstart}
6345 \newhooktoggle@series{Xpstartonlyfirst}%
6346 \newhooktoggle@series{Xpstarteverytime}%
6347 \newhookcommand@series{Xpstartseparator}%
6348
6349 \newhooktoggle@series{Xstanza}%
6350 \newhooktoggle@series{Xstanzaonlyfirst}%
6351 \newhookcommand@series{Xstanzaseparator}%
6352
6353 \newhooktoggle@series{Xonlypstart}
6354 \newhooktoggle@series{Xnonbreakableafternumber}
6355 \newhooktoggle@series{Xlemmadisablefontselection}
6356 \newhookcommand@series@reload{Xmaxhnotes}{critical}
6357 \newhookcommand@series@reload{Xbeforenotes}{critical}
6358 \newhooktoggle@series@reload{Xnoteswidthliketwocolumns}{critical}%
6359 \newhookcommand@series@reload{Xnotefontsize}{critical}
6360
6361 \newhookcommand@series{Xboxlinenum}%
6362 \newhookcommand@series{Xboxlinenumalign}%
6363
6364 \newhookcommand@series{Xboxstartlinenum}%
6365 \newhookcommand@series{Xboxendlinenum}%
6366
6367 \newhookcommand@series{Xafternote}%
6368 \newhookcommand@series{Xparafootsep}
6369 \newhookcommand@series@reload{Xwidth}{critical}%
6370
6371 \newhookcommand@series{Xinnotemark}[] [1]%
6372
6373 \ifundef{\Xhsize}%
6374 {%
6375   \newcommandx{\Xhsize}[2] [1,usedefault]{%
6376     \led@warning@Xhsize@deprecated%
6377     \Xwidth[#1]{#2}%

```

```

6378     }%
6379     }%
6380     {}%
6381 \fi
6382 %

```

Now, hooks for critical notes which also apply to crossreferencing and line numbering at the sides of the page.

```

6383 \newhooktoggle@series{Xlineflag}[appref,SEref]
6384 \newhookcommand@series{Xtwolines}[appref,SEref]
6385 \newhookcommand@series{Xmorethantwolines}[appref,SEref]
6386 \newhookcommand@series{Xsublinesep}[appref,SEref,side]%
6387 \newhookcommand@series{Xpagelinesep}[appref,SEref,side]%
6388 \newhooktoggle@series{Xtwolinesbutnotmore}[appref,SEref]
6389 \newhooktoggle@series{Xtwolinesonlyinsamepage}[appref,SEref]
6390 \newhookcommand@series{Xlinerangeseparator}[appref,SEref]
6391 \newhookcommand@series{Xlinenumannotationposition}[side,appref,SEref]%
6392 \newhookcommand@series{Xwraplinenumannotation}[side,ref,appref,SEref]%
6393 \newhooktoggle@series{Xnoidenticallinenumannotation}[side,ref,appref,SEref]
6394 %

```

### XXII.3.5 Options for familiar notes

Before generating the optional commands for familiar notes, we check the `\nofamiliar` option.

```

6395 \unless\ifnofamiliar@
6396 \newhookcommand@series{wrapcontentX}%
6397 \newhookcommand@series{beforeinsertingX}%
6398 \newhooktoggle@series{parindentX}
6399 \newhookcommand@series{hangindentX}
6400 \newhookcommand@series{raggedX}
6401 \newhookcommand@series{hsizetwocolX}
6402 \newhookcommand@series{hsizethreecolX}
6403 \newhookcommand@series{colalignX}%
6404 \newhookcommand@series{notenumfontX}
6405 \newhookcommand@series{bhooknoteX}
6406 \newhookcommand@series@reload{bhookgroupX}{familiar}
6407 \newhookcommand@series@reload{beforenotesX}{familiar}
6408 \newhookcommand@series@reload{maxhnotesX}{familiar}
6409 \newhooktoggle@series@reload{noteswidthliketwocolumnsX}{familiar}%
6410 \newhookcommand@series@reload{afterruleX}{familiar}
6411 \newhookcommand@series@reload{notefontsizeX}{familiar}
6412 \newhookcommand@series{afternoteX}
6413 \newhookcommand@series{parafootsepX}
6414 \newhookcommand@series{txtbeforenotesX}%
6415 \newhooktoggle@series{txtbeforenotesonlyonceX}%
6416 \newhookcommand@series@reload{widthX}{familiar}%

```

```

6417 \newhookcommand@series{innotemarkX}[] [1]%
6418
6419 \ifundef{\hsizeX}%
6420 {%
6421   \newcommand{\hsizeX}[2] [1,usedefault]{%
6422     \led@warning@hsizeX@deprecated%
6423     \widthX[#1]{#2}%
6424   }%
6425 }%
6426 {}%
6427 \fi
6428 %

```

### XXII.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.

```

6429 \unless\ifnoend@
6430   \newhookcommand@series{Xendwraplemma}
6431   \newhookcommand@series{Xendwrapcontent}
6432   \newhookcommand@series{Xendnotenumfont}
6433   \newhookcommand@series{Xendlemmafont}%
6434   \newhookcommand@series{Xendbhooknote}
6435
6436   \newhookcommand@series{Xendboxlinenum}%
6437   \newhookcommand@series{Xendboxlinenumalign}%
6438
6439   \newhookcommand@series{Xendboxstartlinenum}%
6440   \newhookcommand@series{Xendboxendlinenum}%
6441
6442   \newhookcommand@series{Xendnotefontsize}
6443   \newhooktoggle@series{Xendlemmadisablefontselection}
6444   \newhookcommand@series{Xendlemmaseparator}
6445   \newhookcommand@series{Xendbeforelemmaseparator}
6446   \newhookcommand@series{Xendafterlemmaseparator}
6447   \newhookcommand@series{Xendinplaceoflemmaseparator}
6448
6449   \newhookcommand@series{Xendbeforenumber}%
6450   \newhookcommand@series{Xendafternumber}%
6451
6452   \newhooktoggle@series{Xendparagraph}
6453   \newhookcommand@series{Xendafternote}
6454   \newhookcommand@series{Xendsep}
6455
6456   \newhookcommand@series{Xendinplaceofnumber}%
6457   \newhooktoggle@series{Xendnonumber}%
6458
6459   \newhooktoggle@series{Xendnumberonlyfirstinline}%
6460   \newhooktoggle@series{Xendnumberonlyfirstintwolines}%

```

```

6461 \newhooktoggle@series{Xendlinenumannotationonlyfirst}%
6462 \newhooktoggle@series{Xendlinenumannotationonlyfirstintwo}%
6463
6464 \newhookcommand@series{Xendsymmlinenumber}%
6465 \newhookcommand@series{Xendbeforesymmlinenumber}%
6466 \newhookcommand@series{Xendaftersymmlinenumber}%
6467 \newhookcommand@series{Xendboxsymmlinenumber}%
6468
6469 \newhookcommand@series{Xendsymmlinenumberannotation}%
6470
6471 \newhookcommand@series{Xendbhooklinenumber}%
6472 \newhookcommand@series{Xendahooklinenumber}%
6473 \newhookcommand@series{Xendbhookinplaceofnumber}%
6474 \newhookcommand@series{Xendahookinplaceofnumber}%
6475
6476 \newhookcommand@series{Xendhangindent}%
6477
6478 \newhooktoggle@series{Xendpagenumberonlyfirst}%
6479 \newhooktoggle@series{Xendpagenumberonlyfirstifsingle}%
6480 \newhooktoggle@series{Xendpagenumberonlyfirstintwo}%
6481 \newhookcommand@series{Xendsympagenumber}%
6482 \newhookcommand@series{Xendinplaceofpagenumber}%
6483
6484 \newhookcommand@series{Xendtxtbeforenotes}%
6485
6486
6487 \fi
6488 \newhooktoggle@series{Xendlineflag}[apprefwithpage,SErefwithpage]
6489 \newhookcommand@series{Xendtwwlines}[apprefwithpage,SErefwithpage]
6490 \newhookcommand@series{Xendmorethantwwlines}[apprefwithpage,SErefwithpage]
6491 \newhooktoggle@series{Xendtwwlinesbutnotmore}[apprefwithpage,SErefwithpage]
6492 \newhooktoggle@series{Xendtwwlinesonlyinsamepage}[apprefwithpage,
6493 SErefwithpage]
6494 \newhookcommand@series{Xendlinerrangeseparator}[apprefwithpage,SErefwithpage]
6495 ]
6496 \newhookcommand@series{Xendbeforepagenumber}[apprefwithpage,SErefwithpage,
6497 SErefonlypage]
6498 \newhookcommand@series{Xendafterpagenumber}[apprefwithpage,SErefwithpage]
6499 \newhookcommand@series{Xendlineprefixsingle}[apprefwithpage,SErefwithpage]
6500 \newhookcommand@series{Xendlineprefixmore}[apprefwithpage,SErefwithpage]
6501 \newhookcommand@series{Xendsublinesep}[apprefwithpage,SErefwithpage]
6502
6503 \newhookcommand@series{Xendlinenumannotationposition}[apprefwithpage,
6504 SErefwithpage]%
6505 \newhookcommand@series{Xendwraplinenumannotation}[apprefwithpage,
6506 SErefwithpage]%
6507 \newhooktoggle@series{Xendnoidenticallinenumannotation}[apprefwithpage,
6508 SErefwithpage]%
6509 %

```

## XXII.4 Hooks for a particular footnote

`\newhooktoggle@specific` `\newhooktoggle@specific` is a generic command to create boolean hook specific to a note.

```

6504 \newcommand{\newhooktoggle@specific}[1]{%
6505   \newtoggle{#1}%
6506   \listgadd{\hooktoggle@specific}{#1}%
6507   \define@key[mac]{truefootnoteoption}{#1}[]{\global\settoggle{#1}{true}}%
When enabling footnote option
6508   \define@key[mac]{falsefootnoteoption}{#1}[]{\global\settoggle{#1}{false}
6509   }}
6509 }
6510 %

```

`\newhookarg@specific` `\newhookarg@specific` is a generic command to create argumen hook specific to a note.

```

6511 \newcommand{\newhookarg@specific}[1]{%
6512   \listgadd{\hookarg@specific}{#1}%
6513   \define@key[mac]{truefootnoteoption}{#1}{\csgdef{#1@}{##1}}%When enabling
footnote option
6514   \define@key[mac]{falsefootnoteoption}{#1}{\global\csundef{#1@}}%When
disabling footnote option
6515 }
6516 %

```

`\hooktoggle@specific` `\hooktoggle@specific` The `\add@hooktoggle@specific@to@cs` macro stores to a macro all the current values of hook toggle specific to a command. It is useful for the `\Xgroupbyline` option, which does not immediately add notes to the inserts list. It uses the `\hooktoggle@specific` list.

```

6517 \def\hooktoggle@specific{%
6518 \newcommand{\add@hooktoggle@specific@to@cs}[1]{%
6519   \def\do##1{%
6520     \iftoggle{##1@}{%
6521       \ifcsdef{#1}{%
6522         \csgappto{#1}{\toggletrue{##1@}}%
6523       }{%
6524         \csgdef{#1}{\toggletrue{##1@}}%
6525       }%
6526     }{%
6527       \ifcsdef{#1}{%
6528         \csgappto{#1}{\togglefalse{##1@}}%
6529       }{%
6530         \csgdef{#1}{\togglefalse{##1@}}%
6531       }%
6532     }%
6533   }%
6534   \dolistloop{\hooktoggle@specific}%

```

```
6535 }%
6536 %
```

`\hookarg@specific` The same, but for optional argument of critical footnotes with assigned value.  
`\add@hookarg@specific@to@cs`

```
6537 \def\hookarg@specific{%
6538 \newcommand{\add@hookarg@specific@to@cs}[1]{%
6539 \def\do##1{%
6540 \ifcsvoid{##1@}{%
6541 \ifcsdef{#1}{%
6542 \csxappto{#1}{%
6543 \noexpand\csdef%
6544 {##1@}{\csname##1@endcsname}%
6545 }%
6546 }{%
6547 \csxdef{#1}{%
6548 \noexpand\csdef%
6549 {##1@}{\csname##1@endcsname}%
6550 }%
6551 }%
6552 }%
6553 }%
6554 \dolistloop{\hookarg@specific}%
6555 }%
6556 %
```

And now, we define some hooks specific to a note.

```
6557 \newhooktoggle@specific{fulllines}%
6558 \newhooktoggle@specific{nonum}
6559 \newhooktoggle@specific{nosep}
6560 \newhooktoggle@specific{noprefix}%
6561 \newhooktoggle@specific{prefixmore}%
6562 \newhookarg@specific{linangesep}
6563 %
```

`linangesep@` `\linangesep@` is defined by the option `linangesep` 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.

```
6564 %
```

`\nomk@` `\nomk@` toggle is used by `reledpar` to remove the footnote mark in the text when using `\footnotexmk`. Read `reledpar` handbook.

```
6565 \newtoggle{nomk@}%
6566 %
```



## XXII.5 Alias

`\Xnolemmaseparator` `\Xnolemmaseparator[⟨series⟩]` is just an alias for `\Xlemmaseparator[⟨series⟩]{}`.

```
6567 \newcommand*{\Xnolemmaseparator}[1][1]{\Xlemmaseparator[#1]{}}
6568 %
```

## XXIII Output routine

Now we begin the output routine and associated things.

### XXIII.1 Extra footnotes output

With luck we might only have to change `\@makecol` and `\@reinserts` of  $\TeX$ '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

`\l@ddoxtrafeet` `\l@ddoxtrafeet` 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.

```
6569 \newcommand*{\l@ddoxtrafeet}{%
6570   \IfStrEq{familiar-critical}{\@fnpos}
6571   {\do@feetX\do@Xfeet}%
6572   {%
6573     \IfStrEq{critical-familiar}{\@fnpos}%
6574     {\do@Xfeet\do@feetX}%
6575     {%
6576       \setbox\@outputbox \vbox{%
6577         \unvbox\@outputbox%
6578         \do@feet@custom@order}{\@fnpos}%
6579       }%
6580     }%
6581   }%
6582 }%
6583
6584 %
```

`\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.

```
6585 \newcommand{\do@feet@custom@order}[2]{%
6586   \def\do##1{%
6587     \edef\@notesseries{\@firstoftwo##1}%
6588     \edef\@notetype{\@secondoftwo##1}%
6589     \ifdefstring{\@notetype}{critical}%
6590       {\csuse{#1append@Xnotes}{\@notesseries}}%
```

```

6591     {\ifdefstring{\@@notetype}{familiar}}%
6592     {\csuse{#1append@notesX}{\@@notesseries}}}%
6593     {}%
6594   }%
6595 }%
6596 \expandafter\docsvlist\expandafter{#2}%
6597 }%
6598 %

```

**\do@Xfeet** \do@Xfeet is the code extending \@makecol to cater to the extra critical feet.

```

6599 \newcommand*{\do@Xfeet}{%
6600   \setbox\@outputbox \vbox{%
6601     \unvbox\@outputbox
6602     \@opXfeet}}
6603 %

```

**\@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
6604 \newcommand{\append@Xnotes}[1]{%
6605   \ifvoid\csuse{#1footins}\else%
6606     \global\skip\csuse{#1footins}=\csuse{Xbeforenotes@#1}%
6607     \global\advance\skip\csuse{#1footins} by\csuse{Xafterrule@#1}%
6608     \print@Xnotes{#1}%
6609   \fi%
6610 }%
6611 %

```

The normal way to add one series, \print@Xnotes, is replaced by reledpar when using \Pages.

```

6612 \newcommand\print@Xnotes[1]{%
6613   \xdef\@currentseries{#1}%
6614   \csuse{#1footstart}{#1}%
6615   \csuse{#1footgroup}{#1}%
6616 }%
6617 %

```

We print all series of notes by looping on them. We check before printing them that they are not voided.

```

6618 \newcommand*{\@opXfeet}{%
6619   \unless\ifnocritical@%
6620     \gdef\firstXseries@{}%
6621     \def\do##1{%
6622       \append@Xnotes{##1}%
6623     }%
6624     \dolistloop{\@series}%
6625   \fi%
6626 }%
6627 %

```

`\l@ddodoreintrafeet` `\l@ddodoreintrafeet` is the code catering for the extra footnotes within `\@reinserts`. We use the same category and ordering as in `\l@ddoxtrafeet`.

```

6628 \newcommand*{\l@ddodoreintrafeet}{%
6629   \IfStrEq{familiar-critical}{\@fnpos}
6630     {\@doreinfeetX\X@doreinfeet}%
6631   {%
6632     \IfStrEq{critical-familiar}{\@fnpos}%
6633       {\X@doreinfeet\@doreinfeetX}%
6634       {\@doreinfeetX\X@doreinfeet}%
6635     }%
6636 }
6637
6638 %

```

`\X@doreinfeet` `\X@doreinfeet` is the code for catering for the extra critical footnotes within `\@reinserts`.

```

6639 \newcommand*{\X@doreinfeet}{%
6640   \unless\ifnocritical{%
6641     \def\do##1{%
6642       \ifvoid\csuse{##1footins}\else%
6643         \insert\csuse{##1footins}{\unvbox\csuse{##1footins}}%
6644       \fi}%
6645     \dolistloop{\@series}
6646   \fi%
6647 }
6648
6649 %

```

`\print@notesX` We have to add all the new kinds of familiar footnotes to the output routine. A macro  
`\append@notesX` which appends the familiar footnotes of one series onto the output routine, also adding  
`\do@feetX` vertical skip before notes.

```

\@doreinfeetX
6650 \newcommand{\append@notesX}[1]{%
6651   \ifvoid\csuse{footins#1}\else%
6652     \global\skip\csuse{footins#1}=\csuse{beforenotesX#1}%
6653     \global\advance\skip\csuse{footins#1} by\csuse{afterruleX#1}%
6654     \print@notesX{#1}%
6655   \fi%
6656 }%
6657 %

```

The normal way to print one series of notes. `\print@Xnotes` is replaced by `reledpar` when using `\Pages`.

```

6658 \newcommand\print@notesX[1]{%
6659   \xdef\@currentseries{#1}%
6660   \csuse{footstart#1}{#1}%
6661   \csuse{footgroup#1}{#1}%
6662 }%
6663 %

```

We print all the series of notes by looping over them. We check before printing them that they are not voided.

```

6664 \newcommand*{\do@feetX}{%
6665   \unless\ifnofamiliar%
6666     \gdef\firstseriesX@{}%
6667     \setbox\@outputbox \vbox{%
6668       \unvbox\@outputbox%
6669       \def\do##1{%
6670         \append@notesX{##1}%
6671       }%
6672       \dolistloop{\@series}}%
6673   \fi%
6674 }%
6675
6676 \newcommand{\@doreinfeetX}{%
6677   \unless\ifnofamiliar%
6678     \def\do##1{%
6679       \ifvoid\csuse{footins##1}\else
6680         \insert%
6681           \csuse{footins##1}
6682           {\unvbox\csuse{footins##1}}%
6683       \fi%
6684     }%
6685     \dolistloop{\@series}%
6686   \fi%
6687 }%
6688
6689 %

```

## XXIII.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.)

```

6690 \@ifclassloaded{memoir}{%
6691   %

```

memoir is loaded so we use memoir's built in hooks.

```

6692   \g@addto@macro{\m@mmdoextrafeet}{\l@ddoxtrafeet}%
6693   \g@addto@macro{\m@mdodoreinextrafeet}{\l@ddodoreinextrafeet}%
6694 }{%
6695 %

```

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 has 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.

```

6696 \ifboolexpr{%
6697   test{\@ifpackageloaded{fancyhdr}}%
6698   and test {\ifdef{\latex@makecol}}%
6699 }{%
6700   \patchcmd%
6701     {\latex@makecol}%
6702     {\xdef\@freelist{\@freelist\@midlist}}%
6703     {\xdef\@freelist{\@freelist\@midlist}\l@ddoxtrafeet}%
6704     {}%
6705     {\led@error@fail@patch@makecol}%
6706   }{%
6707     \patchcmd%
6708       {\@makecol}%
6709       {\xdef\@freelist{\@freelist\@midlist}}%
6710       {\xdef\@freelist{\@freelist\@midlist}\l@ddoxtrafeet}%
6711       {}%
6712       {\led@error@fail@patch@makecol}%
6713     }%
6714
6715   \patchcmd%
6716     {\@reinserts}%
6717     {\ifvbox}%
6718     {\l@ddodoreinextrafeet\ifvbox}%
6719     {}%
6720     {\led@error@fail@patch@reinserts}%
6721   }
6722
6723   %

```

It turns out that \@docclearpage also needs modifying.

**\if@led@nofoot** We have to check if there are any leftover feet.

```

6724 \newif\if@led@nofoot
6725
6726 %

```

```

6727 \@ifclassloaded{memoir}{%
6728 %

```

If the memoir class is loaded, we hook into its modified \@docclearpage.

```

\@mem@extranofeet 29 \g@addto@macro{\@mem@extranofeet}{%%
6730 \def\do#1{%
6731   \unless\ifnocritical@%
6732     \ifvoid\csuse{#1footins}\else\@mem@nofootfalse\fi%
6733   \fi%
6734   \unless\ifnofamiliar@%

```

```

6735     \ifvoid\csuse{footins#1}\else\@mem@nofootfalse\fi%
6736     \fi%
6737   }
6738   \dolistloop{\@series}%
6739 }%
6740 }{%
6741 %

```

As memoir is not loaded we have patch \@doclearpage.

```

\@led@testifnofoot 42 \newcommand*{\@led@testifnofoot}{%
\@doclearpage 43   \@led@nofoottrue%
6744   \ifvoid\footins\else%
6745     \@led@nofootfalse%
6746   \fi%
6747   \def\do##1{%
6748     \unless\ifnocritical@%
6749       \ifvoid\csuse{##1footins}\else%
6750         \@led@nofootfalse%
6751       \fi%
6752     \fi%
6753     \unless\ifnofamiliar@%
6754       \ifvoid\csuse{footins##1}\else%
6755         \@led@nofootfalse%
6756       \fi%
6757     \fi%
6758   }%
6759   \dolistloop{\@series}%
6760 }%
6761
6762 \pretocmd%
6763 {\@doclearpage}%
6764 {\@led@testifnofoot}%
6765 {}%
6766 {\led@error@fail@patch@@doclearpage}%
6767
6768 \patchcmd%
6769 {\@doclearpage}%
6770 {\ifvoid\footins}%
6771 {\if@led@nofoot}%
6772 {}%
6773 {\led@error@fail@patch@@doclearpage}%
6774
6775 }
6776
6777 %

```

## XXIV Page numbering in parallel typesetting

The `reledpar` package has two options which change the way page numbering works. We need to implement these options on `reledmac` and not on `reledpar` because they have some consequences for the `reledmac` auxiliary files (numbered file; see V.12 p. 132). The `sameparallelpagenunder` option allows the same page number on both left and right side. The `prevpgnotnumbered` option allows an empty (not numbered) right-side page before `\Pages`.

We cannot implement these two options by changing the value of the `page` counter, since its value is used by many  $\TeX$  features to determine whether a page is left (even numbered) or right (odd numbered). Consequently, we have to do it by patching `\thepage`, in order to use the value of the `par@page` counter instead of the value of the `page` counter.

This counter will be increased in a patched version of  $\TeX$ 's `\@outputpage` macro, as is the `page` counter in this macro. However, this increase will take account of the options.

`\par@patch@thepage` `\par@patch@thepage` patches `\thepage` in order to use the value of `par@page` counter and not the value of `page`. It must be called after any redefinition of `\thepage`. That is why we insert it at the end of the  $\TeX$  macro `\pagenumbering`, which is called by some `\xxxmatter` commands. In cases when we are using the `memoir` class, we insert it at the end of `\@mempnum`. When using `\pagenumbering`, we also need to reset the `par@page` counter. Consequently, we put `\par@patch@thepage` and counter reset in `\par@patch@pagenumbering`. We also call `\par@patch@thepage` at the beginning of the document.

```

6778
6779 \newcommand{\par@patch@thepage}{%
6780   \ifbool{expr}{%
6781     bool{sameparallelpagenunder}%
6782     or bool{prevpgnotnumbered}%
6783   }%
6784   {%
6785     \patchcmd{\thepage}%
6786       {page}{par@page}%
6787       {}%
6788       {\led@error@fail@patch@thepage}%
6789   }{}%
6790 }%
6791
6792 \newcommand{\par@patch@pagenumbering}{%
6793   \setcounter{par@page}{1}%
6794   \par@patch@thepage%
6795 }%
6796
6797 \ifl@dmemoir%
6798   \apptocmd{\@mempnum}%
6799     {\par@patch@pagenumbering}%
6800     {}%

```

```

6801     {\led@error@fail@patch@@mempnum}%
6802
6803 \else%
6804   \apptocmd{\pagenumbering}%
6805     {\par@patch@pagenumbering}%
6806     {}%
6807     {\led@error@fail@patch@pagenumbering}%
6808 \fi%
6809
6810 \AtBeginDocument{\par@patch@thepage}%
6811 %

```

**\@outputpage** As its name says, \@outputpage is a  $\TeX$  macro called in the output routine. It is this macro which increases the page counter. We patch it in order to increase, conditionally, the par@page counter, and to set correctly the page number send to the auxiliaries file (.1, .2, etc.).

```

6812 \AtBeginDocument{%
6813   \set@this@c@page%
6814   \apptocmd{\@outputpage}{%
6815     \ifsameparallelpagelength%
6816       \ifl@dprintingpages%
6817         \ifodd\c@page\else%
6818           \stepcounter{par@page}%
6819         \fi%
6820       \else%
6821         \stepcounter{par@page}%
6822       \fi%
6823     \else%
6824       \stepcounter{par@page}%
6825     \fi%
6826   }%
6827   {}%
6828   {\led@error@fail@patch@@@outputpage}%
6829 }%
6830 \catcode`\#=12%
6831 \AtBeginDocument{%
6832   \apptocmd{\setcounter}{%
6833     \IfStrEq{#1}{page}{\set@this@c@page}{}%
6834   }%
6835   {}%
6836   {\led@error@fail@patch@setcounter}%
6837   \apptocmd{\addtocounter}{%
6838     \IfStrEq{#1}{page}{\set@this@c@page}{}%
6839   }%
6840   {}%
6841   {\led@error@fail@patch@addtocounter}%
6842   \ifpackageloaded{calc}{%
6843     \apptocmd{\stepcounter}{%Because calc.sty does not use addtocounter in
stepcounter

```



```

6844 \IfStrEq{#1}{page}{\set@this@c@page}{}%
6845 }%
6846 {}%
6847 {\led@error@fail@patch@steptocounter}%
6848 }{}%
6849 }%
6850 \catcode\#=#6%
6851 %

```

`\thepar@page` The par@page counter.

```

6852 \newcounter{par@page}%
6853 \setcounter{par@page}{1}%
6854 %

```

## XXV 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.

```

6855 \list@create{\labelref@list}
6856 %

```

`\zz@@@` Two convenience macros to zero three / four labeling counters in one go.

```

6857 \zz@@@ \newcommand*\zz@@@{000|000|000}% Set three counters to zero in one go
6858 \zz@@@ \newcommand*\zz@@@{000|000|000|000}% Set four counters to zero in one go
6859 %
6860 %

```

`\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>34</sup>

<sup>34</sup>The remaining macros in this section were kindly revised by Wayne Sullivan, who substantially improved their efficiency and flexibility.

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>35</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.

```

6861 \newcommand*{\edlabel}[1]{%
6862   \leavevmode%
6863   \@bsphack%
6864   \ifboolexpr{bool{ledRcol} or bool{ledRcol}}{%
6865     \ifXnote@%
6866     \protected@write\@auxout{%
6867       {\string\l@dmake@labelsR\space\thepage|\l@dparsedstartline|\
l@dparsedstartsub|\annot@start|\the\c@pstartR|{#1}}%
6868     \ifdef{\hypertarget}%
6869       {\Hy@raisedlink{\hypertarget{#1}{}}}%
6870       {}%
6871     \else%
6872     \write\linenum@outR{\string\@lab}%
6873     \ifx\labelref@listR\empty%
6874     \xdef\label@refs{\zz@@@}%
6875     \else%
6876     \glp\labelref@listR\to\label@refs%
6877     \fi%
6878     \ifvmode%
6879     \advancelabel@refs%
6880     \fi%
6881   }%

```

Use code from the kernel `\label` command to write the correct page number. Also define an `hypertarget` if `hyperref` package is loaded.

```

6882   \protected@write\@auxout{%
6883     {\string\l@dmake@labelsR\space\thepage|\label@refs|\the\c@pstartR
|{#1}}%
6884   \ifdef{\hypertarget}%
6885     {\Hy@raisedlink{\hypertarget{#1}{}}}%
6886     {}%
6887   \fi%
6888 }{%
6889   \ifXnote@%
6890     \ifl@dpairing%pstart or pstartL?
6891     \protected@write\@auxout{%
6892       {\string\l@dmake@labelsR\space\thepage|\l@dparsedstartline|\
l@dparsedstartsub|\annot@start|\the\c@pstartL|{#1}}%
6893     \ifdef{\hypertarget}%
6894       {\Hy@raisedlink{\hypertarget{#1}{}}}%
6895       {}%
6896     \else%
6897     \protected@write\@auxout{%

```

<sup>35</sup>(jdb43@cam.ac.uk) via the ctt thread ‘ledmac cross referencing’, 25 August 2003.

```

6898      {\string\l@dmake@labels\space\thepage|\l@dparsedstartline|\
l@dparsedstartsub||\annot@start|\the\c@pstart|{#1}}}%
6899      \ifdef{\hypertarget}%
6900      {\Hy@raisedlink{\hypertarget{#1}{}}}%
6901      {}}%
6902      \fi%
6903  \else%
6904      \write\linenum@out{\string\@lab}%
6905      \ifx\labelref@list\empty%
6906          \xdef\label@refs{\zz@@@}%
6907      \else%
6908          \gl@p\labelref@list\to\label@refs%
6909      \fi%
6910      \ifvmode%
6911          \advancelabel@refs%
6912      \fi%
6913      \ifl@dpairing%Pstart or PstartL?
6914      \protected@write\@auxout{}%
6915      {\string\l@dmake@labels\space\thepage|\label@refs|\the\c@pstartL
|{#1}}}%
6916      \ifdef{\hypertarget}%
6917      {\Hy@raisedlink{\hypertarget{#1}{}}}%
6918      {}}%
6919      \else%
6920      \protected@write\@auxout{}%
6921      {\string\l@dmake@labels\space\thepage|\label@refs|\the\c@pstart
|{#1}}}%
6922      \ifdef{\hypertarget}%
6923      {\Hy@raisedlink{\hypertarget{#1}{}}}%
6924      {}}%
6925      \fi%
6926  \fi%
6927  }%
6928  \@esphack}%
6929
6930  %

```

`\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 with the `\advancelabel@refs` command.

```

6931 \newcounter{line}%
6932 \newcounter{subline}%
6933 \newcounter{absline}%
6934 \newcommand{\advancelabel@refs}{%
6935     \setcounter{line}{\expandafter\labelrefsparseline\label@refs}%
6936     \stepcounter{line}%
6937     \setcounter{absline}{\expandafter\labelrefsparseabsline\label@refs}%

```

```

6938 \stepcounter{absline}%
6939 \ifsublines@%
6940 \setcounter{subline}{\expandafter\labelrefsparsesubline\label@refs}
6941 %
6942 \stepcounter{subline}{1}%
6943 \def\label@refs{\theline|\thesubline|\theabsline}%
6944 \else%
6945 \def\label@refs{\theline|0|\theabsline}%
6946 \fi%
6947 }
6948 \def\labelrefsparseline#1|#2|#3{#1}%
6949 \def\labelrefsparsesubline#1|#2|#3{#2}%
6950 \def\labelrefsparseabsline#1|#2|#3{#3}%
6951 %

```

`\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 line number annotation, #6 pstart number, #7 label.

```

6951 \newcommand*{\l@dmake@labels}{%
6952 \def\l@dmake@labels#1|#2|#3|#4|#5|#6|#7{%
6953 \expandafter\ifx\csname the@label\csuse{XR@prefix}#7\endcsname%
6954 \relax%
6955 \else%
6956 \led@warn@DuplicateLabel{\csuse{XR@prefix}#7}%
6957 \fi
6958 \global\providetoggle{label@#7\ledRcol}%False is the default value of
this toggle, which tells us whether a label is linked to the right or left
side
6959 \csgdef{the@label\csuse{XR@prefix}#7}{#1|#2|#3|#4|#5|#6|\relax}%
6960 \ignorespaces}
6961 %
6962 %

```

ℒ<sub>TEX</sub> 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.

```

6963 \AtBeginDocument{%
6964 \def\l@dmake@labels#1|#2|#3|#4|#5|#6|#7{%
6965 }
6966 %
6967 %

```

`\@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`.

```

6968 \newcommand*\@lab{%
6969   \ifledRcol
6970     \xright@appenditem{\linenumr@p{\line@numR}}|{%
6971       \ifsublines@ \sublinenumr@p{\subline@numR}\else 0\fi|{%
6972       \the\absline@numR|{%
6973       \current@annot%
6974       }%
6975       \to\labelref@listR
6976   \else
6977     \xright@appenditem{%
6978       \linenumr@p{\line@num}|{%
6979       \ifsublines@ \sublinenumr@p{\subline@num}\else 0\fi|{%
6980       \the\absline@num|{%
6981       \current@annot%
6982       }%
6983       \to\labelref@list
6984   \fi}
6985 %
6986 %

```

`\applabel` `\applabel`, if called in `\edtext` will insert automatically both a starting and an ending label for the current edtext lines.

```

6987 \newcommand*\applabel}[1]{%
6988   \if@edtext@secondarg%
6989   %

```

Label should not be already defined.

```

6990   \ifcsundef{the@label#1}{%
6991     \csdef{the@label#1}{applabel}%
6992   }%
6993   {%
6994     \led@warn@DuplicateLabel{#1 (applabel)}%
6995   }%
6996 %

```

Parse the `\edtext` line numbers and annotations.

```

6997   \expandafter\l@dp@rsefootspec\l@d@nums|{%
6998   \expandafter\parse@annot\l@current@annot|{%
6999   %

```

Use the  $\TeX$  standard hack for label.

```
7000 \bsphack%
7001 %
```

And now, write the data in the auxiliary file.

```
7002 \ifledRcol%
7003 \protected@write\@auxout{}%
7004 {\string\l@dmake@labelsR\space\l@dparsedstartpage|\l@dparsedstartline|\l@dparsedstartsub|\annot@start|\the\c@pstartR|{#1:
start}}}%
7005 \ifdef\hypertarget{%
7006 {\Hy@raisedlink{\hypertarget{#1:start}}}}%
7007 {}%
7008 \protected@write\@auxout{}%
7009 {\string\l@dmake@labelsR\space\l@dparsedendpage|\l@dparsedendline
|\l@dparsedendsub|\annot@end|\the\c@pstartR|{#1:end}}}%
7010 \else%
7011 \ifl@dpairing%pstart or pstartL?
7012 \protected@write\@auxout{}%
7013 {\string\l@dmake@labels\space\l@dparsedstartpage|\l@dparsedstartline|\l@dparsedstartsub|\annot@start|\the\c@pstartL|{#1:
start}}}%
7014 \ifdef\hypertarget{%
7015 {\Hy@raisedlink{\hypertarget{#1:start}}}}%
7016 {}%
7017 \protected@write\@auxout{}%
7018 {\string\l@dmake@labels\space\l@dparsedendpage|\l@dparsedendline|\l@dparsedendsub|\annot@end|\the\c@pstartL|{#1:end}}}%
7019 \else%
7020 \protected@write\@auxout{}%
7021 {\string\l@dmake@labels\space\l@dparsedstartpage|\l@dparsedstartline|\l@dparsedstartsub|\annot@start|\the\c@pstart|{#1:start
}}}%
7022 \ifdef\hypertarget{%
7023 {\Hy@raisedlink{\hypertarget{#1:start}}}}%
7024 {}%
7025 \protected@write\@auxout{}%
7026 {\string\l@dmake@labels\space\l@dparsedendpage|\l@dparsedendline|\l@dparsedendsub|\annot@end|\the\c@pstart|{#1:end}}}%
7027 \fi%
7028 \fi%
7029 %
```

Use the  $\TeX$  standard hack for label.

```
7030 \esphack%
7031 %
```

Warning if `\apptlabel` is called outside of `\edtext`.

```
7032 \else%
```

```

7033 \led@warn@AppLabelOutSecondArgEdtext{#1}%
7034 \fi%
7035 %

End of \applabel

7036 }%
7037 %

```

**\edlabels** `\edlabels` and `\edlabelE` are just used to mark the beginning and the end of a passage.

```

7038 \edlabelE
\edlabelSE \newcommand{\edlabels}[1]{%
7039 \edlabel{#1:start}%
7040 }
7041 \newcommand{\edlabelE}[1]{%
7042 \edlabel{#1:end}%
7043 }
7044 \newcommand{\edlabelSE}[1]{%
7045 \edlabels{#1}%
7046 \edlabelE{#1}%
7047 }
7048 %

```

**\wrap@edcrossref** `\wrap@edcrossref` is called around all `reledmac` crossref commands, except those which start with `x`. It adds the hyperlink.

```

7049 \newrobustcmd{\wrap@edcrossref}[2]{%
7050 \ifdef{\hyperlink}%
7051 {\hyperlink{#1}{#2}}%
7052 {#2}%
7053 }
7054 %

```

**\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.

ØT<sub>E</sub>X already defines a `\pageref`, so changing the name to `\edpageref`.

```

7055 \newcommand*{\edpageref}[1]{\l@dref@undefined{#1}\wrap@edcrossref{#1}{\l@dgetref@num{1}{#1}}}
7056 \newcommand*{\xpageref}[1]{\l@dgetref@num{1}{#1}}
7057
7058 %

```

**\edlineref** If the specified label exists, `\lineref` gives its line number.

**\xlineref**

```

7059 \newcommand*{\edlineref}[1]{%
7060   \l@dref@undefined{#1}%
7061   \wrap@edcrossref{#1}{%
7062     \providetoggle{label@#1@ledRcol}%Required for the first run, when the
label has not yet been parsed on the .aux file
7063     \iftoggle{label@#1@ledRcol}%
7064       {\linenumrepR{\l@dgetref@num{2}{#1}}}%
7065       {\linenumrep{\l@dgetref@num{2}{#1}}}%
7066     \xflagref{#1}%
7067   }%
7068 }%
7069 \newcommand*{\xlineref}[1]{\l@dgetref@num{2}{#1}}%
7070
7071 %

```

**\sublineref** If the specified label exists, \sublineref gives its sub-line number.

```

\sublineref
\sublineref
7072 \newcommand*{\sublineref}[1]{%
7073   \l@dref@undefined{#1}%
7074   \wrap@edcrossref{#1}{%
7075     \providetoggle{label@#1@ledRcol}%Required for the first run, when the
label has not yet been parsed on the .aux file
7076     \iftoggle{label@#1@ledRcol}%
7077       {\sublinenumrepR{\l@dgetref@num{3}{#1}}}%
7078       {\sublinenumrep{\l@dgetref@num{3}{#1}}}%
7079     }%
7080   }%
7081 \newcommand*{\xsublineref}[1]{\l@dgetref@num{3}{#1}}
7082
7083 %

```

**\xabslineref** If the specified label exists, \xabslineref gives its absolute line number. This is generally used only by reledmac internal macros.

```

7084 \newcommand*{\xabslineref}[1]{\l@dgetref@num{4}{#1}}%
7085 %

```

**\annotationref** If the specified label exists, \edannotationref gives the line number annotation with which it is associated.

**\xannotationref**

```

7086 \newcommand*{\annotationref}[1]{%
7087   \l@dref@undefined{#1}%
7088   \wrap@edcrossref{#1}{%
7089     \l@wrap@ifnotemptybox{\Xwraplinenumannotation@ref}{%
7090       \l@dgetref@num{5}{#1}%
7091     }%
7092   }%
7093 }%
7094 \newcommand*{\xannotationref}[1]{%
7095   \l@dgetref@num{5}{#1}%

```



```

7096 }%
7097 %

```

**\pstarteref** If the specified label exists, \pstarteref gives its pstart number.

```

\pstarteref
7098 \newcommand*\pstarteref[1]{\l@dref@undefined{#1}\wrap@edcrossref{#1}{\
l@dgetref@num{6}{#1}}}%
7099 \newcommand*\xpstarteref[1]{\l@dgetref@num{6}{#1}}%
7100 %
7101 %

```

**\xflagref** \xflagref finds the side flag of any ref defined with \edlabel.

```

7102 \newcommand*\xflagref[1]{\l@dgetref@num{7}{#1}}%
7103 %

```

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.

```

7104 \newcommand*\l@dref@undefined[1]{%
7105 \expandafter\ifx\csname the@label#1\endcsname\relax
7106 \led@warn@RefUndefined{#1}%
7107 \fi}
7108 %
7109 %

```

**\l@dgetref@num** Next, \l@dgetref@num fetches the number we want. It has two arguments: the first is simply a digit, specifying whether to fetch a page (1), line (2), sub-line (3), (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.

```

7110 \newcommand*\l@dgetref@num[2]{%
7111 \expandafter
7112 \ifx\csname the@label#2\endcsname \relax
7113 000%
7114 \else
7115 \expandafter\expandafter\expandafter
7116 \l@dlabel@parse\csname the@label#2\endcsname|#1%
7117 \fi}
7118 %
7119 %

```

`\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 argument. The `|`-delimited arguments consist of the expanded label-macro (three reference numbers), followed by the switch-number (1, 2, 3, 4, 5, 6) which defines which of the former seven numbers to extract. (It was given earlier as the first argument of `\l@dgetref@num`.)

```

7120 \newcommand*{\l@dlabel@parse}{%
7121 \def\l@dlabel@parse#1|#2|#3|#4|#5|#6|#7|#8{%
7122   \ifcase #8%
7123   \or #1%
7124   \or #2%
7125   \or #3%
7126   \or #4%
7127   \or #5%
7128   \or #6%
7129   \or #7%
7130   \fi}
7131 %

```

`\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.

```

7132 \newcommand*{\xxref}[2]{%
7133   {%
7134     \expandafter\ifx\csname the@label#1\endcsname\relax%
7135       \let\@tempa\zz@@@%
7136       \def\@tempc{%
7137         \else%
7138         \def\@tempa{%
7139           \l@dgetref@num{1}{#1}|%
7140           \l@dgetref@num{2}{#1}|%
7141           \l@dgetref@num{3}{#1}%
7142         }%
7143         \edef\@tempc{%
7144           \l@dgetref@num{5}{#1}%
7145         }%
7146       \fi%
7147     \expandafter\ifx\csname the@label#2\endcsname\relax%
7148       \let\@tempb\zz@@@%
7149       \def\@tempd{%
7150         \else%

```

```

7151 \def\@tempb{%
7152 \l@getref@num{1}{#2}|%
7153 \l@getref@num{2}{#2}|%
7154 \l@getref@num{3}{#2}%
7155 }%
7156 \edef\@tempd{%
7157 \l@getref@num{5}{#2}%
7158 }%
7159 \fi%
7160 \global\appto\@beforeinsertofthisedtext{%
7161 \def\@this@crossref@start{#1}%
7162 }%
7163 \global\appto\@beforeinsertofthisedtext{%
7164 \def\@this@crossref@end{#2}%
7165 }%
7166 \linenum{\@tempa|\@tempb}%
7167 \edef\@tempe{\@tempc|\@tempd}%
7168 \expandafter\lineannot\expandafter{\@tempe}%
7169 }%
7170 }%
7171
7172
7173 %

```

**\appref** \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).

**\Seref** 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.

```

7174 \def\Xtwolines@appref{%
7175 \def\Xtwolines@Seref{%
7176
7177 \def\Xmorethantwolines@appref{%
7178 \def\Xmorethantwolines@Seref{%
7179
7180 \def\Xlinerangeseparator@appref{\endashchar}%
7181 \def\Xlinerangeseparator@Seref{\endashchar}%
7182
7183 \def\Xsublinesep@appref{\fullstop}%
7184 \def\Xsublinesep@Seref{\fullstop}%
7185
7186 \def\Xpagelinesep@appref{\fullstop}%
7187 \def\Xpagelinesep@Seref{\fullstop}%
7188
7189
7190 \newtoggle{Xtwolinesbutnotmore@appref}%

```

```

7191 \newtoggle{Xtwolinesbutnotmore@SEref}%
7192
7193 \newtoggle{Xtwolinesonlyinsamepage@appref}%
7194
7195 \newtoggle{Xtwolinesonlyinsamepage@SEref}%
7196
7197 \newtoggle{Xlineflag@appref}%
7198 \toggletrue{Xlineflag@appref}%Here exception
7199 \newtoggle{Xlineflag@SEref}%
7200 \toggletrue{Xlineflag@SEref}%%Here exception
7201
7202 \newtoggle{Xlinenumannotationonlyfirst@SEref}%Never changed, just for the
test in \setprintlines
7203 \newtoggle{Xlinenumannotationonlyfirst@appref}%Idem
7204
7205 \def\Xlinenumannotationposition@appref{after}%
7206 \def\Xlinenumannotationposition@SEref{after}%
7207
7208 \def\Xwraplinenumannotation@appref{\textsuperscript}%
7209 \def\Xwraplinenumannotation@SEref{\textsuperscript}%
7210 \newtoggle{Xnoidenticallinenumannotation@appref}%
7211 \newtoggle{Xnoidenticallinenumannotation@SEref}%
7212
7213 \def\Xendtwolines@apprefwithpage{}%
7214 \def\Xendtwolines@SErefwithpage{}%
7215
7216 \def\Xendmorethantwolines@apprefwithpage{}%
7217 \def\Xendmorethantwolines@SErefwithpage{}%
7218
7219 \def\Xendlinangeseparator@apprefwithpage{\endashchar}
7220 \def\Xendlinangeseparator@SErefwithpage{\endashchar}
7221 \def\Xendlinangeseparator@SErefonlypage{\endashchar}
7222
7223 \def\Xendbeforepagenumber@apprefwithpage{p.}%
7224 \def\Xendbeforepagenumber@SErefwithpage{p.}%
7225 \def\Xendbeforepagenumber@SEonlypage{p.}%
7226
7227 \def\Xendafterpagenumber@apprefwithpage{} }%
7228 \def\Xendafterpagenumber@SErefwithpage{} }%
7229
7230
7231 \def\Xendlineprefixsingle@apprefwithpage{}%
7232 \def\Xendlineprefixsingle@SErefwithpage{}%
7233
7234 \def\Xendlineprefixmore@apprefwithpage{}%
7235 \def\Xendlineprefixmore@SErefwithpage{}%
7236
7237 \newtoggle{Xendtwolinesbutnotmore@apprefwithpage}%
7238 \newtoggle{Xendtwolinesbutnotmore@SErefwithpage}%
7239

```

```

7240 \def\Xendsublinesep@apprefwithpage{\fullstop}%
7241 \def\Xendsublinesep@SErefwithpage{\fullstop}%
7242
7243 \newtoggle{Xendtwolinesonlyinsamepage@apprefwithpage}%
7244 \newtoggle{Xendtwolinesonlyinsamepage@SErefwithpage}%
7245
7246 \newtoggle{Xendlineflag@apprefwithpage}
7247 \toggletrue{Xendlineflag@apprefwithpage}%Here, exception
7248 \newtoggle{Xendlineflag@SErefwithpage}
7249 \toggletrue{Xendlineflag@SErefwithpage}%Here, exception
7250
7251
7252 \def\Xendlinenumannotationposition@apprefwithpage{after}%
7253 \def\Xendlinenumannotationposition@SErefwithpage{after}%
7254
7255 \def\Xendlinenumannotationposition@apprefwithpage{after}%
7256 \def\Xendlinenumannotationposition@apprefwithpage@SErefwithpage{after}%
7257
7258 \def\Xendwraplinenumannotation@apprefwithpage{\textsuperscript}%
7259 \def\Xendwraplinenumannotation@SErefwithpage{\textsuperscript}%
7260
7261 \newtoggle{Xendnoidenticallinenumannotation@apprefwithpage}%
7262 \newtoggle{Xendnoidenticallinenumannotation@SErefwithpage}%
7263 %

```

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.

```

7264
7265 \newtoggle{Xendlinenumannotationonlyfirst@SEref}%Never changed, just for
the test in \setprintendlines
7266 \newtoggle{Xendlinenumannotationonlyfirst@SErefwithpage}%Idem
7267 \newtoggle{Xendlinenumannotationonlyfirst@appref}%Idem
7268 \newtoggle{Xendlinenumannotationonlyfirst@apprefwithpage}%Idem
7269
7270 \gdef\Xboxstartlinenum@appref{Opt}
7271 \gdef\Xboxstartlinenum@SEref{Opt}
7272
7273 \gdef\Xboxendlinenum@appref{Opt}
7274 \gdef\Xboxendlinenum@SEref{Opt}
7275
7276 \gdef\Xendboxstartlinenum@apprefwithpage{Opt}
7277 \gdef\Xendboxstartlinenum@SErefwithpage{Opt}
7278
7279 \gdef\Xendboxendlinenum@apprefwithpage{Opt}
7280 \gdef\Xendboxendlinenum@SErefwithpage{Opt}
7281
7282 \newtoggle{Xendpagenumberonlyfirst@apprefwithpage}
7283 \newtoggle{Xendpagenumberonlyfirst@SErefwithpage}
7284

```

```

7285 \newtoggle{Xendpagenumberonlyfirsttifsingl@apprefwithpage}
7286 \newtoggle{Xendpagenumberonlyfirsttifsingl@Serefwithpage}
7287
7288 \newtoggle{Xendpagenumberonlyfirstinttwo@apprefwithpage}
7289 \newtoggle{Xendpagenumberonlyfirstinttwo@Serefwithpage}
7290
7291 \gdef\Xendsympagenum@apprefwithpage{}
7292 \gdef\Xendsympagenum@Serefwithpage{}
7293
7294 \gdef\Xendinplaceofpagenumber@apprefwithpage{}
7295 \gdef\Xendinplaceofpagenumber@Serefwithpage{}
7296
7297 %

```

Now, declare the default values of \@apprefprefixsingle and \@apprefprefixmore, \@Serefprefix, \@Serefprefixmore and the commands which defines them.

```

7298 \newcommand\@apprefprefixsingle{}%
7299 \newcommand\@Serefprefixsingle{}%
7300
7301 \newcommand\@apprefprefixmore{}%
7302 \newcommand\@Serefprefixmore{}%
7303
7304 \newcommand{\setapprefprefixsingle}[1]{%
7305   \gdef\@apprefprefixsingle{#1}%
7306 }
7307 \newcommand{\setSerefprefixsingle}[1]{%
7308   \gdef\@Serefprefixsingle{#1}%
7309 }
7310
7311 \newcommand{\setapprefprefixmore}[1]{%
7312   \gdef\@apprefprefixmore{#1}%
7313 }
7314 \newcommand{\setSerefprefixmore}[1]{%
7315   \gdef\@Serefprefixmore{#1}%
7316 }
7317
7318 %

```

And not \setSerefonlypageprefixsingle and \setSerefonlypageprefixmore.

```

7319 \newcommand{\setSerefonlypageprefixsingle}[1]{%
7320   \gdef\Serefonlypage@prefixsingle{#1}%
7321 }%
7322 \newcommand{\setSerefonlypageprefixmore}[1]{%
7323   \gdef\Serefonlypage@prefixmore{#1}%
7324 }%
7325 %

```

And now, the main commands: \appref, \apprefwithpage, \Seref and \Serefwithpage. These commands call \refformated@ and \refformatedwithpage, which calls \printlines

and `\printendlines`. That is why we have previously declared all hooks values tested inside these last commands.

```

7326
7327 \newcommandx{\appref}[2][1,usedefault]{\reformatted@{#1}{#2}{appref}}
7328 \newcommandx{\SEref}[2][1,usedefault]{\reformatted@{#1}{#2}{SEref}}
7329
7330 \newcommandx{\apprefwithpage}[2][1,usedefault]{\reformattedwithpage@
7331 {#1}{#2}{appref}}
7332 \newcommandx{\SErefwithpage}[2][1,usedefault]{\reformattedwithpage@
7333 {#1}{#2}{SEref}}
7334 \newcommandx{\SErefonlypage}[2][1,usedefault]{\reformattedonlypage@
7335 {#1}{#2}{SEref}}
7336
7337 \newcommand{\reformatted@}[3]{%
7338   \def\do##1{%
7339     \setkeys[mac]{truefootnoteoption}{##1}%
7340     }%
7341     \notblank{#1}{\docsvlist{#1}}{ }%
7342     \xdef\@currentseries{#3}%
7343     \iftoggle{noprefix@}{ }%
7344     {%
7345       \ifcsemt{#3}{prefixmore}%
7346       {\@apprefprefixsingle}%
7347       }%
7348       \IfEq{\xlineref{#2:start}}{\xlineref{#2:end}}{%
7349         \iftoggle{prefixmore@}{%
7350           {\csuse{#3prefixmore}}%
7351           {\csuse{#3prefixsingle}}%
7352         }%
7353       }%
7354     }%
7355     \ifboolexpr{%
7356       test{\ifcsundef{the@label#2:start}}%
7357       or test{\ifcsundef{the@label#2:end}}%
7358     }%
7359     {\led@warn@pairRefUndefined{#2}\nfss@text{\reset@font\bfseries ??}}%
7360     {%
7361       \def\@this@crossref@start{#2:start}%
7362       \def\@this@crossref@end{#2:end}%
7363       \xdef\annot@start{\xannotationref{#2:start}}%
7364       \xdef\annot@end{\xannotationref{#2:end}}%
7365       \printlines\xpageref{#2:start}|\xlineref{#2:start}|\xsublineref{#2:
7366 start}|\xpageref{#2:end}|\xlineref{#2:end}|\xsublineref{#2:end}|\relax|\
7367 xflagref{#2:start}|}%
7368     }%
7369     \def\do##1{%
7370       \setkeys[mac]{falsefootnoteoption}{##1}%

```

```

7368 }%
7369 \notblank{#1}{\docsvlist{#1}}{ }%
7370 }%
7371
7372 \newcommand{\reformattedwithpage@}[3]{%
7373   \def\do##1{%
7374     \setkeys[mac]{truefootnoteoption}{##1}%
7375   }%
7376   \notblank{#1}{\docsvlist{#1}}{ }%
7377   \xdef\@currentseries{#3withpage}%
7378   \ifboolexpr{%
7379     test{\ifcsundef{the@label#2:start}}%
7380     or test{\ifcsundef{the@label#2:end}}%
7381   }%
7382   {\led@warn@pairRefUndefined{#2}\nfss@text{\reset@font\bfseries ??}}%
7383   {%
7384     \def\@this@crossref@start{#2:start}%
7385     \def\@this@crossref@end{#2:end}%
7386     \xdef\annot@start{\xannotationref{#2:start}}%
7387     \xdef\annot@end{\xannotationref{#2:end}}%
7388     \printendlines\xpageref{#2:start}|\xlineref{#2:start}|\xsublineref{#2:
start}|\xpageref{#2:end}|\xlineref{#2:end}|\xsublineref{#2:end}|\relax|
xflagref{#2:start}}%
7389   }%
7390   \def\do##1{%
7391     \setkeys[mac]{falsefootnoteoption}{##1}%
7392   }%
7393   \notblank{#1}{\docsvlist{#1}}{ }%
7394 }%
7395
7396 \newcommand{\reformattedonlypage@}[3]{%
7397   \def\do##1{%
7398     \setkeys[mac]{truefootnoteoption}{##1}%
7399   }%
7400   \notblank{#1}{\docsvlist{#1}}{ }%
7401   \xdef\@currentseries{#3onlypage}%
7402   \ifboolexpr{%
7403     test{\ifcsundef{the@label#2:start}}%
7404     or test{\ifcsundef{the@label#2:end}}%
7405   }%
7406   {\led@warn@pairRefUndefined{#2}\nfss@text{\reset@font\bfseries ??}}%
7407   {\ifnumequal{\xpageref{#2:end}}{\xpageref{#2:start}}%
7408     {%
7409       \ifcsvoid{#3onlypage@prefixsingle}%
7410       {}%
7411       {\csletcs{Xendbeforepagenumber@#3onlypage}{#3onlypage@prefixsingle
}}%
7412     }%
7413     \printnpnum{%
7414       \wrap@edcrossref{#2:start}{\xpageref{#2:start}}%
7415     }%

```



```

7415 }%
7416 {%
7417   \ifcsvoid{#3onlypage@prefixmore}%
7418   {}%
7419   {\csletcs{Xendbeforepagenumber@#3onlypage}{#3onlypage@prefixmore}}%
7420   \ifdefined\linangesep%
7421     \printnpnum{%
7422       \wrap@edcrossref{#2:start}{\xpageref{#2:start}}%
7423       \linangesep%
7424       \wrap@edcrossref{#2:end}{\xpageref{#2:end}}%
7425     }%
7426   \else%
7427     \printnpnum{%
7428       \wrap@edcrossref{#2:start}{\xpageref{#2:start}}%
7429       \csuse{Xendlinangeseparator@}\currentseries%
7430       \wrap@edcrossref{#2:end}{\xpageref{#2:end}}%
7431     }%
7432   \fi%
7433 }%
7434 }%
7435 \def\do##1{%
7436   \setkeys[mac]{falsefootnoteoption}{##1}%
7437 }%
7438 \notblank{#1}{\docsvlist{#1}}{}%
7439 }%
7440 %

```

`\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`.

```

7441 \newcommand*{\edmakelabel}[2]{\expandafter\def\csname the@label#1\
7442   endcsname{#2}}
7443 %

```

(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. 147 and V.9 p. 111), since `\xxref` makes a call to `\linenum` in order to do its work.)

## XXV.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`.

```
7444 \pretocmd{\XR@test}%
7445   {\XR@test@mac+++#1#2#3#4+++}%
7446   {}%
7447   {}%
7448   %
```

`\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`.

```
7449 \long\def\xR@test@mac+++#1+++{\XR@test@mac@test#1}
7450 %
```

`\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`.

```
7451 \long\def\xR@test@mac@test#1#2...{%The triple dots (NOT \ldots) are because
of the line 22 of xr.sty v5.02 1994/05/28
7452   \ifx#1\l@dmake@labels%
7453     \l@dmake@labels#2%
7454   \else
7455     \ifx#1\l@dmake@labelsR%
7456       \l@dmake@labelsR #2%
7457   \fi%
7458   \fi%
7459 }%
7460 %
```

## XXVI Sidenotes

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.

```
7461 \pretocmd{\@xympar}%
7462   {\ifnumberedpar@
7463     \led@warn@NoMarginpars
7464     \@esphack
7465   \else}%
7466   {}%
7467   {}%
7468
7469 \apptocmd{\@xympar}%
7470   {\fi}%
```

```

7471 {}
7472 {}
7473
7474 %

```

We provide sidenotes 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 sidenote margin:

**left:** 0

**right:** 1

**outer:** 2

**inner:** 3

```

7475 \newcount\sidenote@margin
7476 \newcommand*{\sidenotemargin}[1]{\%
7477   \l@dgetsidenote@margin{#1}%
7478   \ifnum\@l@dttempcntb>\m@ne
7479     \ifledRcol
7480       \global\sidenote@marginR=\@l@dttempcntb
7481     \else
7482       \global\sidenote@margin=\@l@dttempcntb
7483     \fi
7484   \fi}}
7485 \newcommand*{\l@dgetsidenote@margin}[1]{\%
7486   \def\@tempa{#1}\def\@tempb{left}%
7487   \ifx\@tempa\@tempb
7488     \@l@dttempcntb \z@
7489   \else
7490     \def\@tempb{right}%
7491     \ifx\@tempa\@tempb
7492       \@l@dttempcntb \@ne
7493     \else
7494       \def\@tempb{outer}%
7495       \ifx\@tempa\@tempb
7496         \@l@dttempcntb \tw@
7497       \else
7498         \def\@tempb{inner}%
7499         \ifx\@tempa\@tempb
7500           \@l@dttempcntb \thr@@
7501         \else
7502           \led@warn@BadSidenotemargin
7503           \@l@dttempcntb \m@ne
7504         \fi

```

```

7505     \fi
7506     \fi
7507     \fi}
7508 \sidenotemargin{right}
7509
7510 %

```

`\l@dlp@rbox` We need two boxes to store sidenote texts.

```

\l@drp@rbox
7511 \newbox\l@dlp@rbox
7512 \newbox\l@drp@rbox
7513
7514 %

```

`\ledlsnotewidth` These specify the width of the left/right boxes (initialised to `\marginparwidth`), their distance from the text (initialised to `\linenumsep`), and the fonts used.

```

\ledrsnotewidth
\ledlsnotesep
7515 \newdimen\ledlsnotewidth \ledlsnotewidth=\marginparwidth
\ledrsnotesep
7516 \newdimen\ledrsnotewidth \ledrsnotewidth=\marginparwidth
\ledlsnotefontsetup
7517 \newdimen\ledlsnotesep \ledlsnotesep=\linenumsep
\ledrsnotefontsetup
7518 \newdimen\ledrsnotesep \ledrsnotesep=\linenumsep
7519 \newcommand*\ledlsnotefontsetup{\raggedleft\footnotesize}
7520 \newcommand*\ledrsnotefontsetup{\raggedright\footnotesize}
7521
7522 %

```

`\ledleftnote` `\ledrightnote`, `\ledinnernote`, `\ledouternote` are the user commands for left, right, inner and outer sidenotes. The two last one are just alias for the two first one, depending of the page number. `\ledsidenote{<text>}` is the command for a moveable sidenote.

```

\ledsidenote
7523 \newcommand*\ledleftnote[1]{\edtext{}\l@dlsnote{#1}}
7524 \newcommand*\ledrightnote[1]{\edtext{}\l@drsnote{#1}}
7525 \newcommand*\ledsidenote[1]{\edtext{}\l@dcnote{#1}}%
7526 \newcommand*\ledinnernote[1]{\edtext{}\l@disnote{#1}}%
7527 \newcommand*\ledouternote[1]{\edtext{}\l@dosnote{#1}}%
7528 %

```

`\l@dlsnote` . The ‘footnotes’ for left, right, and moveable sidenotes. The whole scheme is reminiscent of the critical footnotes code.

```

\l@drsnote
\l@dcnote
7529 \newif\ifrighnoteup
\l@desnote
7530 \righnoteuptrue
\l@disnote
7531
7532 \newcommand*\l@dlsnote[1]{%
7533   \begingroup%
7534   \newcommand{\content}{#1}%
7535   \ifnumberedpar@
7536     \ifledRcol%
7537       \xright@appenditem{\noexpand\l@dlsnote{\expandonce\content}}%

```

```

7538         \to\inserts@listR
7539     \global\advance\insert@countR \@ne%
7540 \else%
7541     \xright@appenditem{\noexpand\vl@dlsnote{\expandonce\content}}}%
7542         \to\inserts@list
7543     \global\advance\insert@count \@ne%
7544 \fi
7545 \fi%
7546 \ignorespaces%
7547 \endgroup%
7548 }%
7549
7550 \newcommand*{\l@drsnote}[1]{%
7551     \begingroup%
7552     \newcommand{\content}{#1}%
7553     \ifnumberedpar@
7554         \ifledRcol%
7555             \xright@appenditem{\noexpand\vl@drsnote{\expandonce\content}}}%
7556                 \to\inserts@listR
7557             \global\advance\insert@countR \@ne%
7558         \else%
7559             \xright@appenditem{\noexpand\vl@drsnote{\expandonce\content}}}%
7560                 \to\inserts@list
7561             \global\advance\insert@count \@ne%
7562         \fi
7563     \fi\ignorespaces%
7564 \endgroup%
7565 }%
7566
7567 \newcommand*{\l@dcsnote}[1]{%
7568     \begingroup%
7569     \newcommand{\content}{#1}%
7570     \ifnumberedpar@
7571         \ifledRcol%
7572             \xright@appenditem{\noexpand\vl@dcsnote{\expandonce\content}}}%
7573                 \to\inserts@listR
7574             \global\advance\insert@countR \@ne%
7575         \else%
7576             \xright@appenditem{\noexpand\vl@dcsnote{\expandonce\content}}}%
7577                 \to\inserts@list
7578             \global\advance\insert@count \@ne%
7579         \fi
7580     \fi\ignorespaces%
7581 \endgroup%
7582 }%
7583
7584 \newcommand*{\l@disnote}[1]{%
7585     \begingroup%
7586     \newcommand{\content}{#1}%
7587     \ifnumberedpar@%

```

```

7588 \ifledRcol%
7589 \xright@appenditem{\noexpand\vl@disnote{\expandonce\content}}}%
7590 \to\inserts@listR%
7591 \global\advance\insert@countR \@ne%
7592 \else%
7593 \xright@appenditem{\noexpand\vl@disnote{\expandonce\content}}}%
7594 \to\inserts@list%
7595 \global\advance\insert@count \@ne%
7596 \fi%
7597 \fi\ignorespaces%
7598 \endgroup%
7599 }%
7600
7601 \newcommand*{\l@dosnote}[1]{%
7602 \begingroup%
7603 \newcommand{\content}{#1}%
7604 \ifnumberedpar%
7605 \ifledRcol%
7606 \xright@appenditem{\noexpand\vl@dosnote{\expandonce\content}}}%
7607 \to\inserts@listR%
7608 \global\advance\insert@countR \@ne%
7609 \else%
7610 \xright@appenditem{\noexpand\vl@dosnote{\expandonce\content}}}%
7611 \to\inserts@list%
7612 \global\advance\insert@count \@ne%
7613 \fi%
7614 \fi\ignorespaces%
7615 \endgroup%
7616 }%
7617
7618 %

```

`\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 sidenotes. 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.

```

7619 \newcommand*{\vl@dlsnote}[1]{%
7620   \ifledRcol{%
7621     \@l@tempcntb=\sidenote@marginR%
7622     \ifnum\@l@tempcntb>\@ne%
7623       \advance\@l@tempcntb by\page@numR%
7624     \fi%
7625   \else%
7626     \@l@tempcntb=\sidenote@margin%
7627     \ifnum\@l@tempcntb>\@ne%
7628       \advance\@l@tempcntb by\page@num%
7629     \fi%
7630   \fi%
7631   \ifodd\@l@tempcntb%
7632     \listgadd{\l@dcsnotetext@l}{#1}%
7633   \else%
7634     \listgadd{\l@dcsnotetext}{#1}%
7635   \fi
7636 }
7637 \newcommand*{\vl@drsnote}[1]{%
7638   \ifledRcol{%
7639     \@l@tempcntb=\sidenote@marginR%
7640     \ifnum\@l@tempcntb>\@ne%
7641       \advance\@l@tempcntb by\page@numR%
7642     \fi%
7643   \else%
7644     \@l@tempcntb=\sidenote@margin%
7645     \ifnum\@l@tempcntb>\@ne%
7646       \advance\@l@tempcntb by\page@num%
7647     \fi%
7648   \fi%
7649   \ifodd\@l@tempcntb%
7650     \listgadd{\l@dcsnotetext}{#1}%
7651   \else%
7652     \listgadd{\l@dcsnotetext@r}{#1}%
7653   \fi%
7654 }
7655 \newcommand*{\vl@dcsnote}[1]{\listgadd{\l@dcsnotetext}{#1}}
7656
7657 \newcommand{\vl@disnote}[1]{%
7658   \ifledRcol{%
7659     \@tempcnta=\page@numR%
7660   \else%
7661     \@tempcnta=\page@num%
7662   \fi%
7663   \ifodd\@tempcnta% ODD => right page => inner side = left side
7664     \vl@dlsnote{#1}%
7665   \else%
```

```

7666 \vl@drsnote{#1}%
7667 \fi%
7668 }%
7669
7670 \newcommand{\vl@dosnote}[1]{%
7671 \ifledRcol%
7672 \@tempcnta=\page@numR%
7673 \else%
7674 \@tempcnta=\page@num%
7675 \fi%
7676 \ifodd\@tempcnta% ODD => right page => outer side = right side
7677 \vl@drsnote{#1}%
7678 \else%
7679 \vl@dlsnote{#1}%
7680 \fi%
7681 }%
7682
7683 %

```

`\setl@dlp@rbox` `\setl@dlprbox{<lednums>}{<tag>}{<text>}` puts `<text>` into the `\l@dlp@rbox` box. And similarly for the right side box. It is these boxes that finally get displayed in the margins.

```

7684 \newcommand*{\setl@dlp@rbox}[1]{%
7685 \begingroup%
7686 \parindent\z@\hsize=\ledlsnotewidth%
7687 \ledlsnotefontsetup%We kept it outside of the vbox, because can affect
the ragging
7688 \global\setbox\l@dlp@rbox%
7689 \ifleftnoteup%
7690 =\vbox to\z@{\ledlsnotefontsetup\vss #1}}%We put \
ledlsnotefontsetup inside footnote because required for color command. Note
the {} to keep setting local.
7691 \else%
7692 =\vbox to 0.70\baselineskip{\ledlsnotefontsetup\strut#1\vss}}%
7693 \fi%
7694 \endgroup%
7695 }
7696
7697 \newcommand*{\setl@drp@rbox}[1]{%
7698 \begingroup%
7699 \parindent\z@\hsize=\ledrsnotewidth%
7700 \ledrsnotefontsetup%We kept it outside of the vbox, because can affect
the ragging
7701 \global\setbox\l@drp@rbox%
7702 \ifrightrightnoteup%
7703 =\vbox to\z@{\ledrsnotefontsetup\vss#1}}%We put \ledrsnotefontsetup
inside footnote because required for color command. Note the {} to keep
setting local.
7704 \else%

```



```

7705     =\vbox to0.7\baselineskip{{\ledrsnotefontsetup\strut#1\vss}}}%
7706     \fi%
7707     \endgroup%
7708 }%
7709 \newif\ifleftnoteup
7710 \leftnoteuptrue
7711 %

```

**\@sidenotesep** This macro is used to separate sidenotes of the same line.

```

7712 \newcommand{\setsidenotesep}[1]{\gdef\@sidenotesep{#1}}
7713 \newcommand{\@sidenotesep}{, }
7714 %

```

**\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 `\affixline@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.

```

7715 \newcommand*{\affixside@note}{%
7716     \prepare@edindex@for@note{\the\page@num|\the\line@num|\the\subline@num|\
7717     the\page@num|\the\line@num|\the\subline@num|}%
7718     \def\sidenotecontent@{}%
7719     \numgdef{\itemcount@}{0}%
7720     \def\do##1{%
7721         \ifnumequal{\itemcount@}{0}%
7722         {%
7723             \appto\sidenotecontent@{##1}}% Not print not separator before
7724             the 1st note
7725             \appto\sidenotecontent@{\@sidenotesep ##1}%
7726         }%
7727         \numgdef{\itemcount@}{\itemcount@+\@ne}%
7728     }%
7729     \dolistloop{\l@dcnotesetext}%
7730     \ifnumgreater{\itemcount@}{1}{\led@err@ManySidenotes}{}%
7731 %

```

And we do the same for left and right notes (not movable).

```

7730 \gdef\@templ@d{%
7731 \gdef\@templ@n{\l@dcnotesetext\l@dcnotesetext@l\l@dcnotesetext@r}%
7732 \ifx\@templ@d\@templ@n \else%
7733     \if@twocolumn%
7734         \if@firstcolumn%
7735             \setl@dlp@rbox{##1}{\sidenotecontent@}%
7736         \else%
7737             \setl@drp@rbox{\sidenotecontent@}%
7738         \fi%
7739     \else%

```

```

7740 \l@dttempcntb=\sidenote@margin%
7741 \ifnum\l@dttempcntb>\@ne%
7742 \advance\l@dttempcntb by\page@num%
7743 \fi%
7744 \ifodd\l@dttempcntb%
7745 \setl@drp@rbox{\sidenotecontent@}%
7746 \gdef\sidenotecontent@{}%
7747 \numgdef\itemcount@{0}%
7748 \dolistloop{\l@dcsnotetext@}%
7749 \ifnumgreater\itemcount@{1}{\led@err@ManyLeftnotes}{}%
7750 \setl@dlp@rbox{\sidenotecontent@}%
7751 \else%
7752 \setl@dlp@rbox{\sidenotecontent@}%
7753 \gdef\sidenotecontent@{}%
7754 \numgdef\itemcount@{0}%
7755 \dolistloop{\l@dcsnotetext@r}%
7756 \ifnumgreater\itemcount@{1}{\led@err@ManyRightnotes}{}%
7757 \setl@drp@rbox{\sidenotecontent@}%
7758 \fi%
7759 \fi%
7760 \fi%
7761 \advance\@edindex@fornote@m@ne%
7762 }
7763 %

```

## XXVII 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@dfeetbeginmini` These will be the hooks in `\@iiminipage` and `\endminipage`.  
`\l@dfeetendmini` They can be extended to handle other things if necessary.

```

7764 \ifnoledgroup@else%
7765 \newcommand*{\l@dfeetbeginmini}{\@ledgrouptrue\l@dedbeginmini\l@dfambeginmini}
7766 \newcommand*{\l@dfeetendmini}{%
7767 \IfStrEq{critical-familiar}{\@mpfnpos}%
7768 {\l@dedendmini\l@dfamendmini}%
7769 {%
7770 \IfStrEq{familiar-critical}{\@mpfnpos}%
7771 {\l@dfamendmini\l@dedendmini}%
7772 {\do@feet@custom@order{mp@}{\@mpfnpos}}%
7773 }%
7774 }%
7775 %

```

`\l@dedbeginmini` These handle the initiation and closure of critical footnotes in a minipage environment.

`\l@dedendmini`

`\mp@append@Xnotes`

```

7776 \newcommand*{\l@dedbeginmini}{%
7777   \unless\ifnocritical@%
7778     \def\do##1{%
7779       \csletcs{v##1footnote}{mpv##1footnote}%
7780     }%
7781     \dolistloop{\@series}%
7782   \fi%
7783 }
7784 \newcommand*{\l@dedendmini}{%
7785   \unless\ifnocritical@%
7786     \ifl@dpairing%
7787       \ifledRcol%
7788         \flush@notesR%
7789       \else%
7790         \flush@notes%
7791       \fi%
7792     \fi
7793     \def\do##1{%
7794       \mp@append@Xnotes{##1}%
7795     }%
7796     \dolistloop{\@series}%
7797   \fi%
7798 }%
7799 \newcommand{\mp@append@Xnotes}[1]{%
7800   \ifvoid\csuse{mp#1footins}\else%
7801     \ifl@dpairing%
7802       \ifparledgroup%
7803         \ifledRcol%
7804           \dingdef{\parledgroup@beforenotesR}{\parledgroup@beforenotesR+\skip
\@nameuse{mp#1footins}}%
7805         \else%
7806           \dingdef{\parledgroup@beforenotesL}{\parledgroup@beforenotesL+\skip
\@nameuse{mp#1footins}}%
7807         \fi%
7808       \fi%
7809     \fi%
7810     \ifcsstring{series@display#1}{paragraph}{\}%
7811     \setbox\@nameuse{mp#1footins}=\vbox{%
7812       \csuse{Xnotefontsize@#1}%
7813       \ifcsdef{Xhsize}\csuse{series@display#1}@#1{%
7814         \hsize\csuse{Xhsize}\csuse{series@display#1}@#1}%
7815       }{\}%
7816       \noindent\csuse{Xtxtbeforenotes@#1}%
7817       \unvbox\@nameuse{mp#1footins}%
7818       \@parboxrestore%
7819     }%
7820   }%
7821   \csuse{mp#1footgroup}{#1}%

```

```

7822 \fi%
7823 }%
7824 %

```

`\l@dfambeginmini` These handle the initiation and closure of familiar footnotes in a minipage environment.

```

\l@dfamendmini
\mp@append@notesX
7825 \newcommand*{\l@dfambeginmini}{%
7826 \unless\ifnofamiliar{%
7827 \def\do##1{\csletcs{vfootnote##1}{mpvfootnote##1}}%
7828 \dolistloop{\@series}%
7829 \fi%
7830 }%
7831
7832 \newcommand*{\l@dfamendmini}{%
7833 \unless\ifnofamiliar{%
7834 \def\do##1{%
7835 \mp@append@notesX{##1}%
7836 }%
7837 \dolistloop{\@series}%
7838 \fi%
7839 }%
7840 \newcommand{\mp@append@notesX}[1]{%
7841 \ifvoid\csuse{mpfootins#1}\else%
7842 \ifcsstring{series@displayX#1}{paragraph}{%}%
7843 \setbox\@nameuse{mpfootins#1}=\vbox{%
7844 \csuse{notefontsizeX#1}%
7845 \ifcsdef{hsize\csuse{series@display#1}X#1}{%}%
7846 \hsize\csuse{hsize\csuse{series@display#1}X#1}%
7847 }{}%
7848 \noindent\csuse{txtbeforenotesX#1}%
7849 \unvbox\@nameuse{mpfootins#1}%
7850 \@parboxrestore%
7851 }%
7852 }%
7853 \csuse{mpfootgroup#1}{#1}%
7854 \fi%
7855 }%
7856 %

```

`\@iiiminipage` This is our extended form of the kernel `\@iiiminipage` defined in `ltboxes.dtx`.

```

7857 \patchcmd%
7858 {\@iiiminipage}%
7859 {\let\@footnotetext\@mpfootnotetext}%
7860 {\let\@footnotetext\@mpfootnotetext\l@dfeetbeginmini}%
7861 {}%
7862 {\led@error@fail@patch@%iiiminipage}%
7863 %

```

`\endminipage` This is our extended form of the kernel `\endminipage` defined in `ltboxes.dtx`.

```

7864 \patchcmd%
7865   {\endminipage}%
7866   {\footnoterule}%
7867   {\footnoterule\l@advance@parledgroup@beforenormalnotes}%
7868   {}%
7869   {\led@error@fail@patch@endminipage}
7870
7871 \patchcmd%
7872   {\endminipage}%
7873   {\@minipagefalse}%
7874   {\l@dfeetendmini\@minipagefalse}%
7875   {}%
7876   {\led@error@fail@patch@endminipage}
7877
7878 %

```

`\l@dunboxmpfoot` `\@ldunboxmpfoot` insert normal footnotes for ledgroup.  
`edgroup@beforenormalnotes`

```

7879 \newcommand*{\l@dunboxmpfoot}{%
7880   \vskip\skip\@mpfootins
7881   \normalcolor
7882   \footnoterule
7883   \l@advance@parledgroup@beforenormalnotes
7884   \unvbox\@mpfootins%
7885 }
7886 %

```

When using parallel ledgroup, we need to store the vertical space added before footnote, in order to compensate them between left and right pages.

```

7887 \newcommand{\l@advance@parledgroup@beforenormalnotes}{%
7888   \ifparledgroup
7889     \ifl@dpairing
7890       \ifledRcol
7891         \dimdef{\parledgroup@beforenotesR}{\parledgroup@beforenotesR+
skip\@mpfootins}
7892       \else
7893         \dimdef{\parledgroup@beforenotesL}{\parledgroup@beforenotesL+
skip\@mpfootins}
7894       \fi
7895     \fi
7896     \fi
7897 }
7898 %

```

`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.

```

7899

```

```

7900 \newenvironment{ledgroup}{%
7901   \resetprevpage@num%
7902   \def\@mpfn{mpfootnote}\def\thempfn{\thempfootnote}\c@mpfootnote\z@%
7903   \let\@footnotetext\@mpfootnotetext
7904   \l@dfetbeginmini%
7905 }{%
7906   \par
7907   \unskip
7908   \ifvoid\@mpfootins\else
7909     \l@dunboxmpfoot
7910   \fi
7911   \l@dfetendmini%
7912   \@ledgroupfalse%
7913 }
7914
7915
7916 %

```

`\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.

```

7917 \newenvironment{ledgroupsize}[2][1]{%
7918 %

```

Set the various text measures.

```

7919   \hspace #2\relax
7920 %

```

Initialize fills for centering.

```

7921   \let\ledllfill\hfil
7922   \let\ledrlfill\hfil
7923   \def\@tempa{#1}\def\@tempb{1}%
7924 %

```

Left adjusted numbered lines

```

7925     \ifx\@tempa\@tempb
7926     \let\ledllfill\relax
7927   \else
7928     \def\@tempb{r}%
7929     \ifx\@tempa\@tempb
7930 %

```

Right adjusted numbered lines

```

7931     \let\ledrlfill\relax
7932   \fi
7933   \fi
7934 %

```

Set up the footnoting.

```

7935 \def\@mpfn{mpfootnote}\def\thempfn{\thempfootnote}\c@mpfootnote\z@
7936 \let\@footnotetext\@mpfootnotetext
7937 \l@dfetbeginmini%
7938 }{f%
7939 \par
7940 \unskip
7941 \ifvoid\@mpfootins\else
7942 \l@dunboxmpfoot
7943 \fi
7944 \l@dfetendmini%
7945 }
7946 %
7947 %

```

Close the \ifnoledgroup@else.

```

7948 \fi%
7949 %

```

**\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.

```

7950 \newif\ifledgroupnotesL@
7951 \newif\ifledgroupnotesR@
7952 %

```

## XXVIII Indexing

Here is some code for indexing using page and line numbers.

### XXVIII.1 Looking on package order

First, ensure that imakeidx or indextools is loaded *before* eledmac.

```

7953 \AtBeginDocument{%
7954 \unless\ifl@imakeidx%
7955 \@ifpackageloaded{imakeidx}{\led@error@PackageAfterEledmac{imakeidx}}{}
7956 %
7957 \fi%
7958 \unless\ifl@indextools%
7959 \@ifpackageloaded{indextools}{\led@error@PackageAfterEledmac{indextools}}{}
7960 %
7961 \fi%
7962 \unless\ifl@footmisc%
7963 \@ifpackageloaded{footmisc}{\led@error@PackageAfterEledmac{footmisc}}{}
7964 %
7965 \fi%
7966 %

```

## XXVIII.2 Auxiliary macros for `\edindex`

`\pagelinesep` `\edindexlab` `\c@labidx` In order to get a correct line number we have to use the label/ref mechanism. These macros are for that.

```

7965 \newcommand{\pagelinesep}{-}
7966 \newcommand{\edindexlab}{$&}
7967 \newcounter{labidx}
7968 \setcounter{labidx}{0}
7969
7970 %

```

`\doedindexlabel` This macro sets an `\edlabel`.

```

7971 \newcommand{\doedindexlabel}{%
7972   \stepcounter{labidx}%
7973   \edlabel{\edindexlab\thelabidx}%
7974 }
7975
7976 %

```

`\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`.

```

7977 \newcounter{pageline}%
7978 \renewcommand{\thepageline}{%
7979   \thepage%
7980   \pagelinesep%
7981   \xlineref{\edindexlab\thelabidx}%
7982 }
7983 %

```

`\thestartpageline` `\theendpageline` These macros make up the page/line start/end number when the `\edindex` command is called in critical notes.

```

7984 \newcommand{\thestartpageline}{%
7985   \l@dparsedstartpage%
7986   \pagelinesep%
7987   \l@dparsedstartline%
7988 }
7989 \newcommand{\theendpageline}{%
7990   \l@dparsedendpage%
7991   \pagelinesep%
7992   \l@dparsedendline%
7993 }
7994 %

```



### XXVIII.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.

```
7995 \newcount\@edindex@fornote@
7996 %
```

`\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 |.

```
7997 \newcommand{\prepare@edindex@fornote}[1]{%
7998   \l@dp@rsefootspec#1|}%
7999   \advance\@edindex@fornote@\@ne%
8000 }
8001 %
```

`\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.

```
8002 \newcommand{\get@edindex@ledinnote@command}{%
8003   \ifxindy@%
8004     \gdef\@ledinnote@command{%
8005       ledinnote\thelabidx%
8006     }%
8007     \ifxindyhyperref@%
8008       \immediate\write\eledmac@xindy@out{%
8009         (define-attributes ("ledinnote\thelabidx"))^^J
8010         \space\space(markup-locref^^J
8011         \eledmacmarkuplocrefdepth^^J
8012         :open "\string\ledinnote[\edindexlab\thelabidx]{\@index@command
8013         }\@currentseries}{\@currentnotetype}{^^J
8014         :close "}"^^J
8015         :attr "ledinnote\thelabidx"^^J
8016       )
8017     }%
8018   \else%
8019     \immediate\write\eledmac@xindy@out{%
8020       (define-attributes ("ledinnote\thelabidx"))^^J
8021       \space\space(markup-locref^^J
8022       \eledmacmarkuplocrefdepth^^J
8023       :open "\string\ledinnote{\@index@command}{\@currentseries}{\
8024       \@currentnotetype}{^^J
8025       :close "}"^^J
8026       :attr "ledinnote\thelabidx"^^J
8027     }
```

```

8025     )
8026     }%
8027     \fi%
8028 %

```

If we do not use xindy option, `\@ledinnote@command` will produce something like `ledinnote{formattingcommand}`.

```

8029     \else%
8030     \gdef\@ledinnote@command{%
8031         ledinnote[\edindexlab\thelabidx]{\@index@command}{\@currentseries}{\@currentnotetype}%
8032     }%
8033     \fi%
8034 }
8035 %

```

## XXVIII.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 |.

```

8036 \def\get@index@command#1|#2+{%
8037     \gdef\@index@txt{#1}%
8038     \gdef\@index@command{#2}%
8039     \xdef\@index@parenthesis{}%
8040     \IfBeginWith{\@index@command}{({}%
8041         \StrGobbleLeft{\@index@command}{1}[\@index@command@]%
8042         \global\let\@index@command\@index@command@%
8043         \xdef\@index@parenthesis{({}%
8044     }{}%
8045     \IfBeginWith{\@index@command}{})}%
8046         \StrGobbleLeft{\@index@command}{1}[\@index@command@]%
8047         \global\let\@index@command\@index@command@%
8048         \xdef\@index@parenthesis{)}%
8049     }{}%
8050 }
8051 %

```

## XXVIII.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 note series;
- #4: the type of note (critical/familiar).

- #5: the number itself.

```

8052 \newcommandx{\ledinnote}[5][1,usedefault]{%
8053   \IfStrEq{#4}{critical}{%
8054     \def\tmp{Xinnotemark}%
8055   }{%
8056     \def\tmp{innotemarkX}%
8057   }%
8058   \ifboolexpr{%
8059     test{\ifdefequal{\iftrue}{\ifHy@hyperindex}}%
8060     or%
8061     bool {xindyhyperref@}%
8062   }%
8063   {%
8064     \csuse{#2}{\hyperlink{#1}{\csuse{\tmp @#3}{#5}}}%
8065   }%
8066   {%
8067     \csuse{#2}{\csuse{\tmp @#3}{#5}}%
8068   }%
8069 }%
8070 \newcommand{\ledinnotemark}[1]{#1\emph{n}}%
8071 %

```

## XXVIII.6 Main code

Eledmac and ledmac were using the specific indexing tools of the memoir in order to allow multiple index. However, eledmac used imakeidx or indextools tools when one these two package was loaded. This system forced to maintained a double code, which was not very useful. Since reledmac, we use only the imakeidx or indextools tools.

The memoir class provides more flexible indexing than the standard classes. We need different code if the memoir class is being used, except if imakeidx or indextools is used.

`\edindex` Write the index information to the idx file.

```

\@wredindex
\dummy@edindex
8072 \newcommandx{\@wredindex}[2][1=\expandonce\jobname,usedefault]{%#1 = the
index name, #2 = the text
8073   \begingroup%
8074     \let\emph\@firstofone%
8075     \let\textbf\@firstofone%
8076     \let\textit\@firstofone%
8077     \let\textmd\@firstofone%
8078     \let\textnormal\@firstofone%
8079     \let\textrm\@firstofone%
8080     \let\textsc\@firstofone%
8081     \let\textsf\@firstofone%
8082     \let\textsl\@firstofone%
8083     \let\texttt\@firstofone%

```

```

8084 \let\textup\@firstofone%
8085 \xdef\@tmp{#2}%To be used in IfSubStr instead of #2 directly. Avoid
some expansion bugs (for example with \edindex{textsc{something}})
8086 \endgroup%
8087 \ifl@imakeidx%
8088 \ifnum\@edindex@fornote@>\z@%
8089 \IfSubStr[1]{\@tmp}{|}{\get@index@command#2+}{\get@index@command#2|+}
%
8090 \get@edindex@ledinnote@command%
8091 \expandafter\imki@wrindexentry{#1}{\@index@txt|(\@ledinnote@command
}{\thestartpageline}%
8092 \expandafter\imki@wrindexentry{#1}{\@index@txt|)\@ledinnote@command
}{\theendpageline}%
8093 \else%
8094 \get@edindex@hyperref{#2}%
8095 \imki@wrindexentry{#1}{\@index@txt\@edindex@hyperref}{\thepageline}%
8096 \fi%
8097 \else%
8098 \ifnum\@edindex@fornote@>\z@%
8099 \IfSubStr[1]{\@tmp}{|}{\get@index@command#2+}{\get@index@command#2|+}
%
8100 \get@edindex@ledinnote@command%
8101 \expandafter\protected@write\@indexfile{}%
8102 {\string\indexentry{\@index@txt|(\@ledinnote@command}{\thestartpageline}
}%
8103 \expandafter\protected@write\@indexfile{}%
8104 {\string\indexentry{\@index@txt|)\@ledinnote@command}{\theendpageline}
}%
8105 \else%
8106 \protected@write\@indexfile{}%
8107 {\string\indexentry{#2}{\thepageline}
}%
8108 \fi%
8109 \endgroup
8110 \esphack%
8111 }
8112 %

```

Need to add the definition of `\edindex` to `\makeindex`, and initialise `\edindex` to do nothing.

```

8117 \pretocmd{\makeindex}{%
8118 \def\edindex{%
8119 \ifboolexpr{bool{numbering} or bool{numberingR} or bool{
1@dprintingpages} or bool{1@dprintingcolumns}}{%
8120 \bsphack%
8121 \doedindexlabel%
8122 \begingroup%
8123 \@sanitize%
8124 \@wredindex%

```

```

8125 }%
8126 {%
8127   \led@warn@edinde@outsidenumbering%
8128   \index%
8129 }%
8130 }%
8131 }%
8132 {}%
8133 {\led@error@fail@patch@makeindex}%
8134 \newcommand{\edindex}[1]{\@bsphack\@esphack}
8135 \newcommandx{\dummy@edindex}[2][1=\expandonce\jobname,usedefault]{}%
8136 \newcommand{\disable@edindex}{\let\old@edindex\edindex\let\edindex\
dummy@edindex}%
8137 \newcommand{\restore@edindex}{\let\edindex\old@edindex}%
8138 %

```

## XXVIII.7 Hyperlink

**\hyperlinkformat** \hyperlinkformat command is to be used to have both a internal hyperlink and a format, when indexing.

```

8139 \newcommand{\hyperlinkformat}[3]{%
8140   \ifstrempy{#1}%
8141     {\hyperlink{#2}{#3}}%
8142     {\csuse{#1}{\hyperlink{#2}{#3}}}%
8143   }%
8144 %

```

**\hyperlinkR** \hyperlinkR command is to be used to create a internal hyperlink and \ledRflag, when indexing.

```

8145 \newcommand{\hyperlinkR}[2]{%
8146   \hyperlink{#1}{#2\@Rlineflag}%
8147 }%
8148 %
8149 %

```

**\hyperlinkformatR** \hyperlinkformatR command is to be used to create a internal hyperlink, a format and a \@Rlineflag, when indexing.

```

8150 \newcommand{\hyperlinkformatR}[3]{%
8151   \hyperlinkformat{#1}{#2}{#3\@Rlineflag}%
8152 }%
8153 %
8154 %

```

**\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).

```

8155 \newcommand{\get@edindex@hyperref}[1]{%
8156 %

```

We have to disable temporary spaces to work through a xstring bug (or feature?)

```

8157 \edef\temp@{%
8158 \catcode`\ =9 %space need for catcode
8159 \detokenize{#1}%For active character in unicode
8160 \catcode`\ =10 % space need for catcode
8161 }%
8162 %

```

Now, we define \@edindex@hyperref if the hyperindex of hyperref is enabled.

```

8163 \ifdefequal{\iftrue}{\ifHy@hyperindex}{%
8164 \IfSubStr{\temp@}{|}%
8165 {\get@index@command#1+%
8166 \ifledRcol%
8167 \gdef\@edindex@hyperref{|\@index@parenthesis %space kept
8168 hyperlinkformatR{\@index@command}%
8169 {\edindexlab\thelabidx}}%
8170 \else%
8171 \gdef\@edindex@hyperref{|\@index@parenthesis %space kept
8172 hyperlinkformat{\@index@command}%
8173 {\edindexlab\thelabidx}}%
8174 \fi%
8175 }%
8176 {\get@index@command#1|+%
8177 \ifledRcol%
8178 \gdef\@edindex@hyperref{|\hyperlinkR{\edindexlab\thelabidx}}%
8179 \else%
8180 \gdef\@edindex@hyperref{|\hyperlink{\edindexlab\thelabidx}}%
8181 \fi%
8182 }%
8183 }%
8184 %

```

```

8185 % If we use both xindy and hyperref, first get the \protect\cs{
index@command} command.
8186 % Then define \protect\cs{@edindex@hyperref} in the form \verb+eledmacXXX+
8187 % \begin{macrocode}
8188 {\ifxindyhyperref%
8189 \IfSubStr{\temp@}{|}%
8190 {\get@index@command#1+%
8191 {\get@index@command#1|+%
8192 \gdef\@edindex@hyperref{|\eledmac\thelabidx}%
8193 %

```

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.

```

8194 \IfStrEq{\@index@parenthesis}{(}{%

```

```

8195     {%
8196     \csxdef{xindyparenthesis@\@index@txt}{\thelabidx}%
8197     \gdef\@edindex@hyperref{|\(eledmac\thelabidx}%
8198     }%
8199     {}%
8200 %

```

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`.

```

8201     \IfStrEq{\@index@parenthesis}{})}%
8202     {%
8203     \xdef\@edindex@hyperref{|\)eledmac\csuse{xindyparenthesis@\@index@txt}}%
8204     \global\csundef{xindyparenthesis@\@index@txt}%
8205     }%
8206 %

```

Write in the `.xdy` file the attributes of the location.

```

8207     {%
8208     \immediate\write\eledmac@xindy@out{%
8209     (define-attributes ("eledmac\thelabidx"))^^J
8210     \space\space(markup-locoref^^J
8211     \eledmacmarkuplocorefdepth^^J
8212     :open "\string\hyperlink%
8213     \ifledRcol R\fi%
8214     {\edindexlab\thelabidx}%
8215     {\ifdefempty{\@index@command}%
8216     {}%
8217     {\@backslashchar\@index@command}%
8218     {"^^J
8219     :close "})"^^J
8220     :attr "eledmac\thelabidx"^^J
8221     )
8222     }%
8223     }%
8224 %

```

And now, in any other case.

```

8225     \else%
8226     \gdef\@index@txt{#1}%
8227     \gdef\@edindex@hyperref{}%
8228     \fi%
8229     }%
8230 }
8231 %

```

## XXVIII.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.

```

8232 \newcommand{\led@set@index@fornote}[1]{%
8233   \ifbool{indtl@innote}%
8234     {\let\index\nindex}%
8235     {}%
8236   \ifbool{indtl@notenumber}%
8237     {%
8238     \renewcommand{\index}[2][\indtl@jobname]{%
8239       \orig@@index[#1]{%
8240         ##2|innotenumber{\this@footnoteX@reading}%
8241       }%
8242     }%
8243   }%
8244   {}%
8245 }%
8246 %

```

`\led@reinit@index@fornote` The `\led@reinit@index@fornote` just reset the default value of `\index`.

```

8247 \newcommand{\led@reinit@index@fornote}{%
8248   \ifbool{indtl@innote}%
8249     {\let\index\orig@@index}%
8250     {}%
8251   \ifbool{indtl@notenumber}%
8252     {\let\index\orig@@index}%
8253     {}%
8254 }%
8255 %

```

## XXIX Glossaries

Here, we define the `\gls`-like commands prefixed by `ed`, only if the package `glossaries` is loaded.

```

8256 \AtBeginDocument{%
8257   \@ifpackageloaded{glossaries}{%
8258     %

```

First those which arguments are `[<options>]{<label>}[<insert>]`.

```

8259   \gdef\edglsomo{}%
8260   \listgadd{\edglsomo}{gls}%
8261   \listgadd{\edglsomo}{Gls}%
8262   \listgadd{\edglsomo}{GLS}%
8263   \listgadd{\edglsomo}{glspl}%

```



```

8264 \listgadd{\edglsomo}{Glspl}%
8265 \listgadd{\edglsomo}{GLSpl}%
8266 \listgadd{\edglsomo}{glstext}%
8267 \listgadd{\edglsomo}{Glstext}%
8268 \listgadd{\edglsomo}{GLStext}%
8269 \listgadd{\edglsomo}{Glsfirst}%
8270 \listgadd{\edglsomo}{GLSfirst}%
8271 \listgadd{\edglsomo}{glsplural}%
8272 \listgadd{\edglsomo}{Glsplural}%
8273 \listgadd{\edglsomo}{GLSplural}%
8274 \listgadd{\edglsomo}{glsfirstplural}%
8275 \listgadd{\edglsomo}{GLSfirstplural}%
8276 \listgadd{\edglsomo}{GLSfirstplural}%
8277 \listgadd{\edglsomo}{glsname}%
8278 \listgadd{\edglsomo}{Glsname}%
8279 \listgadd{\edglsomo}{GLSname}%
8280 \listgadd{\edglsomo}{glssymbol}%
8281 \listgadd{\edglsomo}{Glssymbol}%
8282 \listgadd{\edglsomo}{GLSsymbol}%
8283 \listgadd{\edglsomo}{glsdesc}%
8284 \listgadd{\edglsomo}{Glsdesc}%
8285 \listgadd{\edglsomo}{GLSdesc}%
8286 \listgadd{\edglsomo}{glsuseri}%
8287 \listgadd{\edglsomo}{Glsuseri}%
8288 \listgadd{\edglsomo}{GLSuseri}%
8289 \listgadd{\edglsomo}{glsuserii}%
8290 \listgadd{\edglsomo}{Glsuserii}%
8291 \listgadd{\edglsomo}{GLSuserii}%
8292 \listgadd{\edglsomo}{glsuseriii}%
8293 \listgadd{\edglsomo}{Glsuseriii}%
8294 \listgadd{\edglsomo}{GLSuseriii}%
8295 \listgadd{\edglsomo}{glsuseriv}%
8296 \listgadd{\edglsomo}{Glsuseriv}%
8297 \listgadd{\edglsomo}{GLSuseriv}%
8298 \listgadd{\edglsomo}{glsuserv}%
8299 \listgadd{\edglsomo}{Glsuserv}%
8300 \listgadd{\edglsomo}{GLSuserv}%
8301 \listgadd{\edglsomo}{glsuservi}%
8302 \listgadd{\edglsomo}{Glsuservi}%
8303 \listgadd{\edglsomo}{GLSuservi}%
8304 \renewcommand{\do}[1]{%
8305 \expandafter\DeclareRobustCommandx\csname ed#1\endcsname[3][1,3,
usedefault]{%
8306 \doedindexlabel%
8307 \ifdef{\hypertarget}{%
8308 \Hy@raisedlink@left{\hypertarget{pageline.\thepageline}}}%
8309 }{}%
8310 \csname#1\endcsname[counter=pageline,##1]{##2}[##3]%
8311 }%
8312 \expandafter\WithSuffix\expandafter\DeclareRobustCommandx\csname ed

```

```

#1\endcsname*[3][1,3,usedefault]{%
8313   \doedindexlabel%
8314   \ifdef{\hypertarget}{%
8315     \Hy@raisedlink@left{\hypertarget{pageline.\thepageline}}}%
8316   }{}%
8317   \csname#1\endcsname*[counter=pageline,##1]{##2}{##3}%
8318   }%
8319 }%
8320 \dolistloop{\edglsomo}%
8321 %

```

Then those which arguments are [*options*]{*label*}{*link text*}.

```

8322 \gdef\edglsomm{%
8323   \listgadd\edglsomm{glsdisp}%
8324   \listgadd\edglsomm{glslink}%
8325   \renewcommand{\do}[1]{%
8326     \expandafter\DeclareRobustCommandx\csname ed#1\endcsname[3][1,
usedefault]{%
8327     \doedindexlabel%
8328     \ifdef{\hypertarget}{%
8329       \Hy@raisedlink@left{\hypertarget{pageline.\thepageline}}}%
8330     }{}%
8331     \csname#1\endcsname[counter=pageline,##1]{##2}{##3}%
8332     }%
8333     \expandafter\WithSuffix\expandafter\DeclareRobustCommandx\csname ed
#1\endcsname*[3][1,usedefault]{%
8334     \doedindexlabel%
8335     \ifdef{\hypertarget}{%
8336       \Hy@raisedlink@left{\hypertarget{pageline.\thepageline}}}%
8337     }{}%
8338     \csname#1\endcsname*[counter=pageline,##1]{##2}{##3}%
8339     }%
8340   }%
8341   \dolistloop{\edglsomm}%
8342 %

```

Then those which arguments are [*options*]{*label*}.

```

8343 \gdef\edglsom{%
8344   \listgadd\edglsom{glsadd}%
8345   \renewcommand{\do}[1]{%
8346     \expandafter\DeclareRobustCommandx\csname ed#1\endcsname[2][1,
usedefault]{%
8347     \doedindexlabel%
8348     \ifdef{\hypertarget}{%
8349       \Hy@raisedlink@left{\hypertarget{pageline.\thepageline}}}%
8350     }{}%
8351     \csname#1\endcsname[counter=pageline,##1]{##2}%
8352     }%
8353     \expandafter\WithSuffix\expandafter\DeclareRobustCommandx\csname ed
#1\endcsname*[2][1,usedefault]{%

```

```

8354 \doedindexlabel%
8355 \ifdef{\hypertarget}{%
8356 \Hy@raisedlink@left{\hypertarget{pageline.\thepageline}{}}%
8357 }{}%
8358 \csname#1\endcsname*[counter=pageline,##1]{##2}%
8359 }%
8360 }%
8361 \dolistloop{\edglsom}%
8362 %

```

`\l@noexpands@edgl` is a macro to avoid expanding glossaries macros in `\edtext` first argument. It is called by `\no@expands`. Its definition is made dynamically.

```

\l@noexpands@edgl \def\l@noexpands@edgl{%
8363 \renewcommand\do[1]{%
8364 \xappto\l@noexpands@edgl{%
8365 \noexpand\RenewDocumentCommand\csname ed#1\endcsname{omos}{}%
8366 }%
8367 }%
8368 \dolistloop{\edglsomo}%
8369 \renewcommand\do[1]{%
8370 \xappto\l@noexpands@edgl{%
8371 \noexpand\RenewDocumentCommand\csname ed#1\endcsname{omms}{}%
8372 }%
8373 }%
8374 \dolistloop{\edglsomm}%
8375 \renewcommand\do[1]{%
8376 \xappto\l@noexpands@edgl{%
8377 \noexpand\RenewDocumentCommand\csname ed#1\endcsname{oms}{}%
8378 }%
8379 }%
8380 \dolistloop{\edglsom}%
8381 }{\let\l@noexpands@edgl\relax}%
8382 }%
8383 %
8384 %

```

## XXX 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`.

### XXX.1 Hanging symbol management

`\@hangingsymbol` The macro `\@hangingsymbol` is used to insert a symbol on each hanging of verses. It is set by user level macro `\sethangingsymbol`.  
`\sethangingsymbol`  
`\ifinstanza`

For example, in French typography, the symbol is ‘[’. We obtain it by means of the following code:

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

The `\ifinstanza` boolean is used to ensure that we are in a stanza part.

```
8385 \def\@hangingsymbol{}
8386 \newcommand*{\sethangingsymbol}[1]{%
8387   \gdef\@hangingsymbol{#1}%
8388 }%
8389 \newif\ifinstanza
8390 %
```

`\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.

```
8391 \newif\ifinserthangingsymbol
8392 \newcommand{\inserthangingsymbol}{%
8393   \ifinserthangingsymbol%
8394     \ifinstanza%
8395       \@hangingsymbol%
8396     \fi%
8397 \fi%
8398 }
8399 %
```

## XXX.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.

```
8400 \newcommand*{\ampersand}{\char`\&}
8401
8402 %
```

## XXX.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.

```
8403 \chardef\body=\catcode`\@
8404 \catcode`\@=11
8405 \chardef\next=\catcode`\&
8406 \catcode`\&=\active
8407
8408 %
```

### XXX.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.

```
8409 \newcount\stanza@count
8410 \newlength{\stanzaindentbase}
8411 \setlength{\stanzaindentbase}{20pt}
8412
8413 %
```

`\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.

```
8414 \def\strip@szacnt#1,#2|{\def\@tempb{#1}\def\@tempa{#2|}}
8415 \newcommand*\setstanzavalues[2]{\def\@tempa{#2,,|}%
8416   \stanza@count\z@
8417   \def\next{\expandafter\strip@szacnt\@tempa
8418     \ifx\@tempb\empty\let\next\relax\else
8419       \expandafter\mathchardef\csname #1\@number\stanza@count
8420       \@endcsname\@tempb\relax
8421       \advance\stanza@count\@ne\fi\next}%
8422   \next}
8423
8424 %
```

`\setstanzaindents` In the original edmac, `\setstanzavalues{sza}{⟨...⟩}` had to be called to set the in-  
`\setstanzapenalties` dents, 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).

```
8425 \newcommand*\setstanzaindents[1]{\setstanzavalues{sza}{#1}}
8426 \newcommand*\setstanzapenalties[1]{\setstanzavalues{szp}{#1}}
8427 %
8428 %
```

`\managestanza@modulo` Since the `stanzaindentrepetition` counter can be used when the indentation is repeated every  $n$  verses.

The `startstanzaindentrepetition` counter tells in which line of verse the repetition of indent starts.

The `\managestanza@modulo` macro modifies the counter `stanza@modulo`. The command adds 1 to `stanza@modulo`, but if `stanza@modulo` is equal to the `stanzaindentrepetition` counter, the command restarts it.

```

8429 \newcounter{stanzaindentrepetition}
8430 \newcounter{startstanzaindentrepetition}%
8431 \setcounter{startstanzaindentrepetition}{1}%
8432 \newcount\stanza@modulo
8433
8434 \newcommand*{\managestanza@modulo}[0]{%
8435     \advance\stanza@modulo\@ne%
8436     \ifnum\stanza@modulo>\value{stanzaindentrepetition}%
8437         \stanza@modulo\@ne%
8438     \fi%
8439 }
8440 %

```

**\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.

```

8441 \newcommand{\stanzaindent}[1]{%
8442     \hspace{\dimexpr#1\stanzaindentbase-\parindent\relax}%
8443     \ignorespaces%
8444 }%
8445 \WithSuffix\newcommand\stanzaindent*[1]{%
8446     \stanzaindent{#1}%
8447     \global\advance\stanza@modulo-\@ne%
8448     \ifnum\stanza@modulo=0%
8449         \global\stanza@modulo=\value{stanzaindentrepetition}%
8450     \fi%
8451     \ignorespaces%
8452 }%
8453 %

```

### XXX.5 Numbering stanza

Here, macro for numbering stanza. First, the stanza counter.

```

\thestanza54 \newcounter{stanza}
8455 \renewcommand{\thestanza}{%
8456     \textbf{\arabic{stanza}}%
8457 }
8458 %

```

**\ifnumberstanza** Then, macro to activate automatically numbering of stanza.

```

8459 \newif\ifnumberstanza%
8460 %

```

**\@insertstanzanumber** Now, macro called at the first line of of verse of a stanza.

```

8461 \newcommand{\@insertstanzanumber}[0]{%
8462   \ifnumberstanza%
8463     \ifl@dpairing%
8464       \ifledRcol%
8465         \stanzanumwrapper{\thestanzaR}%
8466       \else%
8467         \stanzanumwrapper{\thestanzaL}%
8468       \fi%
8469     \else%
8470       \stanzanumwrapper{\thestanza}%
8471     \fi%
8472     \setline{1}%
8473   \fi%
8474 }%
8475 %

```

**\@advancestanzanumber** Also a command to advance the counter of stanza.

```

8476 \newcommand{\@advancestanzanumber}[0]{%
8477   \ifnumberstanza%
8478     \ifl@dpairing%
8479       \ifledRcol%
8480         \addtocounter{stanzaR}{1}%
8481       \else%
8482         \addtocounter{stanzaL}{1}%
8483       \fi%
8484     \else%
8485       \addtocounter{stanza}{1}%
8486     \fi%
8487   \fi%
8488 }%
8489 %

```

**\stanzanumwrapper** And finally, the wrapper for stanza number

```

8490 \newcommand{\stanzanumwrapper}[1]{%
8491   \flagstanza{#1}%
8492 }%
8493 %

```

## XXX.6 Stanza number in note

Here, the command called when printing stanza number in notes.

```

8494 \newcommand{\printstanza}[0]{%
8495   \ifbool{expr{bool{l@dpairing} or bool{l@dprintingpages} or bool{
8496     l@dprintingcolumns}}}{%
8497     \ifledRcol%
8498       \thestanzaR%

```

```

8498     \else%
8499         \thestanzaL%
8500     \fi%
8501 }{%
8502     \thestanza%
8503 }%
8504 }
8505 %

```

### XXX.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.

`\stanza@indent` 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.

```

8506 \newcommand{\stanza@line}[2][1,2,usedefault]{%
8507     \stanza@indent%
8508     \pstart[#1][#2]\stanza@hang\ignorespaces%
8509 }%
8510 \newcommand{\stanza@indent}{%
8511     \ifbool{expr}%
8512         test {\ifnumequal{\value{stanzaindentrepetition}}{0}}%
8513         or%
8514         test {\ifnumless{\number\stanza@count}{\value{
8515 startstanzaindentrepetition}}}%
8516     }{%
8517         \ifcsdef{sza@\number\stanza@count @}{%
8518             {%
8519                 \parindent=\csname sza@\number\stanza@count @\endcsname\
8520 stanzaindentbase%
8521             }%
8522         }{%
8523             \ifcsdef{sza@\number\stanza@modulo @}{%
8524                 \parindent=\csname sza@%
8525 \the\numexpr%
8526 \stanza@modulo%
8527 - 1%
8528 + \value{startstanzaindentrepetition}%
8529 \relax%
8530 @\endcsname\stanzaindentbase%
8531 \managestanza@modulo%
8532             }%

```



```

8533     {%
8534     \led@err@StanzaIndentNotDefined%
8535     }%
8536 }%
8537 }%
8538 \xdef\stanza@hang{\noexpand\leavevmode\noexpand\startlock
8539     \hangindent\expandafter
8540     \noexpand\csname sza@0@\endcsname\stanzaindentbase
8541     \hangafter\@ne}
8542 \def\sza@penalty{\count@\csname szp@\number\stanza@count @\endcsname
8543     \ifnum\count@>@M\advance\count@-@M\penalty-\else
8544     \penalty\fi\count@}
8545 %

```

\@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  
 \AtEveryStopStanza the enumeration is by verse line rather than by print line. If the print line count is desired,  
 \AtEveryStanza invoke \let\startlock\relax and do the same for \endlock. Here and above we  
 \AtStartEveryStanza have used \xdef to make the stored macros take up a bit less space, but it also makes  
 \BeforeEveryStopStanza them more obscure to the reader. Lines of the stanza are delimited by ampersands &.  
 \newverse The last line of the stanza must end with \&.

```

8546 \xdef\@startstanza[#1][#2]{%
8547     \noexpand\instanzatrue\expandafter
8548     \begingroup%
8549     \catcode`\noexpand\&\active%
8550     \unexpanded{\newdimen\parindent@beforestanza%
8551     \parindent@beforestanza=\parindent}%Keep in memory the standard \
parindent
8552     \global\stanza@count\@ne\stanza@modulo\@ne
8553     \noexpand\ifnum\expandafter\noexpand\csname sza@0@\endcsname=\z@%
8554     \def\noexpand\stanza@hang{\unexpanded{\leavevmode\null}}%
8555     \let\noexpand\endlock\relax%
8556     \noexpand\else%
8557     \interlinepenalty%
8558     \@M\rightskip\z@ plus 1fil\relax%
8559     \noexpand\fi%
8560     \noexpand\ifnum\expandafter\noexpand\csname szp@0@\endcsname=\z@%
8561     \let\noexpand\sza@penalty%
8562     \relax\noexpand\fi%
8563     \def\noexpand&{%
8564         \noexpand\newverse[] []}%
8565     \def\noexpand\&{\noexpand\@stopstanza}%
8566     \noexpand\@advancestanzanumber%
8567     \noexpand\stanza@line[#1][#2]%
8568     \noexpand\@insertstanzanumber%
8569     \let\par\relax\ignorespaces%No paragraph in verses
8570 }

```

```

8571 \newcommandx{\stanza}[2][1,2,usedefault]{%
8572   \ifboolexpr{%
8573     not test{\ifdefvoid{\at@every@stanza}}%
8574     and test{\ifstrempy{#1}}%
8575     and test{\ifstrempy{#2}}}%
8576     {\@startstanza[][\at@every@stanza]\at@start@every@stanza}%
8577     {\@startstanza[#1][#2]\at@start@every@stanza}%
8578   }%
8579 }%
8580
8581 \newcommandx{\@stopstanza}[2][1,2,usedefault]{%
8582   \unskip%
8583   \endlock%
8584   \ifboolexpr{%
8585     not test{\ifdefvoid{\at@every@stop@stanza}}%
8586     and test{\ifstrempy{#1}}%
8587     and test{\ifstrempy{#2}}}%
8588     {\before@every@stop@stanza\pend[][\at@every@stop@stanza]}%
8589     {\before@every@stop@stanza\pend[#1][#2]}%
8590   \endgroup%
8591   \instanzafalse%
8592   \global\undef\parindent@beforestanza%
8593 }
8594
8595 \newcommand{\AtEveryStopStanza}[1]{%
8596   \ifstrempy{#1}%
8597     {\gdef\at@every@stop@stanza{}}%
8598     {\gdef\at@every@stop@stanza{\noindent#1}}%
8599   }%
8600 \WithSuffix\newcommand\AtEveryStopStanza*[1]{%
8601   \ifstrempy{#1}%
8602     {\gdef\at@every@stop@stanza{}}%
8603     {\gdef\at@every@stop@stanza{#1}}%
8604   }%
8605 \def\at@every@stop@stanza{}%
8606
8607 \newcommand{\AtEveryStanza}[1]{%
8608   \ifstrempy{#1}%
8609     {\gdef\at@every@stanza{}}%
8610     {\gdef\at@every@stanza{\noindent#1}}%
8611   }%
8612 \WithSuffix\newcommand\AtEveryStanza*[1]{%
8613   \ifstrempy{#1}%
8614     {\gdef\at@every@stanza{}}%
8615     {\gdef\at@every@stanza{#1}}%
8616   }%
8617
8618
8619
8620 \newcommand{\AtStartEveryStanza}[1]{%

```

```

8621 \ifstrempy{#1}%
8622   {\gdef\at@start@every@stanza{}}}%
8623   {\gdef\at@start@every@stanza{#1}}}%
8624 }%
8625 \def\at@start@every@stanza{}%
8626
8627 \newcommand{\BeforeEveryStopStanza}[1]{%
8628   \ifstrempy{#1}%
8629     {\gdef\before@every@stop@stanza{}}}%
8630     {\gdef\before@every@stop@stanza{#1}}}%
8631 }%
8632 \def\before@every@stop@stanza{}%
8633
8634 \newcommand*{\newverse}[4][1,2,3,4,usedefault]{%
8635   \unskip%
8636   \endlock\pend[#1][#3]\sza@penalty\global%
8637   \advance\stanza@count\@ne\stanza@line[#2][#4]%
8638 }
8639
8640 %

```

**\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`.

```

8641 \newcommand*{\flagstanza}[2][\stanzaindentbase]{%
8642   \hskip -#1\llap{#2}\hskip #1\ignorespaces}
8643
8644 %

```

## XXX.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.

```

8645 \catcode`\&=\next
8646 \catcode`\@=\body
8647 \setstanzavalues{szp}{0}
8648
8649 %

```

## XXX.9 Dealing with antilabe

**\firsthalf@antilabe@wd** If the `antilabe` option is enabled, after having vboxed the content of `\pstart...pend` structure, we get the width of the last line of the `\pend`.  
**get@firsthalf@antilabe@wd**  
**\antilabe**

```

8650 \ifantilabe@%
8651   \newdimen\firsthalf@antilabe@wd%
8652   \newcommand\get@firsthalf@antilabe@wd{%
8653     \ifboolexpr{bool {istanza} or bool {inastanzaL} or bool {inastanzaR}}{
8654       %
8655       \vbadness=10000%
8656       \splittopskip=\z@%
8657       \ifl@dpairing%
8658         \ifledRcol%
8659           \setbox0=\copy\csname l@dRcolrawbox\the\l@dnumpstartsR\endcsname%
8660         \else%
8661           \setbox0=\copy\csname l@dLcolrawbox\the\l@dnumpstartsL\endcsname%
8662         \fi%
8663       \else%
8664         \setbox0=\copy\raw@text%
8665       \fi%
8666       \dimen0=\dimexpr\ht0+\dp0-\baselineskip\relax%
8667       \ifdim\dimen0>0pt%
8668         \setbox1=\vsplit0 to \dimen0%
8669       \fi%
8670       \setbox0=\vbox{\unvbox0\global\setbox1=\lastbox}%
8671       \setbox1=\hbox{\noindent\unhbox1}%
8672       \global\firsthalf@antilabe@wd=\wd1%
8673     }{}%
8674   }%
8675   \def\beforeantilabe{}%
8676   \def\afterantilabe{}%
8677   \newcommand{\antilabe}[1]{%
8678     \beforeantilabe\hbox to\firsthalf@antilabe@wd{#1\hfill}\afterantilabe%
8679   }%
8680 \else%
8681   \let\get@firsthalf@antilabe@wd\relax%
8682 \fi%
8683 %

```

## XXXI Apparatus of Manuscripts

### XXXI.1 User level macro

**\msdata** The user level `\msdata` command only writes the manuscripts data in a numbered auxiliary file. There are two associated etoolbox counters.

```

\msdata@cR
8683 \def\msdata@c{}%
8684 \def\msdata@cR{}%
8685 \newcommand{\msdata}[1]{%
8686   \leavevmode%
8687   \unless\ifstopmsdata@inserted%
8688     \stopmsdata%
8689     \led@warning@msdatawithoutstop%

```

```

8690 \fi%
8691 \global\stopmsdata@inserted@false%
8692 \unless\ifledRcol%
8693   \numgdef{\msdata@c}{\msdata@c+1}%
8694   \ifdef{\hypertarget}{%
8695     \edlabel{\msdata@c:start:msdata}%
8696   }{}%
8697   \protected@write\linenum@out{}{%
8698     \string\@msd{#1}%
8699   }%
8700 \else%
8701   \numgdef{\msdata@cR}{\msdata@cR+1}%
8702   \ifdef{\hypertarget}{%
8703     \edlabel{\msdata@cR:start:msdata}%
8704   }{}%
8705   \protected@write\linenum@outR{}{%
8706     \string\@msd{#1}%
8707   }%
8708 \fi%
8709 }%
8710 %

```

**\stopmsdata** The user level `\stopmsdata` command only writes information about the end of manuscripts data in numbered auxiliary file.

```

8711 \newcommand{\stopmsdata}[0]{%
8712   \leavevmode%
8713   \unless\ifledRcol%
8714     \protected@write\linenum@out{}{%
8715       \string\@stopmsd%
8716     }%
8717     \ifdef{\hypertarget}{%
8718       \edlabel{\msdata@c:end:msdata}%
8719     }{}%
8720   \else%
8721     \protected@write\linenum@outR{}{%
8722       \string\@stopmsd%
8723     }%
8724     \ifdef{\hypertarget}{%
8725       \edlabel{\msdata@cR:end:msdata}%
8726     }{}%
8727   \fi%
8728   \global\stopmsdata@inserted@true%
8729 }%
8730 %

```

**\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`

```

8731 \newif\ifstopmsdata@inserted@%
8732 %

```

### XXXI.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?

```

8733 \newcommand{\setmsdataseries}[1]{%
8734   \gdef\@msdata@series{#1}%
8735 }%
8736 \def\@msdata@series{A}%
8737 %

```

**\setmsdataposition** The label for the manuscripts data.

```

8738 \def\ms@data@position{msdata-regular}%
8739 \newcommand{\setmsdataposition}[1]{%
8740   \gdef\ms@data@position{#1}%
8741 }%
8742 %

```

**\setmsdatalabel** The label for the manuscripts data.

```

8743 \def\ms@data@label{Ms.}%
8744 \newcommand{\setmsdatalabel}[1]{%
8745   \gdef\ms@data@label{#1}%
8746 }%
8747 %

```

### XXXI.3 Counters and lists

**\@msd@c** \@msd@c is a counter incremented at each \@msd read in auxiliary file.

```

8748 \numdef{\@msd@c}{0}
8749 \numdef{\@msd@cR}{0}
8750 %

```

**\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.

```

8751 \numdef{\add@msd@c}{0}%
8752 \numdef{\add@msd@cR}{0}%
8753 %

```

**\@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.

```

8754 \def\@msdata@list{%
8755 %

```

## XXXI.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.

```

8756 \newcommand{\@msd}[1]{%
8757 \unless\ifledRcol%
8758 \global\newdef{\@msd@c}{\@msd@c+\@ne}%
8759 \csgdef{\@msdata@\@msd@c @data}{#1}%
8760 \csxdef{\@msdata@\@msd@c @linenumber}{\the\line@num}%
8761 \csxdef{\@msdata@\@msd@c @sublinenumber}{\the\subline@num}%
8762 \csxdef{\@msdata@\@msd@c @abslinenumber}{\the\absline@num}%
8763 \xright@appenditem{\the\absline@num}\to\actionlines@list%
8764 \xright@appenditem{-1010}\to\actions@list%
8765 \else%
8766 \global\newdef{\@msd@cR}{\@msd@cR+\@ne}%
8767 \csgdef{\@msdata@\@msd@cR @dataR}{#1}%
8768 \csxdef{\@msdata@\@msd@cR @linenumberR}{\the\line@numR}%
8769 \csxdef{\@msdata@\@msd@cR @sublinenumberR}{\the\subline@numR}%
8770 \csxdef{\@msdata@\@msd@cR @abslinenumberR}{\the\absline@numR}%
8771 \xright@appenditem{\the\absline@numR}\to\actionlines@listR%
8772 \xright@appenditem{-1010}\to\actions@listR%
8773 \fi%
8774 }%
8775 %

```

**\@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.

```

8776 \newcommand{\@stopmsd}[0]{%
8777 \unless\ifledRcol%
8778 \ifcsundef{\@msdata@\@msd@c @stoplinenumber}{%
8779 \csxdef{\@msdata@\@msd@c @stopabslinenumber}{\the\absline@num}%
8780 \csxdef{\@msdata@\@msd@c @stoplinenumber}{\the\line@num}%
8781 \csxdef{\@msdata@\@msd@c @stopsublinenumber}{\the\subline@num}%
8782 }}%
8783 \else%
8784 \ifcsundef{\@msdata@\@msd@cR @stoplinenumberR}{%
8785 \csxdef{\@msdata@\@msd@cR @stopabslinenumberR}{\the\absline@numR}%
8786 \csxdef{\@msdata@\@msd@cR @stoplinenumberR}{\the\line@numR}%
8787 \csxdef{\@msdata@\@msd@cR @stopsublinenumberR}{\the\subline@numR}%
8788 }%
8789 }%
8790 \fi%

```

8791 }%  
8792 %

### XXXI.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`.

```
8793 \newcommand{\add@msdata}{%
8794   \bgroup%
8795   \normalfont%
8796   \unless\ifledRcol{%
8797     \numgdef{\add@msd@c}{\add@msd@c+\@ne}%
8798     \ifcsdef{@msdata@\add@msd@c @data}{%
8799       \letcs{@data}{@msdata@\add@msd@c @data}%
8800       \edef\l@d@nums{%
8801         000| % Start page = we don't print it
8802         \csuse{@msdata@\add@msd@c @linenumber}| % Start line number
8803         \csuse{@msdata@\add@msd@c @sublinenumber}| % Start subline number
8804         000| % End page number, we don't print it
8805         \ifnumless{\csuse{@msdata@\add@msd@c @stopabslinenumber}}{\csuse{
@lastabsline@forpage@the\page@num}}%
8806         {\csuse{@msdata@\add@msd@c @stoplinenumber}} % End line number if
in the same page
8807         {\csuse{@lastline@forpage@the\page@num}} % Otherwise, last
number of the page
8808         | %
8809         \ifnumless{\csuse{@msdata@\add@msd@c @stopabslinenumber}}{\csuse{
@lastabsline@forpage@the\page@num}}%
8810         {\csuse{@msdata@\add@msd@c @stopsublinenumber}} % End subline
number if in the same page
8811         {\csuse{@lastsubline@forpage@the\page@num}} % Otherwise, last
subnumber of the page
8812         | %
8813         \edefont@info%Font
8814       }%
8815       \@msd@options@fullpagefalse%
8816       \if@firstlineofpage% Try if the data are for the full page. If yes
, will add options to the list.
8817       \unless\if@msdata@insertedfrompreviouspage%
8818       \ifnumless{\csuse{@lastabsline@forpage@the\page@num}}{\csuse{
@msdata@\add@msd@c @stopabslinenumber}+\@ne}%
8819       {%
8820       \numdef{@tmp}{\add@msd@c+\@ne}%
8821       \ifcsdef{@msdata@\@tmp @abslinenumber}%
8822       {\ifnumequal{\csuse{@msdata@\@tmp @abslinenumber}}{\csuse{
@lastabsline@forpage@the\page@num}}%
```



```

8823         {}%
8824         {\@msd@options@fullpagetrue}%
8825     }%
8826     {\@msd@options@fullpagetrue}%
8827 }%
8828 {}%
8829 \fi%
8830 \fi%
8831 \listxadd{\@msdata@list}{%
8832     \@msd@options@iffullpage%
8833     \ifluatex%
8834         \csxdef{footnote@luatextextdir}{\the\textdir}%
8835         \csxdef{footnote@luatexpardir}{\the\pardir}%
8836     \fi%
8837     \csdef{@this@crossref@start}{\add@msd@c:start:msdata}%
8838     \csdef{@this@crossref@end}{\add@msd@c:end:msdata}%
8839     \unexpanded{%
8840         \def\annot@start{}%
8841         \def\annot@end{}%
8842     }%
8843     \noexpand\csuse{v\@msdata@series footnote}{\@msdata@series}{\{
expandonce\l@d@nums}{\ms@data@label}{\expandonce\data}}%
8844     \reset@msd@options@iffullpage%
8845 }%
8846 }%
8847 {}%
8848 \else%
8849     \numgdef{\add@msd@cR}{\add@msd@cR+\@ne}%
8850     \ifcsdef{@msdata@\add@msd@cR @dataR}{%
8851         \letcs{@data}{\@msdata@\add@msd@cR @dataR}%
8852         \edef\l@d@nums{%
8853             000| % Start page = we don't print it
8854             \csuse{@msdata@\add@msd@cR @linenumberR}| % Start line number
8855             \csuse{@msdata@\add@msd@cR @sublinenumberR}| % Start subline
number, for now, not used
8856             000| % End page number, we don't print it
8857             \ifnumless{\csuse{@msdata@\add@msd@cR @stopabslinenumberR}}{\
csuse{@lastline@forpageR@\the\page@numR}}{%
8858                 {\csuse{@msdata@\add@msd@cR @stoplinenumberR}}}%End line number
if in the same page
8859                 {\csuse{@lastline@forpageR@\the\page@numR}}}%Otherwise, last
number of the page
8860             |%
8861             \ifnumless{\csuse{@msdata@\add@msd@cR @stopabslinenumberR}}{\
csuse{@lastline@forpageR@\the\page@numR}}{%
8862                 {\csuse{@msdata@\add@msd@cR @stopsublinenumberR}}}%End subline
number if in the same page
8863                 {\csuse{@lastsubline@forpageR@\the\page@numR}}}%Otherwise, last
number of the page
8864             000| % End sub line number, for now, not used

```

```

8865 \edfont@info%Font
8866 }%
8867 \msd@options@fullpagefalse%
8868 \if@firstlineofpageR%
8869 \unless\if@msdata@insertedfrompreviouspage%
8870 \ifnumless{\csuse{@lastabsline@forpageR}\the\page@numR}}{\
csuse{@msdata@add@msd@c @stopabslinenumberR}+\@ne}%
8871 {%
8872 \numdef{\@tmp}{\add@msd@cR+\@ne}%
8873 \ifcsdef{@msdata@\@tmp @abslinenumberR}%
8874 {\ifnumequal{\csuse{@msdata@\@tmp @abslinenumberR}}{\csuse{
@lastabsline@forpageR}\the\page@numR}}}%
8875 }%
8876 {\msd@options@fullpagetrue}%
8877 }%
8878 {\msd@options@fullpagetrue}%
8879 }%
8880 {}%
8881 \fi%
8882 \fi%
8883 \listxadd{\msdata@list}{%
8884 \msd@options@iffullpage%
8885 \ifluatex%
8886 \csxdef{footnote@luatextextdir}{\the\textdir}%
8887 \csxdef{footnote@luatexpardir}{\the\pardir}%
8888 \fi%
8889 \csdef{@this@crossref@start}{\add@msd@cR:start:msdata}%
8890 \csdef{@this@crossref@end}{\add@msd@cR:end:msdata}%
8891 \unexpanded{%
8892 \def\annot@start{}}%
8893 \def\annot@end{}}%
8894 }%
8895 \noexpand\csuse{v\msdata@series footnote}{\msdata@series}{\
expandonce\ld@nums}{\msdata@label}{\expandonce\data}}%
8896 \reset@msd@options@iffullpage%
8897 }%
8898 }%
8899 {}%
8900 \fi%
8901 \egroup%
8902 }%
8903 %

```

`\if@msdata@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.

```

8904 \newif\if@msdata@insertedfrompreviouspage%
8905 %

```

`\add@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.

```

8906 \newcommand{\add@msdata@firstlineofpage}{%
8907   \bgroup%
8908   \normalfont%
8909   \unless\ifledRcol{%
8910     \ifcsdef{@msdata@\add@msd@c @data}{%
8911       \ifnumless{\the\absline@num-\@one}{\csuse{@msdata@\add@msd@c
@stopabslinenumber}}}%
8912       {%
8913         \global\@msdata@insertedfrompreviouspagetrue%
8914         \letcs{\@data}{@msdata@\add@msd@c @data}%
8915         \edef\l@d@nums{%
8916           000}% Start page = we don't print it
8917           \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
8918           000}% Start subline number, for now, not used
8919           000}% End page number, we don't print it
8920           \ifnumless{\csuse{@msdata@\add@msd@c @stopabslinenumber}}{\csuse{
@lastabsline@forpage@\the\page@num}}}%
8921           {\csuse{@msdata@\add@msd@c @stoplinenumber}}}%End line number if
in the same page
8922           {\csuse{@lastline@forpage@\the\page@num}}}%Otherwise, last
number of the page
8923           |%
8924           000}% End sub line number, for now, not used
8925           \edef\info@font%Font
8926           }%
8927           \@msd@options@fullpagefalse%
8928           \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
8929           {%
8930             \numdef{\@tmp}{\add@msd@c+\@one}%
8931             \ifcsdef{@msdata@\@tmp @abslinenumber}%
8932             {\ifnumequal{\csuse{@msdata@\@tmp @abslinenumber}}{\csuse{
@lastabsline@forpage@\the\page@num}}}%
8933             {}%
8934             {\@msd@options@fullpagetrue}%
8935             }%
8936             {\@msd@options@fullpagetrue}%
8937             }%
8938             {}%
8939             \listxadd{\@msdata@list}{%
8940               \@msd@options@iffullpage%
8941               \ifluatex%
8942                 \csxdef{footnote@luatextextdir}{\the\textdir}%
8943                 \csxdef{footnote@luatexpardir}{\the\pardir}%

```

```

8944 \fi%
8945 \csdef{@this@crossref@start}{\add@msd@c:start:msdata}%
8946 \csdef{@this@crossref@end}{\add@msd@c:end:msdata}%
8947 \unexpanded{%
8948 \def\annot@start{}%
8949 \def\annot@end{}}%
8950 }%
8951 \noexpand\csuse{v\@msdata@series footnote}{\@msdata@series}{\@
expandonce\l@d@nums}{\ms@data@label}{\expandonce\data}}%
8952 \reset@msd@options@iffullpage%
8953 }%
8954 }%
8955 {\global\@msdata@insertedfrompreviouspagefalse}%
8956 }{}%
8957 \else%
8958 \ifcsdef{@msdata@\add@msd@cR @dataR}{%
8959 \ifnumless{\the\absline@numR-\@one}{\csuse{@msdata@\add@msd@cR
@stopabslinenumberR}}%
8960 {%
8961 \global\@msdata@insertedfrompreviouspagetrue%
8962 \letcs{\@data}{\@msdata@\add@msd@cR @dataR}%
8963 \edef\l@d@nums{%
8964 000}% Start page = we don't print it
8965 \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
8966 000}% Start subline number, for now, not used
8967 000}% End page number, we don't print it
8968 \ifnumless{\csuse{@msdata@\add@msd@cR @stopabslinenumberR}}{\@
csuse{@lastline@forpageR@\the\page@numR}}%
8969 {\csuse{@msdata@\add@msd@cR @stoplinenumberR}}%End line number
if in the same page
8970 {\csuse{@lastline@forpageR@\the\page@numR}}%Otherwise, last
number of the page
8971 |%
8972 000}% End sub line number, for now, not used
8973 \edfont@info%Font
8974 }%
8975 \@msd@options@fullpagefalse%
8976 \ifnumless{\csuse{@lastabsline@forpageR@\the\page@numR}}{\csuse{
@msdata@\add@msd@cR @stopabslinenumberR}+\@one}%
8977 {%
8978 \numdef{\@tmp}{\add@msd@cR+\@one}%
8979 \ifcsdef{@msdata@\@tmp @abslinenumberR}%
8980 {\ifnumequal{\csuse{@msdata@\@tmp @abslinenumberR}}{\csuse{
@lastabsline@forpageR@\the\page@numR}}%
8981 {}%
8982 {\@msd@options@fullpagetrue}%
8983 }%
8984 {\@msd@options@fullpagetrue}%

```

```

8985     }%
8986     {}%
8987     \listxadd{\@msdata@list}{%
8988         \@msd@options@iffullpage%
8989         \ifluatex%
8990             \csxdef{footnote@luatextextdir}{\the\textdir}%
8991             \csxdef{footnote@luatexpardir}{\the\pardir}%
8992         \fi%
8993         \csdef{@this@crossref@start}{\add@msd@cR:start:msdata}%
8994         \csdef{@this@crossref@end}{\add@msd@cR:end:msdata}%
8995         \unexpanded{%
8996             \def\annot@start{}%
8997             \def\annot@end{}%
8998         }%
8999         \noexpand\csuse{v\@msdata@series footnote}{\@msdata@series}{\{
expandonce\l@d@nums}{\ms@data@label}{\expandonce\@data}}}%
9000         \reset@msd@options@iffullpage%
9001     }%
9002 }%
9003 {\global\@msdata@insertedfrompreviouspagefalse}%
9004 }{}%
9005 \fi%
9006 \egroup%
9007 }%
9008 %

```

## XXXI.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 \newcommand{\insert@msdata}{%
9010     \def\do##1{##1}%
9011     \dolistloop{\@msdata@list}%
9012     \global\let\@msdata@list\relax%
9013 }%
9014 %

```

## XXXI.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.

```

9015 \newif\if@msd@options@fullpage%
9016 \newcommand{\@msd@options@iffullpage}[0]{%
9017     \if@msd@options@fullpage%
9018         \noexpand\toggletrue{nonum@}%

```

```

9019 \ifdefvoid{\ms@data@label}%
9020   {\noexpand\toggletrue{nosep@}}%
9021   }%
9022 \fi%
9023 }%
9024 \newcommand{\reset@msd@options@iffullpage}[0]{%
9025   \noexpand\togglefalse{nonum@}%
9026   \noexpand\togglefalse{nosep@}%
9027 }%
9028 %

```

## XXXII Arrays and tables

### XXXII.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.

```

9029 \newtoks\@emptytoks
9030
9031 %

```

The rest is from `amsmath`.

`\l@denvbody` A token register to contain the body.

```

9032 \newtoks\l@denvbody
9033
9034 %

```

`\addtol@denvbody` `\addtol@denvbody{arg}` adds `arg` to the token register `\l@denvbody`.

```

9035 \newcommand{\addtol@denvbody}[1]{%
9036   \global\l@denvbody\expandafter{\the\l@denvbody#1}}
9037 %

```

`\if@edtabular` A boolean set to true inside a `reledmac`’ tabular environment.

```

9038 \newif\if@edtabular%
9039 %

```

`\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`.

```

9040 \newcommand{\l@dcollect@body}[1]{%
9041   \ifmode\else%
9042     \\\%Ensure that we don't have spurious indentation at the very first
line, caused by some text before.
9043   \fi%
9044   \l@denbody{\expandafter#1\expandafter{\the\l@denbody}}%
9045   \edef\processl@denbody{\the\l@denbody\noexpand\end{\@currenenv}}%
9046   \l@denbody\@emptytoks \def\l@dbegin@stack{b}%
9047   \begingroup
9048     \edtabulartrue%
9049     \expandafter\let\csname\@currenenv\endcsname\l@dcollect@@body
9050     \edef\processl@denbody{\expandafter\noexpand\csname\@currenenv\endcsname}%
9051     \processl@denbody%
9052   }%
9053
9054 %

```

`\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.

```

9055 \def\l@dpush@begins#1\begin#2{%
9056   \ifx\end#2\else b\expandafter\l@dpush@begins\fi}
9057
9058 %

```

`\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.

```

9059 \def\l@dcollect@@body#1\end#2{%
9060   \edef\l@dbegin@stack{\l@dpush@begins#1\begin\end
9061     \expandafter\@gobble\l@dbegin@stack}%
9062   \ifx\@empty\l@dbegin@stack
9063     \endgroup
9064     \@checkend{#2}%
9065     \addtol@denbody{#1}%
9066   \else
9067     \addtol@denbody{#1\end{#2}}%
9068   \fi
9069   \processl@denbody % A little tricky! Note the grouping

```

```

9070 }
9071
9072 %

```

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>

```

## XXXII.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`.

### XXXII.2.1 Disabling and restoring commands

`\l@dtabnoexpands` More no expansion for critical and familiar footnotes in tabular environment.

```

9073 \newcommand*{\l@dtabnoexpands}{%
9074   \let\rtab=0%
9075   \let\ctab=0%
9076   \let\ltab=0%
9077   \let\rtabtext=0%
9078   \let\ltabtext=0%
9079   \let\ctabtext=0%
9080   \let\edbeforetab=0%
9081   \let\edaftertab=0%
9082   \let\edatleft=0%
9083   \let\edatright=0%
9084   \let\edvertline=0%
9085   \let\edvertdots=0%
9086   \let\edrowfill=0%
9087 }
9088
9089 %

```

`\disable@familiarnotes` Macros to disable and restore familiar notes, to prevent them from printing multiple times in `edtabularx` and `edarrayx` environments.

```

9090 \newcommand{\disable@familiarnotes}{%
9091   \unless\ifnofamiliar%
9092     \def\do##1{%
9093       \csletcs{footnote@##1}{footnote##1}%
9094       \expandafter\renewcommand \csname footnote##1\endcsname[1]{%
9095         \protected@csxdef{@thefnmark##1}{\csuse{thefootnote##1}}%
9096         \csuse{@footnotemark##1}%
9097       }%
9098     }%
9099     \dolistloop{\@series}%
9100   \fi%
9101 }%
9102 \newcommand{\restore@familiarnotes}{%
9103   \unless\ifnofamiliar%
9104     \def\do##1{%

```

```

9105 \csletcs{footnote##1}{footnote@@##1}%
9106 }%
9107 \dolistloop{\@series}%
9108 \fi%
9109 }%
9110 %
9111 %

```

`\disable@sidenotes` The same for sidenotes.

```

\restore@sidenotes
9112 \newcommand{\disable@sidenotes}{%
9113 \let\@@ledrightnote\ledrightnote%
9114 \let\@@ledleftnote\ledleftnote%
9115 \let\@@ledsidenote\ledsidenote%
9116 \let\ledrightnote@gobble%
9117 \let\ledleftnote@gobble%
9118 \let\ledsidenote@gobble%
9119 }%
9120 \newcommand{\restore@sidenotes}{%
9121 \let\ledrightnote\@@ledrightnote%
9122 \let\ledleftnote\@@ledleftnote%
9123 \let\ledsidenote\@@ledsidenote%
9124 }%
9125 %

```

`\disable@endnotes` The same for endnotes.

```

\restore@endnotes
9126 \newcommand{\disable@endnotes}{%
9127 \unless\ifnoend%
9128 \def\do##1{%
9129 \csletcs{##1@endnote}{##1endnote}%
9130 \expandafter\renewcommand \csname ##1endnote\endcsname[1]{}%
9131 }%
9132 \dolistloop{\@series}%
9133 \fi%
9134 }%
9135 \newcommand{\restore@endnotes}{%
9136 \unless\ifnofamiliar%
9137 \def\do##1{%
9138 \csletcs{##1endnote}{##1@endnote}%
9139 }%
9140 \dolistloop{\@series}%
9141 \fi%
9142 }%
9143 %

```

`\disable@notes` Disable/restore side, familiar and end notes.

```

\restore@notes
9144 \newcommand{\disable@notes}{%
9145 \disable@sidenotes%

```

```

9146 \disable@famiarnotes%
9147 \disable@endnotes%
9148 }%
9149 \newcommand{\restore@notes}{%
9150 \restore@sidenotes%
9151 \restore@famiarnotes%
9152 \restore@endnotes%
9153 }%
9154 %

```

**\EDTEXT** We need to be able to modify the `\edtext` macros and also restore their original definitions.

**\xedtext**

```

9155 \let\EDTEXT=\edtext
9156 \newcommand{\xedtext}[2]{\EDTEXT{#1}{#2}}
9157 %

```

**\EDLABEL** We need to be able to modify and restore the `\edlabel` macro.

**\xedlabel**

```

9158 \let\EDLABEL=\edlabel
9159 \newcommand*{\xedlabel}[1]{\EDLABEL{#1}}
9160 %

```

**\xedindex** Macros supporting modification and restoration of `\edindex`.

**\nulledindex**

```

9161 \AtBeginDocument{\let\xedindex\edindex}%
9162 \newcommand{\nulledindex}[2][\jobname]{\@bsphack\@esphack}
9163
9164 %

```

**\@line@num** Macro supporting restoration of `\linenum`.

```

9165 \let\@line@num=\linenum
9166 %

```

**\l@dgobbleoptarg** `\l@dgobbleoptarg[⟨arg⟩]{⟨arg⟩}` replaces these two arguments (first is optional) by `\relax`.

```

9167 \newcommand*{\l@dgobbleoptarg}[2][\relax]%
9168
9169 %

```

**\l@secondmandarg** `\l@secondoptarg[⟨arg⟩]{⟨arg⟩}` gobble the first (optional) argument, and expand to the second (mandatory) argument.

```

9170 \NewExpandableDocumentCommand{\l@secondmandarg}{om}{#2}%
9171 %

```

```

\Relax72 \let\Relax=\relax
\NEXT73 \let\NEXT=\next
9174
9175 %

```

**\l@dmodforedtext** Modify and restore various macros for when \edtext is used.

**\l@restoreforedtext**

```

9176 \newcommand{\l@dmodforedtext}{%
9177   \let\edtext\relax
9178   \def\do##1{\global\csletcs{##1footnote}{\l@gobbleoptarg}}%
9179   \dolistloop{\@series}%
9180   \let\edindex\nulledindex
9181   \let\linenum\@gobble}
9182 \newcommand{\l@restoreforedtext}{%
9183   \def\do##1{\global\csletcs{##1footnote}{##1@footnote}}
9184   \dolistloop{\@series}%
9185   \let\edindex\xedindex}
9186 %

```

**\l@dnullfills** Nullify and restore some column fillers, etc.

**\l@restorefills**

```

9187 \newcommand{\l@dnullfills}{%
9188   \def\edlabel##1{%
9189     \def\edrowfill##1##2##3{%
9190     }
9191   \newcommand{\l@restorefills}{%
9192     \def\edrowfill##1##2##3{\@EDROWFILL@{##1}{##2}{##3}}%
9193   }
9194
9195 %

```

**\letsforverteilen** Gathers some lets and other code that is common to the \*verteilen\* macros.

```

9196 \newcommand{\letsforverteilen}{%
9197   \let\edtext\xedtext
9198   \let\edindex\xedindex
9199   \def\do##1{\global\csletcs{##1footnote}{##1@footnote}}
9200   \dolistloop{\@series}%
9201   \let\linenum\@line@num
9202   \hilfe skip=\l@dcwidth%
9203   \advance\hilfe skip by -\wd\hilfe box
9204   \def\edlabel##1{\xedlabel{##1}}
9205
9206 %

```

**\disablel@dtabfeet** Declarations for using or using \edtext inside tabulars. The default at this point is for \edtext.

**\enablel@dtabfeet**

```

9207 \newcommand\disablel@dtabfeet{\l@dmodforedtext}%
9208 \newcommand\enablel@dtabfeet{\l@restoreforedtext}%
9209 %

```

### XXXII.2.2 Counters, boxes and lengths

`\l@dampcount` `\l@dampcount` is a counter for the & column dividers and `\l@dcolcount` is a counter for the columns.

```

9210 \newcount\l@dampcount
9211   \l@dampcount=1\relax
9212 \newcount\l@dcolcount
9213   \l@dcolcount=0\relax
9214
9215 %

```

`\hilfsbox` Some (temporary) helper items.

```

\hilfsskip
\Hilfsbox
\hilfscount
9216 \newbox\hilfsbox
9217 \newskip\hilfsskip
9218 \newbox\Hilfsbox
9219 \newcount\hilfscount
9220
9221 %

```

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).

```

9222 \newdimen\dcoli
9223 \newdimen\dcolii
9224 \newdimen\dcoliii
9225 \newdimen\dcoliv
9226 \newdimen\dcolv
9227 \newdimen\dcolvi
9228 \newdimen\dcolvii
9229 \newdimen\dcolviii
9230 \newdimen\dcolix
9231 \newdimen\dcolx
9232 \newdimen\dcolxi
9233 \newdimen\dcolxii
9234 \newdimen\dcolxiii
9235 \newdimen\dcolxiv
9236 \newdimen\dcolxv
9237 \newdimen\dcolxvi
9238 \newdimen\dcolxvii
9239 \newdimen\dcolxviii
9240 \newdimen\dcolxix
9241 \newdimen\dcolxx
9242 \newdimen\dcolxxi
9243 \newdimen\dcolxxii
9244 \newdimen\dcolxxiii
9245 \newdimen\dcolxxiv
9246 \newdimen\dcolxxv
9247 \newdimen\dcolxxvi
9248 \newdimen\dcolxxvii

```

```

9249 \newdimen\dcollxxviii
9250 \newdimen\dcollxxix
9251 \newdimen\dcollxxx
9252 \newdimen\dcollerr    % added for error handling
9253
9254 %

```

**\l@dcollwidth** This is a cunning way of storing the column widths indexed by the column number \l@dcollcount, like an array. (was \Dimenzuordnung)

```

9255 \newcommand{\l@dcollwidth}{\ifcase \the\l@dcollcount \dcoli %???
9256 \or \dcoli \or \dcolii \or \dcoliii
9257 \or \dcoliv \or \dcolv \or \dcolvi
9258 \or \dcolvii \or \dcolviii \or \dcolix \or \dcolx
9259 \or \dcolxi \or \dcolxii \or \dcolxiii
9260 \or \dcolxiv \or \dcolxv \or \dcolxvi
9261 \or \dcolxvii \or \dcolxviii \or \dcolxix \or \dcolxx
9262 \or \dcolxxi \or \dcolxxii \or \dcolxxiii
9263 \or \dcolxxiv \or \dcolxxv \or \dcolxxvi
9264 \or \dcolxxvii \or \dcolxxviii \or \dcolxxix \or \dcolxxx
9265 \else \dcollerr \fi}
9266
9267 %

```

**\step1@dcollcount** This increments the column counter, and issues an error message if it is too large.

```

9268 \newcommand*{\step1@dcollcount}{\advance\l@dcollcount\@ne
9269 \ifnum\l@dcollcount>30\relax
9270 \led@err@TooManyColumns
9271 \fi}
9272
9273 %

```

**\l@dsetmaxcollwidth** Sets the column width to the maximum value seen so far.

```

9274 \newcommand{\l@dsetmaxcollwidth}{%
9275 \ifdim\l@dcollwidth < \wd\hilsbox
9276 \l@dcollwidth = \wd\hilsbox
9277 \else \relax \fi}
9278
9279 %

```

**\measuremcell** Measure (recursively) the width required for a math cell.

```

9280 \def\measuremcell #1&{%
9281 \ifx #1\ \ifnum\l@dcollcount=0\let\NEXT\relax%
9282 \else\l@dcheckcols%
9283 \l@dcollcount=0%
9284 \let\NEXT\measuremcell%
9285 \fi%

```

```

9286 \else\setbox\hilfsbox=\hbox{$\displaystyle{#1}$}%
9287 \step1@dcolcount%
9288 \l@setmaxcolwidth%
9289 \let\NEXT\measuremcell%
9290 \fi\NEXT}
9291
9292 %

```

**\measuretcell** Measure (recursively) the width required for a text cell.

```

9293 \def\measuretcell #1{%
9294 \ifx #1\ \ifnum\l@dcolcount=0\let\NEXT\relax%
9295 \else\l@dcheckcols%
9296 \l@dcolcount=0%
9297 \let\NEXT\measuretcell%
9298 \fi%
9299 \else\setbox\hilfsbox=\hbox{#1}%
9300 \step1@dcolcount%
9301 \l@setmaxcolwidth%
9302 \let\NEXT\measuretcell%
9303 \fi\NEXT}
9304
9305 %

```

**\measuremrow** Measure (recursively) the width required for a math row.

```

9306 \def\measuremrow #1{%
9307 \ifx #1\let\NEXT\relax%
9308 \else\measuremcell #1\&\&\&%
9309 \let\NEXT\measuremrow%
9310 \fi\NEXT}
9311 %

```

**\measuretrow** Measure (recursively) the width required for a text row.

```

9312 \def\measuretrow #1{%
9313 \ifx #1\let\NEXT\relax%
9314 \else\measuretcell #1\&\&\&%
9315 \let\NEXT\measuretrow%
9316 \fi\NEXT}
9317
9318 %

```

**\edtabcolsep** The length \edtabcolsep controls the distance between columns.

```

9319 \newskip\edtabcolsep
9320 \global\edtabcolsep=10pt
9321
9322 %

```



```
\variab23 \newcommand{\variab}{\relax}
```

```
9324  
9325 %
```

**\l@dccheckcols** Check that the number of columns is consistent.

```
9326 \newcommand*{\l@dccheckcols}{%  
9327 \ifnum\l@dcolcount=1\relax  
9328 \else  
9329 \ifnum\l@dampcount=1\relax  
9330 \else  
9331 \ifnum\l@dcolcount=\l@dampcount\relax  
9332 \else  
9333 \l@d@err@UnequalColumns  
9334 \fi  
9335 \fi  
9336 \l@dampcount=\l@dcolcount  
9337 \fi}  
9338  
9339 %
```

**\edfilldimen** A length.

```
9340 \newdimen\edfilldimen  
9341 \edfilldimen=0pt  
9342  
9343 %
```

**\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**

```
9344 \newcounter{addcolcount}  
9345 \renewcommand{\theadcolcount}{\romannumeral \c@addcolcount}  
9346 %
```

### XXXII.2.3 Tabular typesetting

**\setmcellright** Typeset (recursively) cells of display math right justified.

```
9347 \def\setmcellright #1{\def\edlabel##1}%  
9348 \let\edindex\nulledindex  
9349 \ifx #1\ \ifnum\l@dcolcount=0%\removelastskip  
9350 \let\Next\relax%  
9351 \else\l@dcolcount=0%  
9352 \let\Next=\setmcellright%  
9353 \fi%  
9354 \else%  
9355 \disablel@dtabfeet%
```

```

9356      \stepl@dc colcount%
9357      \disable@notes%
9358      \setbox\hilfsbox=\hbox{$\displaystyle{#1}$}%
9359      \restore@notes%
9360      \letsforverteilen%
9361      \hskip\hilfsskip$\displaystyle{#1}$%
9362      \hskip\edtabcolsep%
9363      \let\Next=\setmcellright%
9364      \fi\Next}
9365
9366 %

```

**\settcellright** Typeset (recursively) cells of text right justified.

```

9367 \def\settcellright #1&{\def\edlabel##1{}%
9368   \let\edindex\nulledindex
9369   \ifx #1\\ \ifnum\l@dc colcount=0%\removelastskip
9370     \let\Next\relax%
9371   \else\l@dc colcount=0%
9372     \let\Next=\settcellright%
9373   \fi%
9374 \else%
9375   \disablel@dtabfeet%
9376   \stepl@dc colcount%
9377   \disable@notes%
9378   \setbox\hilfsbox=\hbox{#1}%
9379   \restore@notes%
9380   \letsforverteilen%
9381   \hskip\hilfsskip#1%
9382   \hskip\edtabcolsep%
9383   \let\Next=\settcellright%
9384   \fi\Next}
9385 %

```

**\setmcellleft** Typeset (recursively) cells of display math left justified.

```

9386 \def\setmcellleft #1&{\def\edlabel##1{}%
9387   \let\edindex\nulledindex
9388   \ifx #1\\ \ifnum\l@dc colcount=0 \let\Next\relax%
9389   \else\l@dc colcount=0%
9390     \let\Next=\setmcellleft%
9391   \fi%
9392 \else \disablel@dtabfeet%
9393   \stepl@dc colcount%
9394   \disable@notes%
9395   \setbox\hilfsbox=\hbox{$\displaystyle{#1}$}%
9396   \restore@notes%
9397   \letsforverteilen%
9398   $\displaystyle{#1}$\hskip\hilfsskip\hskip\edtabcolsep%
9399   \let\Next=\setmcellleft%

```

```

9400 \fi\Next}
9401
9402 %

```

**\settcclleft** Typeset (recursively) cells of text left justified.

```

9403 \def\settcclleft #1{\def\edlabel##1}%
9404 \let\edindex\nulledindex
9405 \ifx #1\ \ifnum\l@dc@count=0 \let\Next\relax%
9406 \else\l@dc@count=0%
9407 \let\Next=\settcclleft%
9408 \fi%
9409 \else \disablel@dtabfeet%
9410 \stepl@dc@count%
9411 \disable@notes%
9412 \setbox\hilfsbox=\hbox{#1}%
9413 \restore@notes%
9414 \letsforverteilen%
9415 #1\hskip\hlfsskip\hskip\edtabcolsep%
9416 \let\Next=\settcclleft%
9417 \fi\Next}
9418 %

```

**\setmcellcenter** Typeset (recursively) cells of display math centered.

```

9419 \def\setmcellcenter #1{\def\edlabel##1}%
9420 \let\edindex\nulledindex
9421 \ifx #1\ \ifnum\l@dc@count=0\let\Next\relax%
9422 \else\l@dc@count=0%
9423 \let\Next=\setmcellcenter%
9424 \fi%
9425 \else \disablel@dtabfeet%
9426 \stepl@dc@count%
9427 \disable@notes%
9428 \setbox\hilfsbox=\hbox{$\displaystyle{#1}$}%
9429 \restore@notes%
9430 \letsforverteilen%
9431 \hskip 0.5\hlfsskip$\displaystyle{#1}$\hskip0.5\hlfsskip%
9432 \hskip\edtabcolsep%
9433 \let\Next=\setmcellcenter%
9434 \fi\Next}
9435
9436 %

```

**\settcclcenter** Typeset (recursively) cells of text centered.

```

9437 \def\settcclcenter #1{\def\edlabel##1}%
9438 \let\edindex\nulledindex
9439 \ifx #1\ \ifnum\l@dc@count=0 \let\Next\relax%
9440 \else\l@dc@count=0%

```

```

9441         \let\Next=\settcellcenter%
9442         \fi%
9443     \else \disablel@dtabfeet%
9444           \stepl@dcolcount%
9445           \disable@notes%
9446           \setbox\hilfsbox=\hbox{#1}%
9447           \restore@notes%
9448           \letsforverteilen%
9449           \hskip 0.5\hilfsskip #1\hskip 0.5\hilfsskip%
9450           \hskip\edtabcolsep%
9451           \let\Next=\settcellcenter%
9452     \fi\Next}
9453
9454 %

```

```

\NEXT55 \let\NEXT=\relax

```

```

9456
9457 %

```

**\setmrowright** Typeset (recursively) rows of right justified math.

```

9458 \def\setmrowright #1\{%
9459     \ifx #1& \let\NEXT\relax
9460     \else \centerline{\setmcellright #1&\\&\\&}
9461           \let\NEXT=\setmrowright
9462     \fi\NEXT}
9463 %

```

**\settroright** Typeset (recursively) rows of right justified text.

```

9464 \def\settroright #1\{%
9465     \ifx #1& \let\NEXT\relax
9466     \else \centerline{\settcellright #1&\\&\\&}
9467           \let\NEXT=\settroright
9468     \fi\NEXT}
9469
9470 %

```

**\setmrowleft** Typeset (recursively) rows of left justified math.

```

9471 \def\setmrowleft #1\{%
9472     \ifx #1&\let\NEXT\relax
9473     \else \centerline{\setmcellleft #1&\\&\\&}
9474           \let\NEXT=\setmrowleft
9475     \fi\NEXT}
9476 %

```

**\settrorleft** Typeset (recursively) rows of left justified text.

```

9477 \def\settrrowleft #1\{\%
9478   \ifx #1& \let\NEXT\relax
9479   \else \centerline{\settcelleleft #1&\&\&\&}
9480   \let\NEXT=\settrrowleft
9481   \fi\NEXT}
9482
9483 %

```

**\setmrowcenter** Typeset (recursively) rows of centered math.

```

9484 \def\setmrowcenter #1\{\%
9485   \ifx #1& \let\NEXT\relax%
9486   \else \centerline{\setmcellcenter #1&\&\&\&}
9487   \let\NEXT=\setmrowcenter
9488   \fi\NEXT}
9489 %

```

**\settrrowcenter** Typeset (recursively) rows of centered text.

```

9490 \def\settrrowcenter #1\{\%
9491   \ifx #1& \let\NEXT\relax
9492   \else \centerline{\settrcellcenter #1&\&\&\&}
9493   \let\NEXT=\settrrowcenter
9494   \fi\NEXT}
9495
9496 %

```

**\nullsetzen** 97 \newcommand{\nullsetzen}{\%

```

9498   \step1@dcolcount%
9499   \l@dcolwidth=0pt%
9500   \ifnum\l@dcolcount=30\let\NEXT\relax%
9501   \l@dcolcount=0\relax
9502   \else\let\NEXT\nullsetzen%
9503   \fi\NEXT}
9504
9505 %

```

**\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.

```

9506 \newcommand{\edatleft}[3][\@empty]{\%
9507   \ifx#1\@empty
9508     \vbox to 10pt{\vss\hbox{\$ \left#2\vrule width0pt height #3
9509       depth 0pt \right. \$\hss}\vfil}
9510   \else
9511     \vbox to 4pt{\vss\hbox{\$#1\left#2\vrule width0pt height #3
9512       depth 0pt \right. \$}\vfil}
9513   \fi}
9514 %

```

**\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.

```

9515 \newcommand{\edatright}[3][\@empty]{%
9516   \ifx#1\@empty
9517     \vbox to 10pt{\vss\hbox{$\left.\vrule width0pt height #3
9518               depth 0pt \right#2 $\hss}\vfil}
9519   \else
9520     \vbox to 4pt{\vss\hbox{$\left.\vrule width0pt height #3
9521               depth 0pt \right#2 #1 $\vfil}
9522   \fi}
9523
9524 %

```

**\edvertline** `\edvertline{ $\langle len \rangle$ }` vertical line  $\langle len \rangle$  high.

```

9525 \newcommand{\edvertline}[1]{\vbox to 8pt{\vss\hbox{\vrule height #1}\vfil}}
9526
9527 %

```

**\edvertdots** `\edvertdots{ $\langle len \rangle$ }` vertical dotted line  $\langle len \rangle$  high.

```

9528 \newcommand{\edvertdots}[1]{\vbox to 1pt{\vss\vbox to #1%
9529   {\cleaders\hbox{$\math\hbox{.}\vbox to 0.5em{ }$}\vfil}}}
9530
9531 %

```

**\l@dtabaddcols** `\l@dtabaddcols{ $\langle startcol \rangle$ }{ $\langle endcol \rangle$ }` adds the widths of the columns  $\langle startcol \rangle$  through  $\langle endcol \rangle$  to `\edfilldimen`. It is a  $\text{\LaTeX}$  style reimplementaion of the original `\@add@`.

```

9532 \newcommand{\l@dtabaddcols}[2]{%
9533   \l@dccheckstartend{#1}{#2}%
9534   \ifl@dstartendok
9535     \setcounter{addcolcount}{#1}%
9536     \@whilenum \value{addcolcount}<#2\relax \do
9537     {\advance\edfilldimen by \the \csname dcol\theadcolcount\endcsname
9538     \advance\edfilldimen by \edtabcolsep
9539     \stepcounter{addcolcount}}%
9540     \advance\edfilldimen by \the \csname dcol\theadcolcount\endcsname
9541   \fi
9542 }
9543
9544 %

```

**\ifl@dstartendok** `\l@dccheckstartend{ $\langle startcol \rangle$ }{ $\langle endcol \rangle$ }` checks that the values of  $\langle startcol \rangle$  and  $\langle endcol \rangle$  are sensible. If they are then `\ifl@dstartendok` is set TRUE, otherwise it is set FALSE.

```

9545 \newif\ifl@dstartendok
9546 \newcommand{\l@dccheckstartend}[2]{%

```

```

9547 \l@dstartendoktrue
9548 \ifnum #1<\@ne
9549   \l@dstartendokfalse
9550   \led@err@LowStartColumn
9551 \fi
9552 \ifnum #2>30\relax
9553   \l@dstartendokfalse
9554   \led@err@HighEndColumn
9555 \fi
9556 \ifnum #1>#2\relax
9557   \l@dstartendokfalse
9558   \led@err@ReverseColumns
9559 \fi
9560 }
9561
9562 %

```

`\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.

```

9563 \newcommand*{\edrowfill}[3]{%
9564   \l@dtabaddcols{#1}{#2}%
9565   \hb@xt@ \the\l@dcowidth{\hb@xt@ \the\edfilldimen{#3}\hss}}
9566 \let\@edrowfill=\edrowfill
9567 \def\@EDROWFILL@#1#2#3{\@edrowfill@{#1}{#2}{#3}}
9568
9569 %

```

`\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`.

```

9570 \newcommand{\leftltab}[1]{%
9571   \hb@xt@ \z@{\vbox{\edtabindent%
9572     \moveleft\Hilfsskip\hbox{\ #1}}\hss}}
9573
9574 %

```

`\leftrtab` `\leftrtab{<text>}{<math>}` for `\edbeforetab` in `\rtab`.

```

9575 \newcommand{\leftrtab}[2]{%
9576   #2\hb@xt@ \z@{\vbox{\edtabindent%
9577     \advance\Hilfsskip by\dcoli%
9578     \moveleft\Hilfsskip\hbox{\ #1}}\hss}}
9579
9580 %

```

`\leftctab` `\leftctab{<text>}{<math>}` for `\edbeforetab` in `\ctab`.

```

9581 \newcommand{\leftctab}[2]{%
9582     \hb@xt@#2{\vbox{\edtabindent\l@dcolcount=\l@dampcount%
9583     \advance\Hilfsskip by 0.5\dcoli%
9584     \setbox\hilfsbox=\hbox{\def\edlabel##1{}%
9585     \disablel@dtabfeet$\displaystyle{#2}$}%
9586     \advance\Hilfsskip by -0.5\wd\hilfsbox%
9587     \moveleft\Hilfsskip\hbox{\ #1}}\hss}%
9588     #2}
9589
9590 %

```

`\rightctab` `\rightctab{<math>}{<text>}` for `\edaftertab` in `\ctab`.

```

9591 \newcommand{\rightctab}[2]{%
9592     \setbox\hilfsbox=\hbox{\def\edlabel##1{}%
9593     \disablel@dtabfeet#2}\l@dampcount=\l@dcolcount%
9594     #1\hb@xt@#2{\vbox{\edtabindent\l@dcolcount=\l@dampcount%
9595     \advance\Hilfsskip by 0.5\l@dcolwidth%
9596     \advance\Hilfsskip by -\wd\hilfsbox%
9597     \setbox\hilfsbox=\hbox{\def\edlabel##1{}%
9598     \disablel@dtabfeet$\displaystyle{#1}$}%
9599     \advance\Hilfsskip by -0.5\wd\hilfsbox%
9600     \advance\Hilfsskip by \edtabcolsep%
9601     \moveright\Hilfsskip\hbox{ #2}}\hss}%
9602     }
9603
9604 %

```

`\rightltab` `\rightltab{<math>}{<text>}` for `\edaftertab` in `\ltab`.

```

9605 \newcommand{\rightltab}[2]{%
9606     \setbox\hilfsbox=\hbox{\def\edlabel##1{}%
9607     \disablel@dtabfeet#2}\l@dampcount=\l@dcolcount%
9608     #1\hb@xt@#2{\vbox{\edtabindent\l@dcolcount=\l@dampcount%
9609     \advance\Hilfsskip by\l@dcolwidth%
9610     \advance\Hilfsskip by-\wd\hilfsbox%
9611     \setbox\hilfsbox=\hbox{\def\edlabel##1{}%
9612     \disablel@dtabfeet$\displaystyle{#1}$}%
9613     \advance\Hilfsskip by-\wd\hilfsbox%
9614     \advance\Hilfsskip by\edtabcolsep%
9615     \moveright\Hilfsskip\hbox{ #2}}\hss}%
9616     }
9617
9618 %

```

`\rightrtab` `\rightrtab{<math>}{<text>}` for `\edaftertab` in `\rtab`.

```

9619 \newcommand{\rightrtab}[2]{%

```



```

9620 \setbox\hilfsbox=\hbox{\def\edlabel##1}{%
9621 \disablel@dtabfeet#2}%
9622 #1\hb@xt@#2{\vbox{\edtabindent%
9623 \advance\Hilfsskip by-\wd\hilfsbox%
9624 \advance\Hilfsskip by\edtabcolsep%
9625 \moveright\Hilfsskip\hbox{ #2}}\hss}%
9626 }
9627
9628 %

```

**\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.

```

9629 \newcommand{\rtab}[1]{%
9630 \l@dnnullfills
9631 \def\edbeforetab##1##2{\lefttab{##1}{##2}}%
9632 \def\edaftertab##1##2{\righttab{##1}{##2}}%
9633 \measurebody{#1}%
9634 \l@drestorefills
9635 \variab
9636 \setmrowright #1\&\&\&%
9637 \enablel@dtabfeet}
9638
9639 %

```

**\measurebody** `\measurebody{<body>}` measures the array `<body>`.

```

9640 \newcommand{\measurebody}[1]{%
9641 \disablel@dtabfeet%
9642 \l@dcolcount=0%
9643 \nullsetzen%
9644 \l@dcolcount=0
9645 \measuremrow #1\&\&\&%
9646 \global\l@dampcount=1}
9647
9648 %

```

**\rtabtext** `\rtabtext{<body>}` typesets `<body>` as a tabular with the entries right justified.

```

9649 \newcommand{\rtabtext}[1]{%
9650 \l@dnnullfills
9651 \measuretbody{#1}%
9652 \l@drestorefills
9653 \variab
9654 \setthrowright #1\&\&\&%
9655 \enablel@dtabfeet}
9656
9657 %

```

`\measuretbody` `\measuretbody{⟨body⟩}` measures the tabular `⟨body⟩`.

```

9658 \newcommand{\measuretbody}[1]{%
9659   \disable@notes%
9660   \disablel@dtabfeet%
9661   \l@dcolcount=0%
9662   \nullsetzen%
9663   \l@dcolcount=0
9664   \measuretrrow #1\\&\\%
9665   \restore@notes%
9666   \global\l@dampcount=1}
9667
9668 %

```

`\ltab` Array with entries left justified.

```

\edbeforetab
\edaftertab
9669 \newcommand{\ltab}[1]{%
9670   \l@dnnullfills
9671   \def\edbeforetab##1##2{\leftltab{##1}{##2}}%
9672   \def\edaftertab##1##2{\rightltab{##1}{##2}}%
9673   \measuretbody{#1}%
9674   \l@drestorefills
9675   \variab
9676   \setmrowleft #1\\&\\%
9677   \enablel@dtabfeet}
9678
9679 %

```

`\ltabtext` Tabular with entries left justified.

```

9680 \newcommand{\ltabtext}[1]{%
9681   \l@dnnullfills
9682   \measuretbody{#1}%
9683   \l@drestorefills
9684   \variab
9685   \settrrowleft #1\\&\\%
9686   \enablel@dtabfeet}
9687
9688 %

```

`\ctab` Array with centered entries.

```

\edbeforetab
\edaftertab
9689 \newcommand{\ctab}[1]{%
9690   \l@dnnullfills
9691   \def\edbeforetab##1##2{\leftctab{##1}{##2}}%
9692   \def\edaftertab##1##2{\rightctab{##1}{##2}}%
9693   \measuretbody{#1}%
9694   \l@drestorefills
9695   \variab
9696   \setmrowcenter #1\\&\\%

```

```

9697 \enablel@dtabfeet}
9698
9699 %

```

**\ctabtext** Tabular with entries centered.

```

9700 \newcommand{\ctabtext}[1]{%
9701 \l@dnnullfills
9702 \measuretbody{#1}%
9703 \l@drestorefills
9704 \variab
9705 \settrrowcenter #1\\&\\%
9706 \enablel@dtabfeet}
9707
9708 %

```

```

\spreadtext09 \newcommand{\spreadtext}[1]{%\l@dcolcount=\l@dampcount%
9710 \hb@xt@ \the\l@dcolwidth{\hbox{#1}\hss}}
9711 %

```

```

\spreadmath12 \newcommand{\spreadmath}[1]{%
9713 \hb@xt@ \the\l@dcolwidth{\hbox{$\displaystyle{#1}$}\hss}}
9714
9715 %

```

**\HILFSskip** More helpers.

```

\Hilfsskip
9716 \newskip\HILFSskip
9717 \newskip\Hilfsskip
9718
9719 %

```

```

\EDTABINDENT20 \newcommand{\EDTABINDENT}{%
9721 \ifnum\l@dcolcount=30\let\NEXT\relax\l@dcolcount=0%
9722 \else\step\l@dcolcount%
9723 \advance\Hilfsskip by\l@dcolwidth%
9724 \ifdim\l@dcolwidth=0pt\advance\hilfscount\@ne
9725 \else\advance\Hilfsskip by \the\hilfscount\edtabcolsep%
9726 \hilfscount=1\fi%
9727 \let\NEXT=\EDTABINDENT%
9728 \fi\NEXT}%
9729 %

```

**\edtabindent** (was \tabindent)

```

9730 \newcommand{\edtabindent}{%
9731     \l@dcollectioncount=0\relax
9732     \Hilfsskip=0pt%
9733     \hilfscount=1\relax
9734     \EDTABINDENT%
9735     \hilfsskip=\hsize%
9736     \advance\hilfsskip -\Hilfsskip%
9737     \Hilfsskip=0.5\hilfsskip%
9738 }%
9739
9740 %

```

**\EDTAB** (was \TAB)

```

9741 \def\EDTAB #1|#2|{%
9742     \setbox\tabhilfbox=\hbox{$\displaystyle{#1}$}%
9743     \setbox\tabHilfbox=\hbox{$\displaystyle{#2}$}%
9744     \advance\tabelskip -\wd\tabhilfbox%
9745     \advance\tabelskip -\wd\tabHilfbox%
9746     \unhbox\tabhilfbox\hskip\tabelskip%
9747     \unhbox\tabHilfbox}%
9748
9749 %

```

**\EDTABtext** (was \TABtext)

```

9750 \def\EDTABtext #1|#2|{%
9751     \setbox\tabhilfbox=\hbox{#1}%
9752     \setbox\tabHilfbox=\hbox{#2}%
9753     \advance\tabelskip -\wd\tabhilfbox%
9754     \advance\tabelskip -\wd\tabHilfbox%
9755     \unhbox\tabhilfbox\hskip\tabelskip%
9756     \unhbox\tabHilfbox}%
9757 %

```

**\tabhilfbox** Further helpers.

**\tabHilfbox**

```

9758 \newbox\tabhilfbox
9759 \newbox\tabHilfbox
9760
9761 %

```

## XXXII.2.4 Environments

**edarrayl edarrayc edarrayr** The ‘environment’ forms for \ltab, \ctab and \rtab.

```

9762 \newenvironment{edarrayl}{\l@dcollection@body\ltab}{}
9763 \newenvironment{edarrayc}{\l@dcollection@body\ctab}{}
9764 \newenvironment{edarrayr}{\l@dcollection@body\rtab}{}
9765
9766 %

```

edtabularl edtabularc edtabularr The ‘environment’ forms for \ltabtext, \ctabtext and \rtabtext.

```

9767 \newenvironment{edtabularl}{\l@collect@body\ltabtext}{}
9768 \newenvironment{edtabularc}{\l@collect@body\ctabtext}{}
9769 \newenvironment{edtabularr}{\l@collect@body\rtabtext}{}
9770
9771 %

```

### XXXIII 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
9772 \newcommand{\initnumbering@quote}{
9773   \ifnoquotation@else
9774     \renewcommand{\quotation}{\par\leavevmode%
9775       \parindent=1.5em%
9776       \skipnumbering%
9777       \ifautopar%
9778         \vskip-\parskip%
9779       \else%
9780         \vskip\topsep%
9781       \fi%
9782       \global\leftskip=\leftmargin%
9783       \global\rightskip=\leftmargin%
9784     }
9785     \renewcommand{\endquotation}{\par%
9786       \global\leftskip=0pt%
9787       \global\rightskip=0pt%
9788       \leavevmode%
9789       \skipnumbering%
9790       \ifautopar%
9791         \vskip-\parskip%
9792       \else%
9793         \vskip\topsep%
9794       \fi%
9795     }
9796     \renewcommand{\quote}{\par\leavevmode%
9797       \parindent=0pt%
9798       \skipnumbering%
9799       \ifautopar%
9800         \vskip-\parskip%
9801       \else%
9802         \vskip\topsep%
9803       \fi%

```

```

9804             \global\leftskip=\leftmargin%
9805             \global\rightskip=\leftmargin%
9806         }
9807         \renewcommand{\endquote}{\par%
9808             \global\leftskip=0pt%
9809             \global\rightskip=0pt%
9810             \leavevmode%
9811             \skipnumbering%
9812             \ifautopar%
9813                 \vskip-\parskip%
9814             \else%
9815                 \vskip\topsep%
9816             \fi%
9817         }
9818     \fi
9819 }
9820 %

```

## XXXIV Section's title commands

### XXXIV.1 Commands to disable some feature

**\ledsectnotoc** The `\ledsectnotoc` only disables the `\addcontentsline` macro.

```

9821 \newcommand{\ledsectnotoc}{\let\addcontentsline\@gobblethree}
9822 %

```

**\ledsectnomark** The `\ledsectnomark` only disables the `\chaptermark`, `\sectionmark` and `\subsectionmark` macros.

```

9823 \newcommand{\ledsectnomark}{%
9824     \let\chaptermark\@gobble%
9825     \let\sectionmark\@gobble%
9826     \let\subsectionmark\@gobble%
9827 }
9828 %

```

### XXXIV.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 `reledpar` 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  $\TeX$  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  $\TeX$  sectioning command.
  4. This last command is called when the `pstart` is effectively printed.

### XXXIV.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).

```
9829 \catcode`\#=12
9830 \notbool{@noeled@sec}{%
9831 %
```

`\beforeeledchapter` For technical reasons, not yet solved, page-breaking before chapters can't be made automatically by `eledmac`. Users have to use `\beforeeledchapter`.

```
9832 \ifl@dmemoir
9833   \newcommand\beforeeledchapter{%
9834     \clearforchapter%
9835   }
9836 \else
9837   \newcommand\beforeeledchapter{%
9838     \if@openright%
9839       \cleardoublepage%
9840     \else%
9841       \clearpage%
9842     \fi%
9843   }
9844 \fi
9845 %
```

### XXXIV.4 Auxiliary commands

`\print@leftmargin@eledsection` and `\print@rightmargin@eledsection` are added by `reledmac` inside the code of sectioning command, in order to affix lines numbers. They include tests for RTL languages.

```

9846 \def\print@rightmargin@eledsection{%
9847   \if@eled@sectioning%
9848     \begingroup%
9849     \if@RTL%
9850       \let\llap\rlap%
9851       \let\leftlinenum\rightlinenum%
9852       \let\leftlinenumR\rightlinenumR%
9853       \let\l@drd@ta\l@dld@ta%
9854       \let\l@drsn@te\l@dlsn@te%
9855     \fi%
9856     \hfill\l@drd@ta \csuse{LR}{\l@drsn@te}%
9857     \endgroup%
9858   \fi%
9859 }%
9860
9861 \def\print@leftmargin@eledsection{%
9862   \if@eled@sectioning%
9863     \leavevmode%
9864     \begingroup%
9865     \if@RTL%
9866       \let\rlap\llap%
9867       \let\rightlinenum\leftlinenum%
9868       \let\rightlinenumR\leftlinenumR%
9869       \let\l@dld@ta\l@drd@ta%
9870       \let\l@dlsn@te\l@drsn@te%
9871     \fi%
9872     \l@dld@ta\csuse{LR}{\l@dlsn@te}%
9873     \endgroup%
9874   \fi%
9875 }%
9876
9877 %

```

### XXXIV.5 Patching standard commands

`\M@ssect`  
`\@mem@old@ssect`  
`\@makechapterhead`  
`\@makechapterhead`  
`\@makeschapterhead`  
`\@ssect`  
`\@ssect`

We have to patch L<sup>A</sup>T<sub>E</sub>X, 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.

```

9878 \AtBeginDocument{%
9879
9880
9881 \pretocmd{\M@ssect}

```



```

9882 {\let\old@edtext=\edtext%
9883 \let\edtext=\dummy@edtext@showlemma%
9884 }
9885 {}
9886 {}
9887
9888 \apptocmd{\M@sect}
9889 {\let\edtext=\old@edtext}
9890 {}
9891 {}
9892
9893 \patchcmd{\M@sect}
9894 { #9}
9895 { #9%
9896 \print@rightmargin@eledsection%
9897 }
9898 {}
9899 {}
9900
9901 \patchcmd{\M@sect}
9902 {\hskip #3\relax}
9903 {\hskip #3\relax%
9904 \print@leftmargin@eledsection%
9905 }
9906 {}
9907 {}
9908
9909 \patchcmd{\@mem@old@ssect}
9910 {#5}
9911 {#5%
9912 \print@leftmargin@eledsection%
9913 }
9914 {}
9915 {}
9916
9917 \patchcmd{\@mem@old@ssect}
9918 {\hskip #1}
9919 {\hskip #1%
9920 \print@rightmargin@eledsection%
9921 }
9922 {}
9923 {}
9924
9925
9926
9927 \patchcmd{\scr@startchapter}{\if@openright\cleardoublepage\else\clearpage\fi}{%
9928 \if@eled@sectioning\else%
9929 \ifl@dprintingpages\else%
9930 \if@openright\cleardoublepage\else\clearpage\fi}%No clearpage inside a

```

```

\Pages: will keep critical notes from printing on the title page. Here for
scrbook.
9931   \fi%
9932   \fi%
9933   }
9934   {}
9935   {}
9936
9937 \patchcmd{\@makechapterhead}
9938   {#1}
9939   {\print@leftmargin@eledsection%
9940     #1%
9941     \print@rightmargin@eledsection%
9942   }
9943   {}
9944   {}
9945
9946 \patchcmd{\@makechapterhead}% For BIDI
9947   {\if@RTL\raggedleft\else\raggedright\fi}%
9948   {\if@eled@sectioning\else%
9949     \if@RTL\raggedleft\else\raggedright\fi%
9950   \fi%
9951   }%
9952   {}%
9953   {}%
9954
9955 \patchcmd{\@makeschapterhead}
9956   {#1}
9957   {\print@leftmargin@eledsection%
9958     #1%
9959     \print@rightmargin@eledsection%
9960   }
9961   {}
9962   {}
9963
9964 \pretocmd{\@sect}
9965   {\let\old@edtext=\edtext
9966     \let\edtext=\dummy@edtext@showlemma%
9967   }
9968   {}
9969   {}
9970
9971 \apptocmd{\@sect}
9972   {\let\edtext=\old@edtext}
9973   {}
9974   {}
9975
9976 \pretocmd{\@ssect}
9977   {\let\old@edtext=\edtext%
9978     \let\edtext=\dummy@edtext@showlemma%

```

```

9979 }
9980 {}
9981 {}
9982
9983 \apptocmd{\@ssect}
9984 {\let\edtext=\old@edtext}
9985 {}
9986 {}
9987
9988 %

```

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.

```

9989 \@ifpackageloaded{nameref}{
9990
9991   \patchcmd{\NR@sect}
9992     {#8}
9993     {#8%
9994       \print@rightmargin@eledsection%
9995     }
9996     {}
9997     {}
9998
9999   \patchcmd{\NR@sect}
10000     {\hskip #3\relax}
10001     {\hskip #3\relax%
10002       \print@leftmargin@eledsection%
10003     }
10004     {}
10005     {}
10006
10007   \patchcmd{\NR@ssect}
10008     {#5}
10009     {#5%
10010       \print@rightmargin@eledsection%
10011     }
10012     {}
10013     {}
10014
10015   \patchcmd{\NR@ssect}
10016     {\hskip #1}
10017     {\hskip #1%
10018       \print@leftmargin@eledsection%
10019     }
10020     {}
10021     {}
10022   }%
10023   {
10024     \patchcmd{\@sect}
10025     {#8}

```

```

10026     {#8%
10027     \print@rightmargin@eledsection%
10028     }
10029     {}
10030     {}
10031
10032     \patchcmd{\@sect}
10033     {\hskip #3\relax}
10034     {\hskip #3\relax%
10035     \print@leftmargin@eledsection%
10036     }
10037     {}
10038     {}
10039
10040     \patchcmd{\@ssect}
10041     {#5}
10042     {#5%
10043     \print@rightmargin@eledsection%
10044     }
10045     {}
10046     {}
10047
10048     \patchcmd{\@ssect}
10049     {\hskip #1}
10050     {\hskip #1%
10051     \print@leftmargin@eledsection%
10052     }
10053     {}
10054     {}
10055     }%
10056 }%
10057 %

```

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 `#`.

```

10058 {}}%
10059 \protect\catcode`\#=6 %Space NEEDS by \catcode
10060 %

```

**\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.

```

10061 \AtBeginDocument{%
10062 \patchcmd{\chapter}{\clearforchapter}{%
10063 \if@eled@sectioning\else%
10064 \ifl@dprintingpages\else%
10065 \clearforchapter%
10066 \fi%
10067 \fi%

```

```

10068 }%
10069 {}%
10070 {}%
10071
10072 \patchcmd{\chapter}{\if@openright\cleardoublepage\else\clearpage\fi}{%
10073   \if@eled@sectioning\else%
10074     \ifl@printingpages%
10075       \endgraf%
10076     \else%
10077       \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
10078   \fi%
10079 \fi%
10080 }%
10081 {}%
10082 {}%
10083 }%
10084 %

```

**\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.

```

10085 \newif\if@eled@sectioning%
10086 %

```

We reopen a new \notbool{@noeled@sec} statement, as we will define the \elesection commands.

```

10087 \notbool{@noeled@sec}{%
10088 %

```

## XXXIV.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.

```

10089 \newwrite\eled@sectioning@out
10090 %

```

**\eledchapter** And now, the user sectioning commands, which write to the file, and also add content as a “normal” line.

```

\eledsection
\eledsubsection
10091 \newcommand{\eledchapter}[2][]{%
\eledsubsubsection 10092   \disable@familiarnotes%
\eledchapter* 10093   #2%
\eledsection* 10094   \restore@familiarnotes%
\eledsubsection* 10095   \ifl@Rcol%
\eledsubsubsection*

```

```

10096 \immediate\write\eled@sectioningR@out{%
10097 \string\eled@chapter{#1}{\unexpanded{#2}}{\the\l@dumpstartsR}{R}
10098 }%
10099 \else%
10100 \immediate\write\eled@sectioningR@out{%
10101 \string\eled@chapter{#1}{\unexpanded{#2}}{\the\pstarts@read@L}{R}%
10102 }%
10103 \fi%
10104 }
10105
10106 \newcommand{\eledsection}[2][{}]{%
10107 \disable@familiarnotes%
10108 #2%
10109 \restore@familiarnotes%
10110 \ifledRcol%
10111 \immediate\write\eled@sectioningR@out{%
10112 \string\eled@section{#1}{\unexpanded{#2}}{\the\l@dumpstartsR}{R}
10113 }%
10114 \else%
10115 \immediate\write\eled@sectioningR@out{%
10116 \string\eled@section{#1}{\unexpanded{#2}}{\the\pstarts@read@L}{R}%
10117 }%
10118 \fi%
10119 }
10120
10121 \newcommand{\eledsubsection}[2][{}]{%
10122 \disable@familiarnotes%
10123 #2%
10124 \restore@familiarnotes%
10125 \ifledRcol%
10126 \immediate\write\eled@sectioningR@out{%
10127 \string\eled@subsection{#1}{\unexpanded{#2}}{\the\l@dumpstartsR}{R}
10128 }%
10129 \else%
10130 \immediate\write\eled@sectioningR@out{%
10131 \string\eled@subsection{#1}{\unexpanded{#2}}{\the\pstarts@read@L}{R}%
10132 }%
10133 \fi%
10134 }
10135 \newcommand{\eledsubsubsection}[2][{}]{%
10136 \disable@familiarnotes%
10137 #2%
10138 \restore@familiarnotes%
10139 \ifledRcol%
10140 \immediate\write\eled@sectioningR@out{%
10141 \string\eled@subsubsection{#1}{\unexpanded{#2}}{\the\l@dumpstartsR}
10142 }%

```

```

10143 \else%
10144 \immediate\write\eled@sectioning@out{%
10145 \string\eled@subsubsection{#1}{\unexpanded{#2}}{\the\pstarts@read@L
10146 }{}{}}%
10147 }%
10148 \fi%
10149 }
10150
10151 \WithSuffix\newcommand\eledchapter*[2][]{%
10152 \disable@familiarnotes%
10153 #2%
10154 \restore@familiarnotes%
10155 \ifledRcol%
10156 \immediate\write\eled@sectioningR@out{%
10157 \string\eled@chapter{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR}{*}{R}
10158 }%
10159 \else%
10160 \immediate\write\eled@sectioning@out{%
10161 \string\eled@chapter{#1}{\unexpanded{#2}}{\the\pstarts@read@L}{*}{}}%
10162 }%
10163 \fi%
10164 }
10165
10166 \WithSuffix\newcommand\eledsection*[2][]{%
10167 \disable@familiarnotes%
10168 #2%
10169 \restore@familiarnotes%
10170 \ifledRcol%
10171 \immediate\write\eled@sectioningR@out{%
10172 \string\eled@section{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR}{*}{R}
10173 }%
10174 \else%
10175 \immediate\write\eled@sectioning@out{%
10176 \string\eled@section{#1}{\unexpanded{#2}}{\the\pstarts@read@L}{*}{}}%
10177 }%
10178 \fi%
10179 }
10180
10181 \WithSuffix\newcommand\eledsubsection*[2][]{%
10182 \disable@familiarnotes%
10183 #2%
10184 \restore@familiarnotes%
10185 \ifledRcol%
10186 \immediate\write\eled@sectioningR@out{%
10187 \string\eled@subsection{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR}{*}{
10188 R}
10189 }%
10190 \else%
10191 \immediate\write\eled@sectioning@out{%

```

```

10191 \string\eled@subsection{#1}{\unexpanded{#2}}{\the\pstarts@read@L
10192 }{*}{}%
10193 }%
10194 \fi%
10195 }
10196 \WithSuffix\newcommand\eledsubsubsection*[2] [] {%
10197 \disable@familiarnotes%
10198 #2%
10199 \restore@familiarnotes%
10200 \ifledRcol%
10201 \immediate\write\eled@sectioningR@out{%
10202 \string\eled@subsubsection{#1}{\unexpanded{#2}}{\the\l@dumpstartsR
10203 }{*}{R}
10204 }%
10205 \else%
10206 \immediate\write\eled@sectioning@out{%
10207 \string\eled@subsubsection{#1}{\unexpanded{#2}}{\the\pstarts@read@L
10208 }{*}{}%
10209 }%
10210 \fi%
10211 }
10212 %

```

### XXXIV.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  $\text{\LaTeX}$  sectioning command.
2. Mandatory arguments of  $\text{\LaTeX}$  sectioning command.
3. Pstart number.
4. Side: R if right, nothing if left.
5. Starred or not.

```

10211 \def\eled@chapter#1#2#3#4#5{%
10212 \ifstrempy{#4}%
10213 {%
10214 \ifstrempy{#1}%
10215 {%
10216 \csgdef{eled@sectioning@#3#5}{\let\edtext=\dummy@edtext@showlemma\
chapter{#2}}%
10217 \csgdef{eled@sectmark@#3#5}{\let\edtext=\dummy@edtext}\chaptermark
10218 {#2}}%
10219 }%Need for \pairs, because of using parbox.
10220 }%

```



```

10220 \csgdef{eled@sectioning@#3#5}{\let\edtext=\dummy@edtext@showlemma\
chapter[#1]{#2}}%
10221 \csgdef{eled@sectmark@#3#5}{\let\edtext=\dummy@edtext{}\chaptermark
{#2}}%Need for \pairs, because of using parbox.
10222 }%
10223 }%
10224 {%
10225 \ifstrempy{#1}%
10226 {\csgdef{eled@sectioning@#3#5}{\let\edtext=\dummy@edtext@showlemma\
chapter*{#2}}}%
10227 {\csgdef{eled@sectioning@#3#5}{\let\edtext=\dummy@edtext@showlemma\
chapter*{#1}{#2}}}%Bug in LaTeX!
10228 }%
10229 \listcsgadd{eled@sections#5@@}{#3}%
10230 }
10231 \def\eled@section#1#2#3#4#5{%
10232 \ifstrempy{#4}%
10233 {\ifstrempy{#1}%
10234 {%
10235 \csgdef{eled@sectioning@#3#5}{\section{#2}}%
10236 \csgdef{eled@sectmark@#3#5}{\let\edtext=\dummy@edtext{}\sectionmark
{#2}}%Need for \pairs, because of using parbox.
10237 }%
10238 {%
10239 \csgdef{eled@sectioning@#3#5}{\section[#1]{#2}}%
10240 \csgdef{eled@sectmark@#3#5}{\let\edtext=\dummy@edtext{}\sectionmark
{#1}}%Need for \pairs, because of using parbox.
10241 }%
10242 }%
10243 {\ifstrempy{#1}%
10244 {\csgdef{eled@sectioning@#3#5}{\section*{#2}}}%
10245 {\csgdef{eled@sectioning@#3#5}{\section*{#1}{#2}}}%Bug in LaTeX!
10246 }
10247 \listcsgadd{eled@sections#5@@}{#3}%
10248 }
10249 \def\eled@subsection#1#2#3#4#5{%
10250 \ifstrempy{#4}%
10251 {\ifstrempy{#1}%
10252 {%
10253 \csgdef{eled@sectioning@#3#5}{\subsection{#2}}%
10254 \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)
10255 }%
10256 {%
10257 \csgdef{eled@sectioning@#3#5}{\subsection[#1]{#2}}%
10258 \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)
10259 }%

```

```

10260 }%
10261 {\ifstrempy{#1}%
10262   {\csgdef{eled@sectioning@#3#5}{\subsection*{#2}}}%
10263   {\csgdef{eled@sectioning@#3#5}{\subsection*[#1]{#2}}}%Bug in LaTeX!
10264 }
10265 \listcsgadd{eled@sections#5@@}{#3}%
10266 }
10267 \def\eled@subsubsection#1#2#3#4#5{%
10268   \ifstrempy{#4}%
10269     {\ifstrempy{#1}%
10270       {\csgdef{eled@sectioning@#3#5}{\subsubsection{#2}}}%
10271       {\csgdef{eled@sectioning@#3#5}{\subsubsection[#1]{#2}}}%
10272     }%
10273     {\ifstrempy{#1}%
10274       {\csgdef{eled@sectioning@#3#5}{\subsubsection*{#2}}}%
10275       {\csgdef{eled@sectioning@#3#5}{\subsubsection*[#1]{#2}}}%Bug in
LaTeX!
10276     }
10277     \listcsgadd{eled@sections#5@@}{#3}%
10278   }
10279
10280 %

```

End of the conditional test about noeledsec option.

```

10281 }{}
10282 %

```

## XXXV 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.

```

10283 \def\normal@page@break{}
10284 %

```

**\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.

```

10285 \def\l@prev@pb{}
10286 \def\l@prev@nopb{}
10287 %

```

**\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.

```

10288 \newcommand{\ledpb}{\write\linenum@out{\string\led@pb}}
10289 \newcommand{\ledpbnum}[1]{\write\linenum@out{\string\led@pbnum{#1}}}
10290 \newcommand{\lednopb}{\write\linenum@out{\string\led@nopb}}
10291 \newcommand{\lednopbnum}[1]{\write\linenum@out{\string\led@nopbnum{#1}}}
10292 %

```

**\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.

```

10293 \newcommand{\led@pb}{\listxadd{\l@prev@pb}{\the\absline@num}}
10294 \newcommand{\led@pbnum}[1]{\listxadd{\l@prev@pb}{#1}}
10295 \newcommand{\led@nopb}{\listxadd{\l@prev@nopb}{\the\absline@num}}
10296 \newcommand{\led@nopbnum}[1]{\listxadd{\l@prev@nopb}{#1}}
10297 %

```

**\ledpbsetting** The `\ledpbsetting` macro only changes the value of `\led@pb@macro`, for which the default value is `before`.

```

10298 \def\led@pb@setting{before}
10299 \newcommand{\ledpbsetting}[1]{\gdef\led@pb@setting{#1}}
10300 %

```

**\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`.

```

10301 \newcommand{\led@check@pb}{\xifinlist{\the\absline@num}{\l@prev@pb}{\pagebreak[4]}}
10302 \newcommand{\led@check@nopb}{%
10303   \IfStrEq{\led@pb@setting}{before}{%
10304     \xifinlist{\the\absline@num}{\l@prev@nopb}{%
10305       {\numdef{\abs@prevline}{\the\absline@num-1}}%
10306       \xifinlist{\abs@prevline}{\normal@page@break}{%
10307         {\nopagebreak[4]\enlargethispage{\baselineskip}}%
10308       }%
10309     }%
10310   }%
10311 }%
10312 \IfStrEq{\led@pb@setting}{after}{%

```

```

10313 \xifinlist{\the\absline@num}{\l@prev@nopb}{%
10314 \xifinlist{\the\absline@num}{\normal@page@break}%
10315 {\nopagebreak[4]\enlargethispage{\baselineskip}}%
10316 {}%
10317 }%
10318 {}%
10319 {}%
10320 {}%
10321 }
10322 %

```

## XXXVI 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.

```

10323 \newcommand{\check@pb@in@verse}{%
10324 \ifinstanza\iflednopbinverse\ifinserthangingsymbol% Using stanzas and
enabling page breaks in verse control, while on a hanging verse.
10325 \ifnum\page@num=\last@page@num\else%If we have change page
10326 \IfStrEq{\led@pb@setting}{before}{%
10327 \numgdef{\abs@line@verse}{\the\absline@num-1}%
10328 \ledpbnum{\abs@line@verse}%
10329 }{}%
10330 \IfStrEq{\led@pb@setting}{after}{%
10331 \numgdef{\abs@line@verse}{\the\absline@num-1}%
10332 \lednopbnum{\abs@line@verse}%
10333 }{}%
10334 \fi%
10335 \fi\fi\fi%
10336 }
10337 %

```

## XXXVII 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>36</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>37</sup>

```

10338 \def\Hy@raisedlink@left#1{%
10339     \ifvmode
10340         #1%
10341     \else
10342         \Hy@SaveSpaceFactor
10343         \llap{\smash{%
10344             \begingroup
10345                 \let\HyperRaiseLinkLength\@tempdima
10346                 \setlength\HyperRaiseLinkLength\HyperRaiseLinkDefault
10347                 \HyperRaiseLinkHook
10348             \expandafter\endgroup
10349             \expandafter\raise\the\HyperRaiseLinkLength\hbox{%
10350                 \Hy@RestoreSpaceFactor
10351                 #1%
10352                 \Hy@SaveSpaceFactor
10353             }%
10354         }}%
10355         \Hy@RestoreSpaceFactor
10356         \penalty\@M\hskip\z@\relax
10357     \fi
10358 }
10359 %

```

## XXXVIII Compatibility with eledmac

Here, we define some commands for the `eledmac-compat` option.

```

10360 \ifeledmaccompat%
10361
10362 \newcommand{\footnormalX}[1]{\arrangementX[#1]{normal}}%
10363 \newcommand{\footparagraphX}[1]{\arrangementX[#1]{paragraph}}%
10364 \newcommand{\foottwocolX}[1]{\arrangementX[#1]{twocol}}%
10365 \newcommand{\footthreecolX}[1]{\XarrangementX[#1]{threecol}}%
10366
10367 \unless\ifnocritical@
10368     \newcommand{\footnormal}[1]{\Xarrangement[#1]{normal}}%
10369     \newcommand{\footparagraph}[1]{\Xarrangement[#1]{paragraph}}%

```

<sup>36</sup><http://tex.stackexchange.com/a/17138/7712>.

<sup>37</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-coupeure-de-ligne/801>.

```

10370 \newcommand{\foottwocol}[1]{\Xarrangement[#1]{twocol}}%
10371 \newcommand{\footthreecol}[1]{\Xarrangement[#1]{threecol}}%
10372 \let\hsizetwocol\Xhsizetwocol
10373 \let\hsizethreecol\Xhsizethreecol
10374 \let\bhookXnote\Xbhooknote
10375 \let\boxsymlinenum\Xboxsymlinenum
10376 \let\symlinenum\Xsymlinenum
10377 \let\beforenumberinfootnote\Xbeforenumber
10378 \let\afternumberinfootnote\Xafternumber
10379 \let\beforeXsymlinenum\Xbeforesymlinenum
10380 \let\afterXsymlinenum\Xaftersymlinenum
10381 \let\inplaceofnumber\Xinplaceofnumber
10382 \let\Xlemmaseparator\lemmaseparator
10383 \let\afterlemmaseparator\Xafterlemmaseparator
10384 \let\beforelemmaseparator\Xbeforelemmaseparator
10385 \let\inplaceoflemmaseparator\Xinplaceoflemmaseparator
10386 \let\txbeforeXnotes\Xtxbeforenotes
10387 \let\afterXrule\Xafterrule
10388 \let\numberonlyfirstinline\Xnumberonlyfirstinline
10389 \let\numberonlyfirstintwlines\Xnumberonlyfirstintwlines
10390 \let\nonumberinfootnote\Xnonumberinfootnote
10391 \let\pstartinfootnote\Xpstart
10392 \let\pstartinfootnoteeverytime\Xpstarteverytime
10393 \let\onlyXpstart\Xonlypstart
10394 \let\Xnonumberinfootnote\Xnonumber
10395 \let\nonbreakableafternumber\Xnonbreakableafternumber
10396 \let\maxhXnotes\Xmaxhnotes
10397 \let\beforeXnotes\Xbeforenotes
10398 \let\boxlinenum\Xboxlinenum
10399 \let\boxlinenumalign\Xboxlinenumalign
10400 \let\boxstartlinenum\Xboxstartlinenum
10401 \let\boxendlinenum\Xboxendlinenum
10402 \let\twlines\Xtwlines
10403 \let\morethantwlines\Xmorethantwlines
10404 \let\twlinesbutnotmore\Xtwlinesbutnotmore
10405 \let\twlinesonlyinsamepage\Xtwlinesonlyinsamepage
10406 \fi
10407
10408 \unless\ifnofamiliar@
10409   \let\notesXwidthliketwocolumns\noteswidthliketwocolumnsX
10410 \fi
10411 \newcommandx{\parafootsep}[2][1,usedefault]{%
10412   \Xparafootsep[#1]{#2}%
10413   \parafootsepX[#1]{#2}
10414 }%
10415
10416 \newcommandx{\afternote}[2][1,usedefault]{%
10417   \Xafternote[#1]{#2}%
10418   \afternoteX[#1]{#2}%
10419 }%

```

```

10420
10421 \unless\ifnoend@
10422   \let\XendXtwolines\Xendtwolines
10423   \let\XendXmorethantwolines\Xendmorethantwolines
10424   \let\hookXendnote\Xendhooknote
10425   \let\boxXendlinenum\Xendboxlinenum%
10426   \let\boxXendlinenumalign\Xendboxlinenumalign%
10427   \let\boxXendstartlinenum\Xendboxstartlinenum%
10428   \let\boxXendendlinenum\Xendboxendlinenum%
10429   \let\XendXlemmaseparator\Xendlemmaseparator
10430   \let\XendXbeforelemmaseparator\Xendbeforelemmaseparator
10431   \let\XendXafterlemmaseparator\Xendafterlemmaseparator
10432   \let\XendXinplaceoflemmaseparator\Xendinplaceoflemmaseparator
10433 \fi
10434
10435 \AtBeginDocument{%
10436   \ifdef\lineref{}\let\lineref\edlineref}%
10437 }%
10438
10439
10440 \fi%
10441 %

```

</code>

## Appendix A Things to do when changing versions

### 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>38</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:

<code>I saw my friend \critext{Smith}</code>	1 I saw my friend
<code>\Afootnote{Jones C, D.}/</code>	2 Smith on Tuesday.
on Tuesday.	<u>2 Smith]</u> Jones C, D.

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

The `⟨lemma⟩` may contain further `\critext` commands. Nesting makes it possible to print an explanatory note on a long passage together with notes on variants for individual words within the passage. For example:

<code>\critext{I saw my friend</code>	1 I saw my friend
<code>\critext{Smith}{\Afootnote{Jones</code>	2 Smith on Tuesday.
<code>C, D.}/ on Tuesday.}</code>	<u>2 Smith]</u> Jones C, D.
<code>\Bfootnote{The date was</code>	<u>1–2 I saw my friend</u>
<code>July 16, 1954.}</code>	Smith on Tuesday.] The
<code>/</code>	date was July 16, 1954.

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

The second argument of the `\critext` macro, `⟨commands⟩`, is the same as the second argument to the `\edtext` macro.

It is possible to define aliases for `\critext`, which can be easier to type. You can make a single character substitute for `\critext` by saying this:

```
\catcode'\<=\active
```

<sup>38</sup>A name like `\text` is likely to be defined by other  $\text{\LaTeX}$  packages (it certainly is by the AMS packages) and it seems sensible to try and avoid clashes with other definitions.



```
\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. 138 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.

## 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. 41), 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. 43) or `\Xfootnote` options (6.2.2 p. 30). Otherwise please add a new ticket on Github to request a new function for doing this.<sup>39</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>39</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.

### A.3 Migration to eledmac 1.5.1

The version 1.5.1 corrects a bug in `stanzaindentrepetition` (cf. 9.3 p. 58). This bug had two consequences:

1. `stanzaindentrepetition` did not work when its value was greater than 2.
2. `stanzaindentrepetition` worked wrong when its value was equal to 2.

So, if you used `stanzaindentrepetition` with a value equal to 2, you had to change your `\setstanzaindent`. Explanation:

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

to:

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

### 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. 19). 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. 76.

## 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 every footnote, whatever the section, without having to use `\numberpstarttrue`, you can use `\Xpstarteverytime`.

## A.6 Migration to *eledmac* 1.21.0

### 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.

### 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.

## 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`.

## A.8 Migration to *eledmac* 1.23.0

You must delete the numbered auxiliary files before compiling with the new version of *eledmac*.

## A.9 Migration from eledmac to reledmac

There are many changes in reledmac which require the user to make modifications.

### 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. 79 in order to know how to limit it.

### 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.

### 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.

#### 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>\sidenotesep</code>	<code>\setsidenotsep</code>
<code>\startstanzahook</code>	Optional argument of <code>\stanza</code>
<code>\symplinenum</code>	<code>\Xsymplinenum</code>

#### 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>

### A.9.6 Endnotes

With *reledmac*, there is now one auxiliary file for each endnotes set (.Aend, .Bend, .Cend etc.). If you have overridden `\doendnotes` (which you should not have done) you must adapt your code.

### 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.

### 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.

## 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` to the argument of `\Xbhooknote/\bhooknoteX`.

## 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`.

## 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 the optional argument of `\stanza` or `\endverse`.

### 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` (7.2.5 p. 46).

### A.14 Migration to reledmac 2.5.0

It is strongly recommended to stop redefining `\printnpnum` and to use the hooks documented in 7.2.4 p. 46.

`\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.

### 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.

### 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.

### 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 `\Xparindent` and `\parindentX`.

### 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.

### 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.

### A.20 Migration to reledmac 2.21.0

Previously, there was a bug, which meant that the description in the handbook was incorrect. If you wrote

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

“elephant” was indexed in the main text and in the critical footnotes. With the new version of reledmac, it is indexed only in main text. If you also want to index it in critical footnotes, do

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

### A.21 Migration to reledmac 2.24.0

When using `\labelpstarttrue`, a spurious space<sup>a</sup> was introduced after the pstart number (only for normal typesetting, not while typesetting in parallel). The new version of the package has deleted this spurious space. If you consider that it was NOT a spurious space, you should add it manually in your definition of `\thepstart`.

### A.22 Migration to reledmac 2.26.0

You must delete your `.aux` file after having upgraded to this new version of reledmac.

### A.23 Migration to reledmac 2.27.1

This release fixes spurious space in `\hidenumbering`. If you considered this spurious space as normal, you must insert it manually using.

```
xx\hidenumbering\ xx
```

To get the space after `xx`.

### A.24 Migration to reledmac 2.30.0

If you have multiple annotations for the same line, these are now separated with a comma in the margin. You can use `\setlinenumannotationsep` to change the separator.

### A.25 Migration to reledmac 2.31.1

If you use `\Xgroupbyline`, the indentation of the notes has been deleted, in order to have the same behavior as for normal critical notes.

Use `\Xparindent` to restore indentation.

### A.26 Migration to reledmac 2.32.7

For paragraphed familiar footnotes, the separator between notes was `\Xparafootsep` instead of `\parafootsepX`. This release fixes this bug. You may adapt your settings.

### A.27 Migration to reledmac 2.33.0

You should not redefine anymore `\ledinnotemark`, but use instead `\innotemarkX` and `\Xinnotemark`.

If you have personal script which parses for `\ledinnote` or if you redefine `\ledinnote` (but normally, you should not), this macro have new arguments. Look at the code for more details.

## Appendix B Auxiliary softwares

This appendix lists some software and tools related to *reledmac* that may be useful.

### B.1 *samewords*

The *samewords* program (Michael Stenskjær Christensen) automatically adds `\sameword` commands (6.3 p. 33) into a `.tex` file.

<https://samewords.readthedocs.io/en/latest/>

### B.2 critical-keys for *Emacs*

The *Emacs* editor has a package to help inserting *reledmac*'s commands (Juan Manuel Macías Chaín).

<https://gitlab.com/maciaschain/critical-keys>

### B.3 critical-marks for *Emacs*

A set of functions for *Emacs* that run in the export process from Org Mode to LaTeX and replace a series of simple textual marks with the basic *reledmac*'s commands (Juan Manuel Macías Chaín) <https://gitlab.com/maciaschain/critical-marks>

### B.4 Import from TEI

### B.5 Import from TEI

There are multiple tools to convert from XML-TEI to *reledmac*. Here is a non exhaustive-list:

- <http://ciham-digital.huma-num.fr/teicat/>
- [https://github.com/fizzbucket/tei\\_transformer](https://github.com/fizzbucket/tei_transformer)
- <https://github.com/TEIC/Stylesheets>
- <http://lombardpress.org/print/>
- <https://github.com/Jean-Baptiste-Camps/TEItoLaTeX>

Please, tell us if you know of other tools.

## References

- [Bre96] Herbert Breger. `tabmac`. October 1996. (Available from CTAN in `macros/plain/contrib/tabmac`)
- [Bur01] John Burt. ‘Typesetting critical editions of poetry’. *TUGboat*, **22**, 4, pp 353–361, December 2001. (Code available from CTAN in `macros/latex/contrib/poemscol`)
- [Eck03] Matthias Eckermann. *The Parallel-Package*. April 2003. (Available from CTAN in `macros/latex/contrib/parallel`)
- [Fai03] Robin Fairbairns. *footmisc—a portmanteau package for customising footnotes in L<sup>A</sup>T<sub>E</sub>X*. February 2003. (Available from CTAN in `macros/latex/contrib/footmisc`)
- [LW90] John Lavagnino and Dominik Wujastyk. ‘An overview of edmac: a PLAIN TeX format for critical editions’. *TUGboat*, **11**, 4, pp. 623–643, November 1990. (Code available from CTAN in `macros/plain/contrib/edmac`)
- [Lüc03] Uwe Lück. ‘ednotes—critical edition typesetting with LaTeX’. *TUGboat*, **24**, 2, pp. 224–236, 2003. (Code available from CTAN in `macros/latex/contrib/ednotes`)
- [Sul92] Wayne G. Sullivan. *The file edstanza.doc*. June 1992. (Available from CTAN in `macros/plain/contrib/edmac`)
- [Wil02] Peter Wilson. *The memoir class for configurable typesetting*. November 2002. (Available from CTAN in `macros/latex/contrib/memoir`)
- [Wil04] Peter Wilson and Maïeul Rouquette. *Parallel typesetting for critical editions: the eledpar package*. December 2004. (Available from CTAN in `macros/latex/contrib/ledmmac`)

## Index

### Symbols

<code>\&amp;</code> .....	57
<code>\@EDROWFILL@</code> .....	1
<code>\@adv</code> .....	1
<code>\@advancestanzanumber</code> .....	1
<code>\@annot</code> .....	1
<code>\@annot@end@print</code> .....	1
<code>\@annot@start@print</code> .....	1
<code>\@beforeinsertofthisedtext</code> .....	1
<code>\@check@edtext@args</code> .....	1
<code>\@docclearpage</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
<code>\@emptytoks</code>	1
<code>\@fnpos</code>	1
<code>\@footnotemark</code>	1
<code>\@footnotetext</code>	1
<code>\@getfirstseries</code>	1
<code>\@gobblefour</code>	1
<code>\@gobbleseven</code>	1
<code>\@gobblethree</code>	1
<code>\@h</code>	1
<code>\@changingsymbol</code>	1
<code>\@iiiminipage</code>	1
<code>\@insertstanzanumber</code>	1
<code>\@insidethisline</code>	1
<code>\@k</code>	1
<code>\@l@dttempcnta</code>	1
<code>\@l@dttempcntb</code>	1
<code>\@lab</code>	1
<code>\@led@testifnofoot</code>	1
<code>\@lemma</code>	1
<code>\@line@#num</code>	1
<code>\@linenumannotationsep</code>	1
<code>\@lock</code>	1
<code>\@lopL</code>	1
<code>\@lopR</code>	1
<code>\@makechapterhead</code>	1
<code>\@makeschapterhead</code>	1
<code>\@mem@extranofeet</code>	1
<code>\@mem@old@ssect</code>	1
<code>\@mpfnpos</code>	1
<code>\@msd</code>	1
<code>\@msd@c</code>	1
<code>\@msd@options@iffullpage</code>	1
<code>\@msdata@list</code>	1
<code>\@nl</code>	1
<code>\@nl@reg</code>	1
<code>\@opXfeet</code>	1
<code>\@outputpage</code>	1
<code>\@page</code>	1
<code>\@pend</code>	1
<code>\@pendR</code>	1
<code>\@ref</code>	1
<code>\@ref@reg</code>	1
<code>\@ref@reg@parse</code>	1
<code>\@sect</code>	1
<code>\@series</code>	1
<code>\@set</code>	1

\@sidenoteseop	1
\@ssect	1
\@startstanza	1
\@stopmsd	1
\@stopstanza	1
\@sw	1
\@tag	1
\@wredindex	1
\@xloop	1
\@xympar	1
CLASSarticle	79
CLASSbook	79, 384
CLASSmemoir	226, 292–295, 331, 384, 404, 472, 476
CLASSscrbook	476
COMMAND\*footnote	80
COMMAND\...\@footnotemark...	229
COMMAND\...d@ta	167
COMMAND\<hook	
@<series	279
COMMAND\<hookname	
<pseudoserries	281
COMMAND\<type	
footfmt	212
COMMAND\@@line	202
COMMAND\@MM	188, 473
COMMAND\@Rlineflag	333, 473
COMMAND\@Serefprefix	310
COMMAND\@Serefprefixmore	310
COMMAND\@add@	374
COMMAND\@adv	121
COMMAND\@annot	173
COMMAND\@annot@end@print	216
COMMAND\@annot@start@print	216, 220, 260
COMMAND\@apprefprefixmore	310
COMMAND\@apprefprefixsingle	310
COMMAND\@beforeinsertofthisedtext	145
COMMAND\@bsphack	298
COMMAND\@docclearpage	293, 294, 466, 476
COMMAND\@doreinfeetX	476
COMMAND\@dprintingcolumns	473
COMMAND\@edindex@hyperref	333–335
COMMAND\@edtext@level	142
COMMAND\@esphack	298
COMMAND\@firstofone	38
COMMAND\@fnpos	248, 289
COMMAND\@footnotemark	226, 466, 476
COMMAND\@footnotetext	226, 227, 466
COMMAND\@gobble	27, 37, 140, 141, 173, 273
COMMAND\@gobblefive	475
COMMAND\@gobblefour	472

COMMAND\@gobbleseven	275
COMMAND\@gobblethree	465
COMMAND\@h	205
COMMAND\@hangingsymbol	339
COMMAND\@iiiminipage	322, 324, 465, 476
COMMAND\@iiiminpage	322
COMMAND\@l	470
COMMAND\@l@tempcnta	170, 175, 182
COMMAND\@l@tempcntb	175
COMMAND\@l@reg	470
COMMAND\@lab	117, 297, 301, 305, 465
COMMAND\@ldunboxmpfoot	325
COMMAND\@led@extranofeet	405
COMMAND\@ledinnote@command	329, 330
COMMAND\@lemma	145, 147
COMMAND\@lock	111, 340
COMMAND\@lopL	466
COMMAND\@lopR	466
COMMAND\@makecol	289, 290, 292, 293, 476
COMMAND\@mempnum	295
COMMAND\@mpfnpos	248
COMMAND\@msd	350, 351
COMMAND\@msd@c	350
COMMAND\@msd@options@iffullpage	357
COMMAND\@msdata@list	350, 352
COMMAND\@nl	117–122, 132, 174, 301, 465, 466
COMMAND\@nl@reg	118, 402, 466, 470
COMMAND\@opXfeet	466
COMMAND\@opfeetX	476
COMMAND\@opxtrafeeti	476
COMMAND\@outputpage	295, 296
COMMAND\@page	118, 119, 301
COMMAND\@pend	466
COMMAND\@pendR	466
COMMAND\@ref	117, 128, 129, 133, 134, 140
COMMAND\@ref@later	129, 135
COMMAND\@ref@reg	128, 173, 466
COMMAND\@ref@reg@parsearg	129
COMMAND\@reinserts	289, 291, 292, 476
COMMAND\@secondoftwo	81
COMMAND\@sect	387
COMMAND\@series	277
COMMAND\@set	122
COMMAND\@sidenotesep	321
COMMAND\@stopmsd	351
COMMAND\@sw	129, 149, 152, 153
COMMAND\@tag	141, 143, 147
COMMAND\@tempcnta	95
COMMAND\@tempcntb	95
COMMAND\@this@crossref@start	225

COMMAND\@toksa	103
COMMAND\@toksb	103
COMMAND\@xloop	184
COMMAND\@xympar	314, 476
COMMAND\Aendnote	17, 30
COMMAND\Afootfmt	188
COMMAND\Afootgroup	188
COMMAND\Afootnote	9, 17, 29, 30, 33, 143, 197, 225, 249, 267, 475
COMMAND\Afootstart	188
COMMAND\AtBeginDocument	293
COMMAND\AtEndEveryPend	20, 482
COMMAND\AtEveryPend	19, 20, 61, 160, 472, 474, 476, 482
COMMAND\AtEveryPend*	20
COMMAND\AtEveryPstart	19, 20, 61, 472, 474, 476, 479, 482
COMMAND\AtEveryPstart*	20
COMMAND\AtEveryStanza	60, 480, 482
COMMAND\AtEveryStopStanza	61, 480, 482
COMMAND\AtStartEveryPstart	20, 482
COMMAND\AtStartEveryStanza	60, 482
COMMAND\BeforeEveryStopStanza	61, 482
COMMAND\Bendnote	17, 29
COMMAND\Bfootnote	9, 17, 225, 249, 267
COMMAND\Centering	53
COMMAND\Cfootnote	225
COMMAND\Columns	96, 156, 193
COMMAND\Dfootnote	225
COMMAND\Efootnote	225
COMMAND\Gls	72
COMMAND\Hy@raisedlink	397
COMMAND\Hy@raisedlink@left	397
COMMAND\LTR	53
COMMAND\NR	387
COMMAND\Pages	96, 156, 290, 291, 295
COMMAND\ProcessOptionsX	85
COMMAND\RL	52
COMMAND\RaggedLeft	53
COMMAND\RaggedRight	53
COMMAND\RenewExpandableDocumentCommand	38, 151
COMMAND\SEonlypage	307, 478
COMMAND\SEref	65–68, 307, 310, 479, 481, 485
COMMAND\SErefonlypage	65–68, 408, 478
COMMAND\SErefwithpage	65, 67, 68, 307, 310, 408, 478, 480, 486
COMMAND\Stanza	471
COMMAND\Waklam	375
COMMAND\X@doreinfeet	291, 476
COMMAND\XR@prefix	314
COMMAND\XR@test	314
COMMAND\XR@test@mac	314
COMMAND\XR@test@mac@test	314
COMMAND\XXXXXXfmt	401



COMMAND\XXXXXfmt	401
COMMAND\Xafterenumber	48
COMMAND\Xafterlemmaseparator	50, 405
COMMAND\Xafternote	53, 54, 404, 405
COMMAND\Xafternumber	48, 405
COMMAND\Xafterrule	55, 249, 405, 471, 474
COMMAND\Xaftersymlinenum	48, 405
COMMAND\Xarrangement	43, 54, 80, 189, 190, 279, 405
COMMAND\Xarrangement@footparagraph	195
COMMAND\Xarrangement@normal	190
COMMAND\Xarrangement@paragraph	195
COMMAND\Xbeforeinserting	52, 53
COMMAND\Xbeforelemmaseparator	50, 405
COMMAND\Xbeforenotes	55, 248, 405, 471, 474
COMMAND\Xbeforenumber	48, 405
COMMAND\Xbeforesymlinenum	48, 405
COMMAND\Xbhookendnote	406
COMMAND\Xbhookgroup	55, 478, 479
COMMAND\Xbhooknote	52, 406, 407, 476, 477
COMMAND\Xboxendlinum	49, 406, 475
COMMAND\Xboxlinenum	48, 49, 406
COMMAND\Xboxlinenumalign	49, 406, 475
COMMAND\Xboxstartlinenum	49, 406, 475
COMMAND\Xboxsymlinenum	49, 406
COMMAND\Xcolalign	53, 474
COMMAND\Xdo@feet	476, 481
COMMAND\Xemdnoidenticallinenumannotation	67
COMMAND\Xend	275
COMMAND\XendXafterlemmaseparator	406
COMMAND\XendXbeforelemmaseparator	406
COMMAND\XendXinplaceoflemmaseparator	406
COMMAND\XendXlemmaseparator	406
COMMAND\XendXmorethantwolines	406
COMMAND\XendXtwolines	406
COMMAND\Xendafterenumber	48, 477
COMMAND\Xendafterlemmaseparator	50, 406
COMMAND\Xendafternote	56, 479
COMMAND\Xendafternumber	49
COMMAND\Xendafterpagenumbe	480
COMMAND\Xendafterpagenumber	46, 67
COMMAND\XendafterpagenumberSErefonlypage	408
COMMAND\Xendaftersymlinenum	48, 50, 477
COMMAND\Xendahookinplaceofnumber	50, 477
COMMAND\Xendahooklinenumber	49, 477
COMMAND\Xendbeforelemmaseparator	50, 406
COMMAND\Xendbeforelinenumber	49
COMMAND\Xendbeforenumber	48, 477
COMMAND\Xendbeforepagenumber	46, 66, 67
COMMAND\XendbeforepagenumberSErefonlypage	66
COMMAND\Xendbeforesymlinenum	48, 50, 477

COMMAND\Xendbhookinplaceofnumber	50, 477
COMMAND\Xendbhooklinenumber	49, 477
COMMAND\Xendbhooknote	52
COMMAND\Xendboxendlinenum	49, 475
COMMAND\Xendboxlinenum	49, 406, 473
COMMAND\Xendboxlinenumalign	49, 406, 475
COMMAND\Xendboxstartlinenum	49, 475
COMMAND\Xendboxsymlinenum	49, 477
COMMAND\Xendhangindent	52, 477, 479
COMMAND\Xendinplaceoflemmaseparator	31, 50, 406
COMMAND\Xendinplaceofnumber	48, 476
COMMAND\Xendinplaceofpagenumber	45, 481
COMMAND\Xendinsertsep@	256
COMMAND\Xendlemmadisablefontselection	51
COMMAND\Xendlemmafont	51, 478
COMMAND\Xendlemmaseparator	31, 50, 406
COMMAND\Xendletboxendlinenum	406
COMMAND\Xendlineflag	67
COMMAND\Xendlinenumannotationonlyfirst	45, 260, 485
COMMAND\Xendlinenumannotationonlyfirstintwo	45, 260, 485
COMMAND\Xendlinenumannotationposition	27, 67
COMMAND\Xendlineprefixmore	46, 67
COMMAND\Xendlineprefixsingle	46, 67
COMMAND\Xendlinerangeseparator	46, 67, 210, 408, 477
COMMAND\Xendmorethantwolines	31, 47, 67, 406, 474, 475
COMMAND\Xendnoidenticallinenumannotation	28, 259, 484
COMMAND\Xendnonumber	47, 476
COMMAND\Xendnote	252, 274, 275, 474
COMMAND\Xendnotefontsize	51, 405
COMMAND\Xendnotenumfont	49–51, 405
COMMAND\Xendnotes	255
COMMAND\Xendnumberonlyfirstinline	44, 477
COMMAND\Xendnumberonlyfirstintwolines	44, 477
COMMAND\Xendpagenumberonlyfirst	45, 481
COMMAND\Xendpagenumberonlyfirstifsingle	45, 481
COMMAND\Xendpagenumberonlyfirstintwo	45, 481
COMMAND\Xendparagraph	56, 471
COMMAND\Xendsep	56, 485
COMMAND\Xendstorelineinfo	253
COMMAND\Xendsublinesep	48, 67, 210
COMMAND\Xendsymlinenum	44, 477, 485
COMMAND\Xendsymlinenumannotation	45, 485
COMMAND\Xendsympagenum	45, 481
COMMAND\Xendtwolines	31, 46, 47, 67, 406, 474, 475
COMMAND\Xendtwolinesbutnotmore	47, 67, 474, 475
COMMAND\Xendtwolinesonlyinsamepage	47, 67, 474, 475
COMMAND\Xendtxtbeforenotes	54, 483
COMMAND\Xendwrapcontent	52, 480
COMMAND\Xendwraplemma	52, 480
COMMAND\Xendwraplinenumannotation	27, 67

COMMAND\Xfootfmt	407
COMMAND\Xfootgroup	194
COMMAND\Xfootins	193
COMMAND\Xfootnote	64, 70, 141, 401, 468, 472–474, 478, 480
COMMAND\Xfootstarts	194
COMMAND\Xgroupbyline	54, 181, 224, 287, 410, 483–485
COMMAND\Xgroupbylines	481
COMMAND\Xgroupbylineseparetwolines	54
COMMAND\Xhangindent	52, 477, 485
COMMAND\Xhsize	408, 478, 479
COMMAND\Xsizethreecol	53, 56, 406
COMMAND\Xsizetwocol	53, 56, 281, 406
COMMAND\Xinnotemark	70, 410, 486
COMMAND\Xinplaceoflemmaseparator	30, 50, 406
COMMAND\Xinplaceofnumber	30, 48, 406, 473, 475
COMMAND\Xinsertparafootsep	200, 201
COMMAND\Xledsetnormalparstuff	403, 404, 474
COMMAND\Xlemmadisablefontselection	51
COMMAND\Xlemmafont	51, 478
COMMAND\Xlemmaseparator	50, 210, 282, 285, 289, 406
COMMAND\Xlineflag	66, 408, 409, 480
COMMAND\Xlinenumannotationonlyfirst	44, 45, 220, 485
COMMAND\Xlinenumannotationonlyfirstintwo	45, 220, 485
COMMAND\Xlinenumannotationposition	26, 27, 67, 173
COMMAND\Xlinenumannotationposition@side	173
COMMAND\Xlinenumannotationpositionside	27
COMMAND\Xlinerangeseparator	46, 67, 210, 408, 477
COMMAND\Xmaxhnotes	55, 79, 80, 405, 406, 471, 473
COMMAND\Xmorethantwolines	30, 46, 47, 67, 406, 473
COMMAND\Xnoidenticallinenumannotation	28, 67, 220, 484
COMMAND\Xnoindent	477
COMMAND\Xnolemmaseparator	50, 289, 406
COMMAND\Xnonbreakableafternumber	48, 406, 469
COMMAND\Xnonumber	47, 406
COMMAND\Xnonumberinfootnote	406
COMMAND\Xnotefontsize	51, 405
COMMAND\Xnotefontsize@⟨s⟩	200, 204, 205
COMMAND\Xnotenumfont	51, 405
COMMAND\Xnoteswidthliketwocolumns	56, 472
COMMAND\Xnumberonlyfirstinline	44, 46, 54, 114, 212, 253, 281, 282, 285, 406, 468, 473, 482
COMMAND\Xnumberonlyfirstintwolines	44, 54, 406, 468, 482
COMMAND\Xonlypstart	47, 406, 468, 473
COMMAND\Xpagelinesep	48, 481
COMMAND\Xparafootsep	54, 114, 405, 406, 410, 480, 482, 483
COMMAND\Xparafootsep@series	200
COMMAND\Xparindent	52, 408, 410, 474, 477, 479, 485
COMMAND\Xprenotes	55, 249, 480
COMMAND\Xprenotes@	193, 249, 468
COMMAND\Xprintlines	113
COMMAND\Xpstart	47, 403, 406, 468, 473

COMMAND\Xpstarteverytime	47, 403, 406, 473
COMMAND\Xpstartonlyfirst	47, 486
COMMAND\Xpstartseparator	47, 486
COMMAND\Xragged	54
COMMAND\Xrigidbalance	202, 408, 478
COMMAND\Xsethangindent	209
COMMAND\Xstanza	47, 61
COMMAND\Xstanzaonlyfirst	47, 486
COMMAND\Xstanzaseparator	47
COMMAND\Xstorelineinfo	212
COMMAND\Xsublinesep	23, 47, 48, 67, 210, 408
COMMAND\Xsublinesepside	23, 48
COMMAND\Xsymlinenum	44, 54, 405, 406, 475, 482
COMMAND\Xsymlinenumannotation	45, 485
COMMAND\Xtextbeforenotes	184
COMMAND\Xtoendnotes	32, 275
COMMAND\Xtwolines	30, 46, 47, 67, 219, 221, 281, 406, 473
COMMAND\Xtwolinesappref	281
COMMAND\Xtwolinesbutnotmore	46, 47, 67, 406, 474
COMMAND\Xtwolinesbutnotmoreappref	281
COMMAND\Xtwolinesonlyinsamepage	46, 47, 67, 406, 474
COMMAND\Xtxtbeforenotes	54, 55, 406, 480, 481, 483
COMMAND\Xtxtbeforenotesonlyonce	54, 483
COMMAND\Xtxtbeforenumber	46
COMMAND\Xunvxh	197, 407
COMMAND\Xwidth	56, 408, 479
COMMAND\Xwrapcontent	52, 480
COMMAND\Xwraplemma	51–53, 480
COMMAND\Xwraplinenumannotation	27, 67, 173
COMMAND\Xwraplinenumannotation@ref	173
COMMAND\Xwraplinenumannotation@side	173
COMMAND\Xwraplinenumannotationref	27
COMMAND\Xwraplinenumnotationside	27
COMMAND\&	405
COMMAND\absline@num	111, 169
COMMAND\accent	140
COMMAND\actionlines@list	112, 170
COMMAND\actions@list	112
COMMAND\add@Xgroupbyline	181
COMMAND\add@hooktoggle@specific@to@cs	287
COMMAND\add@inserts	112, 180, 181
COMMAND\add@inserts@next	180, 181
COMMAND\add@msd@	350
COMMAND\add@msdata	350, 352
COMMAND\add@msdata@firstlineofpage	355
COMMAND\add@msddata	350
COMMAND\add@penalties	168, 182
COMMAND\addcontentsline	382
COMMAND\addfootins	401, 404
COMMAND\addfootinsX	401, 404

COMMAND\addtoendnotes	275
COMMAND\advancelabel@refs	299
COMMAND\advanceline	24, 113, 121, 136, 476
COMMAND\affixline@num	174, 177, 179, 321, 466
COMMAND\affixpstart@num	179
COMMAND\afterXrule	405
COMMAND\afterXsymlinenum	405
COMMAND\afterantilabe	62
COMMAND\aftergroup	139, 144
COMMAND\afterlemmaseparator	405
COMMAND\afternote	405
COMMAND\afternoteX	54, 404, 405
COMMAND\afternumberinfootnote	405
COMMAND\afterruleX	55, 471, 474
COMMAND\annot@end@print	220, 260
COMMAND\annot@list	113, 174
COMMAND\annotationref	27, 63
COMMAND\antilabe	16, 62, 486
COMMAND\applabel	66, 301–303, 474, 480
COMMAND\appref	64, 66–68, 307, 310, 478, 479, 485
COMMAND\apprefprefixmore	66, 405
COMMAND\apprefprefixsingle	66, 405
COMMAND\apprefwithpage	66–68, 307, 310, 475, 478, 486
COMMAND\arrangementX	43, 80, 229, 279, 405
COMMAND\arrangementX@normal	235
COMMAND\article	16
COMMAND\at@every@pend	160
COMMAND\autopar	19, 156, 161, 162, 246, 467, 469, 470, 474, 484
COMMAND\ballast	80
COMMAND\ballast@count	168, 182
COMMAND\baselineskip	43, 196, 200
COMMAND\beforeXnotes	405
COMMAND\beforeXsymlinenum	405
COMMAND\beforeantilabe	62
COMMAND\beforeeledchapter	11, 77, 383
COMMAND\beforeinsertingX	52
COMMAND\beforelemmaseparator	405
COMMAND\beforenotesX	55, 407, 470, 471, 474
COMMAND\beforenumberinfootnote	405
COMMAND\begin	359
COMMAND\beginnumbering	18–21, 96, 97, 101, 102, 110, 115, 131, 161, 251, 349, 468, 471, 475, 476, 481, 482
COMMAND\beginnumberingR	96
COMMAND\bf	468
COMMAND\bfseries	51, 468
COMMAND\bhookXnote	406
COMMAND\bhookgroupX	55, 478
COMMAND\bhooknoteX	52, 407, 476, 477
COMMAND\body	340
COMMAND\bodyfootmarkA	40

COMMAND\book	16
COMMAND\boxXendlinenum	406
COMMAND\boxXendlinenumalign	406
COMMAND\boxXendstartlinenum	406
COMMAND\boxendlinenum	406
COMMAND\boxlinefootnote	214
COMMAND\boxlinenum	406
COMMAND\boxlinenumalign	406
COMMAND\boxstartlinenum	406
COMMAND\boxsymlinenum	406
COMMAND\break	44, 197
COMMAND\brokenpenalty	182
COMMAND\c@page	132
COMMAND\c@page@page	132
COMMAND\centering	53
COMMAND\ch@ck@l@ck	466
COMMAND\ch@cksub@l@ck	177, 466
COMMAND\chapter	76, 384, 388, 471, 474, 476, 480
COMMAND\chaptermark	382
COMMAND\check@pb@in@verse	396
COMMAND\colalignX	53, 474
COMMAND\collect@body	360
COMMAND\color	480
COMMAND\colorbox	81
COMMAND\columns	56
COMMAND\columnwidth	196, 472
COMMAND\command names	281
COMMAND\continuousnumberingwithcolumns@sync@linenumber@singletext	172
COMMAND\copyright	140
COMMAND\correct@Xfootins@box	473
COMMAND\correct@footinsX@box	473
COMMAND\count	203
COMMAND\critex	467
COMMAND\critext	147, 400, 401, 404
COMMAND\csname	85
COMMAND\csquotes	271
COMMAND\ctab	376, 380
COMMAND\ctabtext	381
COMMAND\current@annot	174
COMMAND\dc col	369
COMMAND\def	82, 99
COMMAND\dimen	203
COMMAND\dimexpr	56
COMMAND\discretionary	197
COMMAND\displaywidowpenalty	182
COMMAND\do@Xfeet	290, 466, 476, 481
COMMAND\do@actions	168, 169, 171, 466
COMMAND\do@actions@fixedcode	466
COMMAND\do@actions@next	169, 170
COMMAND\do@ballast	168, 169, 182

COMMAND\do@feet@custom@order	289
COMMAND\do@insidelinehook	469
COMMAND\do@line	112, 139, 159, 162, 166, 168, 180, 182, 340, 466, 467, 469, 471
COMMAND\do@linehook	466
COMMAND\do@lockoff	113
COMMAND\do@lockon	113
COMMAND\dodoreinextrafeet	465
COMMAND\doendnotes	31, 256, 407, 475, 482
COMMAND\doendnotesbysection	31, 256, 275, 475, 482
COMMAND\doennotes	482
COMMAND\doinsidelinehook	28, 471
COMMAND\doinsidethislinehook	28, 166, 484
COMMAND\dolinehook	28, 471
COMMAND\doreinextrafeeti	476
COMMAND\doreinextrafeetii	476
COMMAND\doxtrafeet	465
COMMAND\doxtrafeeti	476
COMMAND\doxtrafeetii	476
COMMAND\dummy@ref	140
COMMAND\edaftertab	75, 375, 376
COMMAND\edannotationref	304
COMMAND\edatleft	75, 373
COMMAND\edatright	75, 374
COMMAND\edbeforetab	75, 375, 376
COMMAND\edfilldimen	374
COMMAND\edfont@info	147
COMMAND\edgls	72, 328
COMMAND\edgls...	479
COMMAND\edglsadd	482
COMMAND\edindex	38, 70, 72, 328, 329, 332, 334, 335, 364, 469, 471, 473, 476, 477, 481–483, 485
COMMAND\edindexlab	72
COMMAND\edlabel	63– 66, 68, 140, 297, 299, 301, 303, 305, 313, 328, 364, 397, 409, 465, 468–470, 473, 478, 483
COMMAND\edlabelE	65, 303
COMMAND\edlabelS	65, 303
COMMAND\edlabelSE	65
COMMAND\edlineref	63, 297, 406, 473, 475, 478, 482
COMMAND\edmakelabel	65, 313
COMMAND\edpageref	63, 297, 303, 313
COMMAND\edrowfill	375
COMMAND\edsublineref	63
COMMAND\edtabcolsep	368
COMMAND\edtext	7, 25, 26, 29, 30, 32, 33, 35–37, 57, 63, 64, 66, 70, 73, 80, 100, 111–113, 128, 129, 133–135, 138–147, 150, 152, 154, 172– 174, 301, 302, 306, 339, 364, 365, 384, 397, 400, 401, 404, 465, 467, 469, 471–475, 481–485
COMMAND\edtext@level	475
COMMAND\edtextlater	130
COMMAND\edvertdots	76, 374
COMMAND\edvertline	75, 76, 374
COMMAND\elechapter	77

COMMAND\eled@sectioning@out	389
COMMAND\eledchapter	76, 404, 471, 476
COMMAND\eledchapter*	76
COMMAND\eledmac@error	465
COMMAND\eledsection	7, 17, 76, 140, 165, 383, 404, 473, 482, 484, 485
COMMAND\eledsection*	76
COMMAND\eledsubsection	76, 404
COMMAND\eledsubsection*	76
COMMAND\eledsubsubsection	76, 404
COMMAND\eledsubsubsection*	76
COMMAND\eledxxx	11, 77, 389, 471
COMMAND\eledxxxx	382
COMMAND\elesection	389
COMMAND\else	327, 383
COMMAND\emph	38
COMMAND\empty	95, 175, 176, 297
COMMAND\end	359
COMMAND\end@lemmas	139
COMMAND\endashchar	210
COMMAND\endgraf	159, 199, 246
COMMAND\endlock	24, 113, 137, 345
COMMAND\endminipage	322, 324, 465, 476
COMMAND\endmsdata	40
COMMAND\endnotes	474, 478
COMMAND\endnumbering	18, 19, 21, 97, 99, 100, 131, 466, 475, 481, 482
COMMAND\endnumberingR	96
COMMAND\endprint	251, 254, 275, 403
COMMAND\endstanzaextra	405
COMMAND\endsub	23, 113, 136
COMMAND\endverse	407
COMMAND\everypar	161
COMMAND\extensionchars	78, 97
COMMAND\externaldocument	68, 313
COMMAND\f@x@l@cks	466
COMMAND>falseverse	404, 469, 471
COMMAND\fi	383
COMMAND\firstlinenum	22, 175, 467
COMMAND\firstsublinenum	22, 467
COMMAND\flag@end	133–135, 146, 471
COMMAND\flag@end@RTL	135
COMMAND\flag@end@later	135
COMMAND\flag@start	133–135, 146, 471, 472
COMMAND\flag@start@RTL	135
COMMAND\flag@start@later	135
COMMAND\flagstanza	62
COMMAND\floatingpenalty	188, 473
COMMAND\flush@notes	183, 184
COMMAND\fnpos	42, 248, 470, 481
COMMAND\footfmt	188, 190
COMMAND\footfmt...	230



COMMAND\footfootmarkA	40
COMMAND\footfudgefactor	198
COMMAND\footfudgefiddle	80, 195, 196, 465
COMMAND\footgroup	188
COMMAND\footins	193
COMMAND\footnormal	280, 405, 465
COMMAND\footnormalX	405
COMMAND\footnote	40, 79, 225–227, 402, 466
COMMAND\footnote@lang	210
COMMAND\footnoteA	17, 40
COMMAND\footnoteB	17
COMMAND\footnoteC	29
COMMAND\footnoteE	40
COMMAND\footnoteX	9, 41, 271, 273, 481
COMMAND\footnoteX@reading	272
COMMAND\footnoteXmark	273, 482, 485
COMMAND\footnoteXmk	288
COMMAND\footnoteXtext	273, 482, 485
COMMAND\footnote⟨X⟩	141
COMMAND\footnote⟨X⟩mark	41
COMMAND\footnote⟨X⟩mk	41
COMMAND\footnote⟨X⟩nomk	41
COMMAND\footnote⟨X⟩text	41
COMMAND\footnotelang@lua	187
COMMAND\footnotelang@poly	187
COMMAND\footnotemark	41, 273
COMMAND\footnoteoption@	186, 477
COMMAND\footnoterule	203
COMMAND\footnotesize	51
COMMAND\footnotetext	41, 273
COMMAND\footparagraph	196, 280, 405, 471
COMMAND\footparagraphX	240, 405, 471
COMMAND\footsplitskips	466, 473
COMMAND\footstart	188, 193, 203
COMMAND\footstrut	199
COMMAND\footthreecol	405
COMMAND\footthreecolX	405, 474
COMMAND\foottwocol	405
COMMAND\foottwocolX	405, 474
COMMAND\foreignlanguage	52
COMMAND\fullstop	408
COMMAND\get@edindex@hyperref	333
COMMAND\get@edindex@ledinnote@command	329
COMMAND\get@fnmark	227
COMMAND\get@index@command	470
COMMAND\get@linelistfile	466
COMMAND\get@thisfootnote	234
COMMAND\getline@num	168, 170
COMMAND\gl@p	103
COMMAND\global	117

COMMAND\globaldefs	117
COMMAND\gls	72, 336
COMMAND\hangindentX	52, 474, 477, 485
COMMAND\hangingsymbol	405, 467
COMMAND\hbox	197
COMMAND\hfill	469
COMMAND\hidenumbering	25, 126, 409, 474, 484, 486
COMMAND\hidenumberingonleftpage	25, 126, 480
COMMAND\hidenumberingonrightpage	25, 126, 480
COMMAND\hline	73
COMMAND\hooktoggle@specific	287
COMMAND\hrulefill	375
COMMAND\hspace	44, 193, 196, 198, 204, 207, 246, 466, 472
COMMAND\hspaceX	408, 478, 479
COMMAND\hsizethreecol	406
COMMAND\hsizethreecolX	53, 56
COMMAND\hsizetwocol	406
COMMAND\hsizetwocolX	53, 56
COMMAND\hyperlinkR	333
COMMAND\hyperlinkformat	333
COMMAND\hyperlinkformatR	333
COMMAND\if@RTL	87, 143
COMMAND\if@edtext@	472, 475
COMMAND\if@eled@sectioning	389
COMMAND\if@firstlineofpage	87
COMMAND\if@firstlineofpageR	87
COMMAND\if@msd@options@fullpage	357
COMMAND\if@msdata@insertedfrompreviouspage	354
COMMAND\if@nobreak	160
COMMAND\if@noneed@Footnote	133
COMMAND\ifXnote@	96
COMMAND\ifafterendnumberingR	96
COMMAND\ifautopar	161
COMMAND\ifby@autopar	161
COMMAND\ifbypage@	103
COMMAND\ifbypage@R	104
COMMAND\ifbypstart@	103
COMMAND\ifbypstart@R	104
COMMAND\iffirst@linenum@out@	131
COMMAND\ifindtl@innote	96
COMMAND\ifindtl@notenumber	96
COMMAND\ifinserthangingsymbol	340
COMMAND\ifinstanza	340
COMMAND\ifstwofollowinglines	221
COMMAND\ifl@d@Xmorethantwolines	216, 473
COMMAND\ifl@d@Xtwolines	216
COMMAND\ifl@d@dash	216
COMMAND\ifl@d@elin	216
COMMAND\ifl@d@esl	216
COMMAND\ifl@d@pnum	216

COMMAND\ifl@d@ssub	216
COMMAND\ifl@dend@X	274
COMMAND\ifl@dmemoir	465
COMMAND\ifl@dpaging	472
COMMAND\ifl@dpairing	95, 467
COMMAND\ifl@dprintingpages	473
COMMAND\ifl@dskipnumber	175
COMMAND\ifl@dstartendok	374
COMMAND\ifl@imakeidx	87
COMMAND\ifledRcol	96, 467
COMMAND\ifledRcol@	96, 471
COMMAND\iflemmacommand@	472
COMMAND\ifnoend@	256
COMMAND\ifnoedgroup@	327
COMMAND\ifnoteschanged@	114
COMMAND\ifnumberedpar@	156
COMMAND\ifnumbering	97, 100
COMMAND\ifnumberingR	96, 467
COMMAND\ifnumberline	146, 175
COMMAND\ifpst@rted	467
COMMAND\ifpst@rtedL	98
COMMAND\ifresumenummering@start	101
COMMAND\ifseriesbefore	278
COMMAND\ifstopmsdata@inserted@	349
COMMAND\ifsublines@	110, 124
COMMAND\iftrue	475
COMMAND\ifvmode	299
COMMAND\ifxxx	383
COMMAND\ignorespaces	143
COMMAND\imki@wrindexentry	87
COMMAND\immediate	131, 251
COMMAND\indent	19, 161, 407
COMMAND\index	336, 481
COMMAND\indtl@wrindexentry	87
COMMAND\initnumbering@quote	381, 476
COMMAND\initnumbering@reg	466
COMMAND\initnumbering@sectcmd	476
COMMAND\innotemarkX	70, 410, 486
COMMAND\inplaceoflemmaseparator	406
COMMAND\inplaceofnumber	406
COMMAND\insert	145, 180, 188, 190, 191, 205, 224, 230
COMMAND\insert@Xtxtbeforenotes	184, 205
COMMAND\insert@count	128, 134, 143
COMMAND\insert@countR	143
COMMAND\insert@msdata	350, 357
COMMAND\insert@txtbeforenotesX	185
COMMAND\inserthangingsymbol	469
COMMAND\insertlines@list	112, 128
COMMAND\insertparafootsepX	244
COMMAND\inserts@list	139, 156, 180, 197

COMMAND\interAfootnotelinepenalty	467
COMMAND\interfootnotelinepenalty	467
COMMAND\interlinepenalty	188
COMMAND\interparanoteglua	404
COMMAND\justifying	53
COMMAND\keep@this@crossref@forinserting	225
COMMAND\l@advance@parledegroupp@beforenormalnotes	476
COMMAND\l@d@@wrindexhyp	472
COMMAND\l@d@add	148
COMMAND\l@d@end	251, 274
COMMAND\l@d@nums	143, 146–148, 216, 217
COMMAND\l@d@section	251
COMMAND\l@d@set	122, 137
COMMAND\l@dampcount	366
COMMAND\l@dbfnote	227, 466
COMMAND\l@dcheckstartend	374
COMMAND\l@dchset@num	122
COMMAND\l@dcolcount	366, 367
COMMAND\l@dcollect@@body	359
COMMAND\l@dcollect@body	359
COMMAND\l@dcsnote	471
COMMAND\l@dcsnotetext	167, 318
COMMAND\l@dcsnotetext@l	167, 318
COMMAND\l@dcsnotetext@r	167, 318
COMMAND\l@ddodoreintrafeet	291, 465
COMMAND\l@ddoxtrafeet	289, 291, 465
COMMAND\l@emptyd@ta	466
COMMAND\l@dend@close	251
COMMAND\l@dend@open	251
COMMAND\l@dend@stuff	251
COMMAND\l@denvbody	359
COMMAND\l@dfeetbeginmini	465
COMMAND\l@dfeetendmini	465
COMMAND\l@dgetline@margin	467
COMMAND\l@dgetlock@disp	467
COMMAND\l@dgetref@num	305, 306
COMMAND\l@dgetsidenote@margin	315, 467
COMMAND\l@dgobbeloptarg	472
COMMAND\l@dgobblearg	472
COMMAND\l@dgobbleoptarg	364
COMMAND\l@dlabel@parse	305, 306
COMMAND\l@dld@ta	174, 176
COMMAND\l@dlp@rbox	320
COMMAND\l@dlsn@te	467
COMMAND\l@dlsnote	471
COMMAND\l@dmake@labels	300, 314
COMMAND\l@dmake@labelsR	314
COMMAND\l@dnumpsstartsL	98, 467
COMMAND\l@dp@rsefootspec	217
COMMAND\l@dp@parsefootspec	217

COMMAND\l@dpush@begins	359
COMMAND\l@dtd@ta	174, 176
COMMAND\l@dref@undefined	305
COMMAND\l@drrn@te	467
COMMAND\l@drrnote	471
COMMAND\l@dtabaddcols	374
COMMAND\l@dtabnoexpands	465
COMMAND\l@dumboxmpfoot	476
COMMAND\l@dunboxmpfoot	466
COMMAND\l@dunhbox@line	162
COMMAND\l@dzeropenalties	467, 472
COMMAND\l@noexpands@edgl	339
COMMAND\l@pb	395
COMMAND\l@prev@nopb	394
COMMAND\l@prev@pb	394
COMMAND\l@reg	402
COMMAND\l@secondoptarg	364
COMMAND\l@wrap@ifnotemptybox	94
COMMAND\label	20, 65, 68, 72, 298, 306
COMMAND\label@refs	297
COMMAND\labelstarttrue	20, 409, 468, 483
COMMAND\labelref@list	297, 301
COMMAND\language	197
COMMAND\lastbox	161
COMMAND\lastskip	136
COMMAND\latex@makecol	292
COMMAND\leavevmode	19, 161
COMMAND\led@check@nopb	395
COMMAND\led@check@pb	395
COMMAND\led@nopb	395, 396
COMMAND\led@nopbnum	395
COMMAND\led@pb	395, 396
COMMAND\led@pb@macro	395
COMMAND\led@pbnum	395
COMMAND\led@reinit@index@fornote	336
COMMAND\led@set@index@fornote	336
COMMAND\ledRflag	333
COMMAND\ledchapter	404, 468
COMMAND\ledfootinsdim	405
COMMAND\ledinnernote	68, 316, 471, 480
COMMAND\ledinnote	330, 403, 410, 475
COMMAND\ledinnotemark	403, 410, 474, 486
COMMAND\ledleftnote	69, 316
COMMAND\ledlinenum	109, 467
COMMAND\ledllfill	168
COMMAND\ledlsnotefontsetup	480
COMMAND\ledlsnotesep	69
COMMAND\ledlsnotewidth	69
COMMAND\lednopb	78, 395
COMMAND\lednopbinverse	396

COMMAND\lednopbinversettrue	60, 78
COMMAND\lednopbnum	395
COMMAND\ledouternote	68, 316, 471, 480
COMMAND\ledpb	78, 395, 404
COMMAND\ledpbnum	395
COMMAND\ledpbsetting	78, 395, 477
COMMAND\ledrightnote	69, 316
COMMAND\ledrsnotefontsetup	480
COMMAND\ledrsnotesep	69
COMMAND\ledrsnotewidth	69
COMMAND\ledsection	404
COMMAND\ledsectnomark	382
COMMAND\ledsectnotoc	382
COMMAND\ledsetnormalparstuff	403, 404, 474
COMMAND\ledsetnormalparstuff@common	246
COMMAND\ledsetnormalparstuffX	403, 404, 474
COMMAND\ledsidenote	68, 69, 316, 318
COMMAND\ledsubsection	404
COMMAND\ledsubsubsection	404
COMMAND\ledxxx	471
COMMAND\left	75
COMMAND\leftctab	376
COMMAND\leftheadline	109
COMMAND\leftlinenum	23, 109, 465, 467
COMMAND\leftltab	375
COMMAND\leftnoteupfalse	69
COMMAND\leftpstartnum	179
COMMAND\leftrtab	375
COMMAND\leftsidenote	318
COMMAND\leftskip	193, 196, 198
COMMAND\lemma	3, 30, 32, 33, 35–37, 138, 142–144, 147, 150, 400, 467, 468, 475, 476, 478
COMMAND\lemmaseparator	406
COMMAND\let	33, 57, 141, 145, 273, 292, 345, 465
COMMAND\letboxXendendlinenum	406
COMMAND\line	202, 205
COMMAND\line@list	111, 113, 129, 146
COMMAND\line@list@stuff	97, 115, 131, 465, 467
COMMAND\line@list@version	117
COMMAND\line@margin	105, 176, 315
COMMAND\line@num	110–112, 175, 465
COMMAND\line@set	148
COMMAND\lineannot	26, 64, 148
COMMAND\lineannot@set	148
COMMAND\lineation	22, 104, 483–485
COMMAND\linebreak	44
COMMAND\lineneation	479
COMMAND\linenum	26, 30, 32, 33, 64, 66, 138, 147, 148, 303, 306, 313, 400
COMMAND\linenum@out	130, 131, 297, 301
COMMAND\linenumannotation	25, 172, 483, 484
COMMAND\linenumberlist	22, 95, 175, 176, 465

COMMAND\linenumberstyle	24, 108, 465, 482
COMMAND\linenumincrement	22, 467
COMMAND\linenummargin	23, 105, 315
COMMAND\linenumr@p	108, 465, 467
COMMAND\linenumrep	108, 467
COMMAND\linenumsep	23, 69, 109, 316
COMMAND\linerangesep@	288
COMMAND\lineref	297, 303, 313, 406, 473
COMMAND\list@clear	102
COMMAND\list@clearing@reg	467
COMMAND\list@create	102
COMMAND\lock@disp	107
COMMAND\lock@off	125
COMMAND\lock@on	124
COMMAND\lockdisp	24, 107
COMMAND\loop	184, 340
COMMAND\ltab	375, 376, 380
COMMAND\ltabtext	381
COMMAND\m@mmf@prepare	226
COMMAND\mainmatter	483, 484
COMMAND\makeatletter	166
COMMAND\makehboxoffhboxes	198, 200
COMMAND\makeindex	70, 332
COMMAND\makelabel	313
COMMAND\manage stanza@modulo	341
COMMAND\marginpar	68, 79, 314, 315, 466
COMMAND\marginparwidth	69, 316
COMMAND\markboth	28, 166
COMMAND\mathchardef	341
COMMAND\maxhXnotes	406
COMMAND\maxhnotesX	55, 80, 405, 470, 471, 473, 474
COMMAND\maxlinesinpar@list	115
COMMAND\measurebody	377
COMMAND\measuretbody	378
COMMAND\memorybreak	21
COMMAND\metaXXXvfootnote	224
COMMAND\microtype	162
COMMAND\morenoexpands	80, 81, 139, 140
COMMAND\morethantwolines	406
COMMAND\mpfnpos	42, 248, 470, 481
COMMAND\mpnormalfootgroup	465
COMMAND\mpnormalvfootnote	466
COMMAND\msdata	39, 40, 348, 349, 351, 482, 484
COMMAND\msdataposition	40
COMMAND\multfootsep	41, 226
COMMAND\multiplefootnotemarker	226
COMMAND\musixtex	470
COMMAND\n@num	467, 474
COMMAND\n@num@ref	474
COMMAND\new@line	132, 466

COMMAND\new@series	141
COMMAND\newcommand	33, 82, 99, 226, 300
COMMAND\newcommandx	33
COMMAND\newhookarg@specific	287
COMMAND\newhookcommand@series	281, 282, 474, 486
COMMAND\newhookcommand@series@reload	282
COMMAND\newhookcommand@toggle@reload	282, 472
COMMAND\newhooktoggle@series	281, 282, 474
COMMAND\newhooktoggle@specific	287
COMMAND\newif	474
COMMAND\newline	44
COMMAND\newlinechar	274
COMMAND\newseries	41, 401, 404, 405
COMMAND\newseries@	264, 265, 279
COMMAND\newverse	60, 62, 404, 471, 482
COMMAND\next	340
COMMAND\next@action	116
COMMAND\next@actionline	116
COMMAND\next@insert	181
COMMAND\next@line@list@stuff	131, 483
COMMAND\nl@regR	118
COMMAND\no@expands	80, 141, 147, 273, 339, 465
COMMAND\noXlemmaseparator	406
COMMAND\nobreak	216
COMMAND\nocritical	265
COMMAND\noeledsec	404
COMMAND\noendnotes	404
COMMAND\noexpand	402
COMMAND\nofamiliar	284
COMMAND\noindent	19, 20, 60, 61, 161, 477
COMMAND\noindentX	477
COMMAND\nomk@	288
COMMAND\nonbreakableafternumber	406
COMMAND\nonumberinfootnote	406
COMMAND\norelax	57
COMMAND\normal@footnotemarkX	230
COMMAND\normal@page@break	394
COMMAND\normal@pars	246
COMMAND\normalbfnoteX	467
COMMAND\normalbodyfootmarkX	230
COMMAND\normalfont	481
COMMAND\normalfootfmt	57, 192, 199, 210, 251, 480
COMMAND\normalfootfmtX	231
COMMAND\normalfootfootmarkX	231
COMMAND\normalfootgroup	194
COMMAND\normalfootgroupX	232
COMMAND\normalfootnoterule	189
COMMAND\normalfootstart	193, 196
COMMAND\normalfootstartX	231
COMMAND\normalvfootnote	190, 192



COMMAND\normalvfootnote@inserted	191, 192
COMMAND\normalvfootnoteX	230
COMMAND\notbool	383
COMMAND\notfontsetup	405
COMMAND\notfontsizeX	51, 405
COMMAND\notenumfont	405
COMMAND\notenumfontX	51, 405
COMMAND\notesXwidthliketwocolumns	406
COMMAND\noteswidthliketwocolumnsX	56, 406, 472, 474
COMMAND\num@lines	156, 182
COMMAND\numberlinefalse	21, 484
COMMAND\numberlinetrue	21, 484
COMMAND\numberonlyfirstinline	278, 406
COMMAND\numberonlyfirstintwolines	406
COMMAND\numberpstartfalse	20
COMMAND\numberpstarttrue	20, 47, 403, 467, 476, 482, 484
COMMAND\numberstanza	47
COMMAND\numberstanzafalse	61
COMMAND\numberstanzatrue	61
COMMAND\numlabfont	23, 57, 109
COMMAND\one@line	156
COMMAND\onehalfspacing	477
COMMAND\onlyXpstart	406
COMMAND\onlysideX	272
COMMAND\page@action	112, 123
COMMAND\page@start	112, 467
COMMAND\pagecontents	112
COMMAND\pagenumbering	295
COMMAND\pageparbreak	404
COMMAND\pageref	65, 303
COMMAND\par	32, 44, 161, 246
COMMAND\par@line	156, 182
COMMAND\par@patch@pagenumbering	295
COMMAND\par@patch@thepage	132, 295
COMMAND\para@footgroup	196
COMMAND\para@footgroupX	243
COMMAND\para@footsetup	196, 465
COMMAND\para@footsetupX	241, 465, 472
COMMAND\para@vfootnoteX	242
COMMAND\parafootfmt	198, 199, 480
COMMAND\parafootfmtX	243
COMMAND\parafootftm	201
COMMAND\parafootftmX	244
COMMAND\parafootftmsep	405
COMMAND\parafootsep	406, 470, 475
COMMAND\parafootsepX	54, 114, 405, 406, 410, 480, 485, 486
COMMAND\parafootstart	196
COMMAND\parafootstartX	241
COMMAND\paravfootnote	197, 200
COMMAND\parfillskip	199

COMMAND\parindent	477
COMMAND\parindentX	52, 408, 477, 479, 485
COMMAND\parse@annot	174
COMMAND\parshape	80
COMMAND\parskip	20, 60, 61, 161
COMMAND\pause@numbering	100
COMMAND\pausenumbering	21, 100, 101, 115, 117, 162, 470, 472, 479
COMMAND\pausenumbering@page@num	100
COMMAND\penalty	199
COMMAND\pend	2, 7, 18–21, 23, 24, 77, 137, 139, 142, 149, 156, 158–162, 179, 180, 347, 402, 470, 471, 480–483
COMMAND\preXnotes	474, 480
COMMAND\prenotesX	55, 250, 474
COMMAND\prepare@Xgroupbyline	224, 225
COMMAND\prepare@Xprenotes	248
COMMAND\prev@nopb	395
COMMAND\prev@pb	395
COMMAND\prevlineX	114
COMMAND\prevpageX@num	114
COMMAND\print@Xfootnoterule	474
COMMAND\print@Xnotes	290, 291
COMMAND\print@Xnotes@forpages	473
COMMAND\print@eledsection	165
COMMAND\print@footnoteXrule	474
COMMAND\print@leftmargin@eledsection	383
COMMAND\print@lemma	211
COMMAND\print@line	163
COMMAND\print@notesX@forpages	473
COMMAND\print@rightmargin@eledsection	383
COMMAND\printendlines	257, 264, 311, 408, 465, 467
COMMAND\printlinefootnote	212, 213, 473
COMMAND\printlinefootnotearea	214, 216, 473
COMMAND\printlinefootnotenumbers	212
COMMAND\printlines	192, 210, 216, 217, 257, 310, 408, 465, 467, 473, 478
COMMAND\printnpnum	403, 408
COMMAND\printpstart	211
COMMAND\protect	140, 402
COMMAND\providecommand	226, 465
COMMAND\pstarL	156
COMMAND\pstart	2, 7, 18–21, 23, 24, 76, 77, 122, 137, 142, 145, 149, 156, 159–161, 165, 180, 347, 388, 402, 403, 467–469, 471, 472, 474–476, 479–486
COMMAND\pstartinfootnote	406
COMMAND\pstartinfootnoteeverytime	406
COMMAND\pstartnum	179
COMMAND\pstartref	63, 297, 305, 470, 482
COMMAND\pstarts	467, 468
COMMAND\pstarts@readL	156
COMMAND\pstarts@typeset@L	156
COMMAND\raggedX	54
COMMAND\raggedleft	53

COMMAND\raggedright	53
COMMAND\raw@text	156
COMMAND\rbracket	50
COMMAND\read@linelist	115–117
COMMAND\ref	65, 68, 72
COMMAND\reformatted@	310
COMMAND\reformattedwithpage	310
COMMAND\relax	20, 122, 145, 169, 180, 345, 364, 402
COMMAND\renewcommand	80, 405, 408
COMMAND\reset@current@annot	174
COMMAND\reset@msd@options@iffullpage	357
COMMAND\resetprevline@	114
COMMAND\resetprevpage@	114
COMMAND\resumenumbering	21, 96, 97, 101, 102, 115, 117, 162, 467, 471, 472, 479
COMMAND\right	75
COMMAND\rightctab	376
COMMAND\rightlinenum	23, 109, 465, 467
COMMAND\rightltab	376
COMMAND\rightnoteupfalse	69
COMMAND\rightrtab	376
COMMAND\rightrightnote	318
COMMAND\rightrightskip	193, 196, 198, 199
COMMAND\rightrightstartnum	179
COMMAND\rigidbalance	202, 203, 205, 408, 478
COMMAND\rigidbalanceX	202, 408, 478
COMMAND\robustify	44, 45
COMMAND\roman	369, 478
COMMAND\rtab	375–377, 380
COMMAND\rtabtext	377, 381
COMMAND\sameparallelpagenumber	132
COMMAND\sameword	17, 33–37, 130, 149–152, 154, 410, 473, 475, 477, 480, 483, 485
COMMAND\sameword@inedtext	150
COMMAND\saweword	150
COMMAND\scriptsize	109
COMMAND\section	76, 467, 481
COMMAND\section@num	97
COMMAND\sectionmark	382
COMMAND\select@lemmafont	57, 186
COMMAND\series	264, 265
COMMAND\series@	264
COMMAND\seriesatbegin	42, 277, 474
COMMAND\seriesatend	42, 278, 474
COMMAND\set@Xtxtbeforenotes	184
COMMAND\set@continuousnumberingforl	102
COMMAND\set@line	146
COMMAND\set@line@action	113, 123
COMMAND\set@txtbeforenotesX	184
COMMAND\setSErefonlypageprefixmore	66, 310, 479, 482
COMMAND\setSErefonlypageprefixsingle	66, 310, 479, 482
COMMAND\setSErefprefixmore	66

COMMAND\setSErefprefixsingle	66
COMMAND\setapprefprefixmore	66, 405
COMMAND\setapprefprefixsingle	66, 68, 405, 478
COMMAND\setcommand@series	280
COMMAND\sethangindentX	209
COMMAND\sethangingsymbol	60, 339, 405, 477
COMMAND\sethanginsymbol	58
COMMAND\setistwofollowinglines	221
COMMAND\setl@dlprbox	320
COMMAND\setline	24, 113, 118, 122, 137, 140, 476
COMMAND\setlinenum	24, 118, 122, 137, 160, 465, 483
COMMAND\setlinenumannotationsep	28, 409, 484
COMMAND\setmsdatalabel	40
COMMAND\setmsdataposition	482
COMMAND\setmsdataseries	40
COMMAND\setprintendlines	257, 260, 467
COMMAND\setprintlines	217, 221, 257, 467
COMMAND\setsidenoteseq	69
COMMAND\setsidenotsep	405
COMMAND\setstanzaindent	342
COMMAND\setstanzaindents	59, 341, 402, 486
COMMAND\setstanzapenalties	341
COMMAND\setstanzavalues	341
COMMAND\settoggle@series	279, 468, 472
COMMAND\showlemma	140, 465, 466
COMMAND\showwordrank	37, 151
COMMAND\sidenote@margin	466
COMMAND\sidenotemargin	68, 466, 471
COMMAND\sidenoteseq	405
COMMAND\sidepstartnumtrue	20
COMMAND\skip	193
COMMAND\skipnumbering	25, 126, 138, 467, 474, 475
COMMAND\skipnumbering@reg	474
COMMAND\small	51
COMMAND\somemacro	38
COMMAND\special	14
COMMAND\splitmaxdepth	188, 204
COMMAND\splitoff	202
COMMAND\splittopskip	188, 204, 205
COMMAND\stanza	24, 25, 60–62, 96, 345, 405, 407, 477, 482
COMMAND\stanza@hang	344
COMMAND\stanza@line	344
COMMAND\stanzaindent	59, 342, 473
COMMAND\stanzaindent*	59
COMMAND\stanzaindentbase	341
COMMAND\stanzanumwrapper	61
COMMAND\startlock	22, 24, 113, 137, 345
COMMAND\startstanzahook	405
COMMAND\startsub	23, 113, 136
COMMAND\stopmsd	351

COMMAND\stopmsdata	39, 349
COMMAND\store@annot@to@absline	173
COMMAND\strip@pt	196
COMMAND\strutbox	204
COMMAND\sub@action	113, 124
COMMAND\sub@lock	111
COMMAND\sub@off	121, 301
COMMAND\sub@on	121, 301
COMMAND\subline@num	110–112
COMMAND\sublinenum@rep	465
COMMAND\sublinenumberstyle	24, 108, 465
COMMAND\sublinenumincrement	22
COMMAND\sublinenumr@p	108, 465, 467
COMMAND\sublinenumrep	108, 467
COMMAND\sublineref	63, 297, 304
COMMAND\subsectionmark	382
COMMAND\sw@inthisedtext	142
COMMAND\sw@list@inedtext	147, 154
COMMAND\sw@txt	151
COMMAND\swnoexpands	37, 483
COMMAND\symlinenum	406
COMMAND\symplinenum	405
COMMAND\sza@penalty	344
COMMAND>tag	473
COMMAND\text	400
COMMAND\text<language>	52
COMMAND\textbeforenotesX	184
COMMAND\textcolor	81
COMMAND\textheight	80
COMMAND\textsuperscript	27, 94, 173
COMMAND\the	132, 465
COMMAND\thefootnoteA	40
COMMAND\thefootnoteX	469
COMMAND\thelabidx	334, 335
COMMAND\thepage	118, 132, 295
COMMAND\thepstart	20, 409
COMMAND\thepstartL	468
COMMAND\thepstartR	468
COMMAND\thestanza	61
COMMAND\this@crossref@end	225
COMMAND\this@line@list@version	131
COMMAND\this@section@next@page@num	100
COMMAND\this@section@page@num	114
COMMAND\thisfootnote	234
COMMAND\threecol@begin@insert	205
COMMAND\threecolfootfmt	204, 205, 480
COMMAND\threecolfootfmtX	239
COMMAND\threecolfootgroup	203
COMMAND\threecolfootgroupX	240
COMMAND\threecolfootsetup	203

COMMAND\threecolfootsetupX	239
COMMAND\threecolvfootnote	204
COMMAND\threecolvfootnote@inserted	204
COMMAND\threecolvfootnoteX	239
COMMAND\toendnotes	31, 255, 480
COMMAND\twocolfootfmt	480
COMMAND\twocolfootfmtX	237
COMMAND\twocolfootgroupX	237
COMMAND\twocolfootsetupX	236
COMMAND\twocolvfootnoteX	237
COMMAND\twolines	279, 406
COMMAND\twolines@A	279
COMMAND\twolines@B	279
COMMAND\twolines@C	279
COMMAND\twolinesbutnotmore	406
COMMAND\twolinesonlyinsamepage	406
COMMAND\txtbeforeXnotes	406
COMMAND\txtbeforenotesX	54, 483
COMMAND\txtbeforenotesonlyonceX	54, 483
COMMAND\uline	41
COMMAND\unhbox	162, 197
COMMAND\unpenalty	198–200
COMMAND\unskip	199
COMMAND\unvxh	199, 407
COMMAND\unvxhX	407
COMMAND\upbracefill	375
COMMAND\usingcritext	401, 404
COMMAND\usingdtext	401, 404
COMMAND\vAfootnote	188
COMMAND\vadjust	134
COMMAND\variant	33
COMMAND\ vbox	159, 161, 197, 202, 248
COMMAND\vfootnote	188, 193, 197, 204
COMMAND\vl@dbfnote	227, 466
COMMAND\vl@disnote	319
COMMAND\vl@dlsnote	319
COMMAND\vl@dosnote	319
COMMAND\vl@drsnote	319
COMMAND\vnumfootnoteX	467
COMMAND\vsizer	55, 80
COMMAND\vsplit	182
COMMAND\waklam	375
COMMAND\waklamec	375
COMMAND\wapunktel	375
COMMAND\wastricht	375
COMMAND\widthX	56, 408, 479
COMMAND\wrap@edcrossref	303, 472
COMMAND\wrapcontentX	52, 480
COMMAND\wrapped@bodyfootmarkX	245
COMMAND\wrapped@footfootmarkX	245

COMMAND\x...	64
COMMAND\xabslineref	304
COMMAND\xannotationref	64
COMMAND\xdef	103, 345
COMMAND\xflagref	64, 305, 408, 478
COMMAND\xleft@appenditem	103, 139
COMMAND\xlineref	64, 408, 478
COMMAND\xpageref	64
COMMAND\xparse	38
COMMAND\xpstartref	64, 470, 483
COMMAND\xr	68
COMMAND\xright@appenditem	103
COMMAND\xsublineref	64
COMMAND\xxref	64, 306, 313, 470, 473, 480
COMMAND\xxxmatter	295
ENVIRONMENTastanza	97, 478
ENVIRONMENTedarrayc	380
ENVIRONMENTedarrayl	380
ENVIRONMENTedarrayr	380
ENVIRONMENTedtabular	485
ENVIRONMENTedtabularc	381
ENVIRONMENTedtabularl	381
ENVIRONMENTedtabularr	381
ENVIRONMENTledgroup	86, 325, 408, 478
ENVIRONMENTledgroupsize	326
PACKAGE(r)eledmac	42
PACKAGEEledmac	112, 331, 403, 404, 473, 474
PACKAGEEledpar	474
PACKAGEEtoolbox	85
PACKAGEParallel	412
PACKAGEReledmac	61, 407, 408
PACKAGEamsgen	358
PACKAGEamsmath	358
PACKAGEbabel	52, 81, 369, 478
PACKAGEbiblatex	79
PACKAGEbidi	52, 53, 87, 477, 485
PACKAGEcalc	486
PACKAGEccaption	95
PACKAGEcolor	81
PACKAGEcsquotes	482
PACKAGEedmac	1, 6, 11, 13–15, 82, 216, 225, 298, 341, 400, 412, 465
PACKAGEedstanza	1, 14, 339
PACKAGEeledmac	1, 11, 15, 16, 71, 225, 327, 331, 362, 384, 397, 401, 403–405, 469, 471, 473
PACKAGEeledpar	188, 412, 467, 471–473
PACKAGEetex	477
PACKAGEetoolbox	102, 149, 264, 279, 289, 318, 383, 394
PACKAGEfancyhdr	292, 293, 481
PACKAGEfloatrow	79, 408
PACKAGEfootmisc	41, 81, 87, 225, 226, 412, 481
PACKAGEgeometry	16

PACKAGEglossaries	72, 336, 478
PACKAGEhandout	472
PACKAGEhyperlink	273
PACKAGEhyperref	64, 142, 245, 298, 333, 334, 387, 397, 470–472, 479, 482, 485
PACKAGEifluatex	85
PACKAGEifxetex	85
PACKAGEimakeidx	70, 79, 86, 87, 327, 331, 404, 469–471, 473
PACKAGEindextols	336
PACKAGEindextool	404
PACKAGEindextools	70, 79, 86, 87, 96, 327, 331, 336, 404, 473, 478, 481
PACKAGEledarab	81
PACKAGEledmac	1, 11, 14, 15, 81, 102, 331, 400, 401, 404, 407
PACKAGEledpar	81
PACKAGEMemoir	86, 331, 404, 412, 471
PACKAGEMorewrites	79
PACKAGEMusixtex	471
PACKAGEperpage	478
PACKAGEpolyglossia	50, 81, 187, 210, 478, 485
PACKAGERagged2e	53, 85
PACKAGEReledmac	1, 2, 6, 11–13, 15–18, 21, 22, 25, 28–30, 32–35, 38, 41, 44–46, 49, 52, 53, 55, 57, 60, 62, 64–66, 68, 70–73, 77, 79–83, 104, 106, 112, 113, 116, 117, 121, 130–132, 140, 143, 145, 148, 149, 180, 189, 193, 198, 210, 225, 226, 251, 265, 270, 279, 289, 295, 303, 304, 307, 313, 314, 331, 354, 358, 362, 382, 383, 396, 397, 404, 405, 407–410, 476, 479, 481
PACKAGEReledpar	1, 4, 6, 9, 17, 20, 22, 41, 56, 64, 66, 67, 77–79, 81, 83, 95, 104, 115, 121, 131, 132, 143, 145, 156, 190, 193, 246, 247, 265, 271, 288, 290, 291, 295, 327, 339, 382, 477, 478, 480, 481, 483, 484
PACKAGESuffix	85
PACKAGETabmac	1, 14, 412
PACKAGETHALIE	62
PACKAGEulem	41
PACKAGEuninormalize	34
PACKAGExargs	33, 85
PACKAGExkeyval	83, 288
PACKAGExparse	85, 484
PACKAGExr	5, 68, 314, 479
PACKAGExref	313
PACKAGExstring	85, 334

## A

\absline@num	1
Abu Kamil Shuja' b. Aslam	14
\actionlines@list	1
\actions@list	1
\add@hookarg@specific@to@cs	1
\add@hooktoggle@specific@to@cs	1
\add@inserts	1
\add@inserts@next	1
\add@msd@	1
\add@msdata	1
\add@msdata@firstlineofpage	1



<code>\add@penalties</code> .....	1
<code>\add@Xgroupbyline</code> .....	1
<code>\addtol@denbody</code> .....	1
Adelard II .....	14
<code>\advancelabel@refs</code> .....	1
<code>\advanceline</code> .....	1, 24
<code>\Aendnote</code> .....	30
<code>\affixline@num</code> .....	1
<code>\affixpstart@num</code> .....	1
<code>\affixside@note</code> .....	1
<code>\Afootnote</code> .....	30
<code>\afterantilabe</code> .....	62
<code>\afternoteX</code> .....	53
<code>\afterruleX</code> .....	55
<code>\ampersand</code> .....	1, 62
<code>\annot@list</code> .....	1
<code>\annotationref</code> .....	1, 63
<code>\antilabe</code> .....	1, 61
<code>\append@notesX</code> .....	1
<code>\append@Xnotes</code> .....	1
<code>\applabel</code> .....	1, 66
<code>\appref</code> .....	1, 66
<code>\apprefwithpage</code> .....	1, 66
<code>\arrangementX</code> .....	1, 43
<code>\arrangementX@normal</code> .....	1
<code>\arrangementX@paragraph</code> .....	1
<code>\arrangementX@threecol</code> .....	1
<code>\arrangementX@twocol</code> .....	1
<code>\at@every@pend</code> .....	1
<code>\AtEndEveryPend</code> .....	1, 20
<code>\AtEveryPend</code> .....	1, 20
<code>\AtEveryPstart</code> .....	1, 20
<code>\AtEveryStanza</code> .....	1, 60
<code>\AtEveryStopStanza</code> .....	1, 61
<code>\AtStartEveryPstart</code> .....	1, 20
<code>\AtStartEveryStanza</code> .....	1, 60
<code>\autopar</code> .....	1, 19

## B

<code>\ballast</code> .....	80
<code>\ballast@count</code> .....	1
Beeton, Barbara Ann Neuhaus Friend .....	19
<code>\beforeantilabe</code> .....	62
<code>\beforeeledchapter</code> .....	1
<code>\BeforeEveryStopStanza</code> .....	1, 61
<code>\beforeinsertingX</code> .....	52
<code>\beforeinsertion@X</code> .....	1
<code>\beforenotesX</code> .....	55
<code>\beginnumbering</code> .....	1, 18
<code>\Bendnote</code> .....	30

<code>\Bfootnote</code> .....	30
<code>\bhookgroupX</code> .....	55
<code>\bhooknoteX</code> .....	52
<code>\bodyfootmarkA</code> .....	40
<code>\boxfootnotenumbers</code> .....	1
Bredon, Simon .....	14
Breger, Herbert .....	14, 362
Brey, Gerhard .....	14
Busard, Hubert L. L. ....	14
<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> .....	30
<code>\Cfootnote</code> .....	30
<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 .....	14
Claassens, Geert H. M. ....	14
<code>\colalignX</code> .....	53
<code>\continuousnumberingwithcolumns@sync@linenumber@singletext</code> .....	1
Copernicus, Nicolaus .....	14
<code>\critext</code> .....	400
<code>\ctab</code> .....	1
<code>\ctabtext</code> .....	1
<code>\current@annot</code> .....	1

## D

Dekker, Dirk-Jan .....	81
<code>\Dendnote</code> .....	30
<code>\Dfootnote</code> .....	30
<code>\disable@endnotes</code> .....	1
<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, 31
<code>\doendnotesbysection</code> .....	1, 31
<code>\doinsidelinehook</code> .....	1, 28
<code>\doinsidethislinehook</code> .....	1, 28
<code>\dolinehook</code> .....	1, 28
<code>\dosplits</code> .....	1
Downes, Michael .....	80, 197, 199
<code>\dummy@edindex</code> .....	1
<code>\dummy@edtext</code> .....	1
<code>\dummy@edtext@showlemma</code> .....	1
<code>\dummy@ref</code> .....	1

## E

<code>\edaftertab</code> .....	1, 75, 375
<code>edarrayc</code> (environment) .....	73
<code>edarrayl</code> (environment) .....	73
<code>edarrayr</code> (environment) .....	73
<code>\edatleft</code> .....	1, 75
<code>\edatright</code> .....	1, 75
<code>\edbeforetab</code> .....	1, 75, 375
<code>\edfilldimen</code> .....	1
<code>\edfont@info</code> .....	1
<code>\edindex</code> .....	1, 70
<code>\edindexlab</code> .....	1, 72
<code>\EDLABEL</code> .....	1
<code>\edlabel</code> .....	1, 63
<code>\edlabelE</code> .....	1, 65
<code>\edlabelS</code> .....	1, 65
<code>\edlabelSE</code> .....	1, 65
<code>\edlineref</code> .....	1, 63
<code>\edmakelabel</code> .....	1, 65
<code>\edpageref</code> .....	1, 63
<code>\edrowfill</code> .....	1, 74
<code>\EDTAB</code> .....	1
<code>\edtabcolsep</code> .....	1, 73
<code>\EDTABINDENT</code> .....	1
<code>\edtabindent</code> .....	1
<code>\EDTABtext</code> .....	1

edtabularc (environment)	73
edtabularl (environment)	73
edtabularr (environment)	73
\EDTEXT	1
\edtext	1, 29
\edvertdots	1, 75
\edvertline	1, 75
\Endnote	30
\Footnote	30
\eled@chapter	1
\eled@section	1
\eled@sectioning@out	1
\eled@subsection	1
\eled@subsubsection	1
\eledchapter	1
\eledchapter*	1
\eledsection	1
\eledsection*	1
\eledsubsection	1
\eledsubsection*	1
\eledsubsubsection	1
\eledsubsubsection*	1
\enablel@dtabfeet	1
\end@lemmas	1
\endashchar	1
\endline@num	1
\endlock	1, 24
\endminipage	1
\endnumbering	1, 18
\endpage@num	1
\endprint	1
\endquotation	1
\endquote	1
\endsub	1, 23
\endsubline@num	1
environments:	
edarrayc	73
edarrayl	73
edarrayr	73
edtabularc	73
edtabularl	73
edtabularr	73
ledgroup	62
ledgroupsize	63
minipage	62
Euclid	14
\extensionchars	1, 78
<b>F</b>	
\f@x@l@cks	1

Fairbairns, Robin	41
\first@linenum@out@false	1
\first@linenum@out@true	1
\firsthalf@antilabe@wd	1
\firstlinenum	1, 22
\firstseriesX@	1
\firstsublinenum	1, 22
\firstXseries@	1
\flag@end	1
\flag@end@later	1
\flag@end@RTL	1
\flag@start	1
\flag@start@later	1
\flag@start@RTL	1
\flagstanza	1, 62
\flush@notes	1
\fnpos	1, 42
Folkerts, Menso	14
\footfootmarkA	40
\footfudgefiddle	1, 80
\footnote	1
\footnoteA	40
\footnoteB	40
\footnoteC	40
\footnoteD	40
\footnoteE	40
\footnotelang@lua	1
\footnotelang@poly	1
\footnoteoptions@	1
\footnoteXmark	41
\footnoteXtext	41
\footsplitskips	1
\fullstop	1

## G

Gädeke, Nora	14
\get@edindex@hyperref	1
\get@edindex@ledinnote@command	1
\get@firsthalf@antilabe@wd	1
\get@fnmark	1
\get@fnmarkX	1
\get@index@command	1
\get@linelistfile	1
\get@pstartcounter	1
\get@stanzacounter	1
\get@sw@txt	1
\get@thisfootnote	1
\get@thisfootnoteX	1
\getline@num	1
\gl@p	1

**H**

<code>\h@num</code> .....	1
<code>\hangindentX</code> .....	52
<code>\hidenumbering</code> .....	1, 25
<code>\hidenumberingonleftpage</code> .....	1, 25
<code>\hidenumberingonrightpage</code> .....	1
<code>\Hilfsbox</code> .....	1
<code>\hilfsbox</code> .....	1
<code>\hilfscount</code> .....	1
<code>\HILFSskip</code> .....	1
<code>\Hilfsskip</code> .....	1
<code>\hilfsskip</code> .....	1
<code>\hookarg@specific</code> .....	1
<code>\hooktoggle@specific</code> .....	1
<code>\hsizethreecolX</code> .....	53
<code>\hsizetwocolX</code> .....	53
<code>\Hy@raisedlink@left</code> .....	1
<code>\hyperlinkformat</code> .....	1
<code>\hyperlinkformatR</code> .....	1
<code>\hyperlinkR</code> .....	1

**I**

<code>\if@addsw</code> .....	1
<code>\if@edtabular</code> .....	1
<code>\if@edtext@secondarg@</code> .....	1
<code>\if@eled@sectioning</code> .....	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>\ifafterendnumberingR</code> .....	1
<code>\ifantilabe@</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>\ifeledmacompat@</code> .....	1
<code>\iffirst@linenum@out@</code> .....	1
<code>\ifinastanzaL</code> .....	1
<code>\ifinastanzaR</code> .....	1
<code>\ifindtl@innote</code> .....	1
<code>\ifindtl@notenumber</code> .....	1
<code>\ifinserthangingsymbol</code> .....	1

<code>\ifistanza</code> .....	1
<code>\ifistanzaL</code> .....	1
<code>\ifistanzaR</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@Xmorethantwoline</code> .....	1
<code>\ifl@d@Xtwoline</code> .....	1
<code>\ifl@dend@X</code> .....	1
<code>\ifl@dhidenumber</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, 78
<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>\ifnoresetlinenumannotation@</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>\ifprevpgnotnumbered</code> .....	1
<code>\ifpst@rtedL</code> .....	1
<code>\ifresumenumbering@start</code> .....	1
<code>\ifsameparallelpagenumber</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>\innotemarkX</code> .....	70
<code>\insert@count</code> .....	0, 1
<code>\insert@msdata</code> .....	1
<code>\insert@txtbeforenotesX</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 .....	14
-----------------	----

## K

Kabelschacht, Alois .....	184
<code>\keep@this@crossref@forinserting</code> .....	1

## 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@dcolwidth</code> .....	1
<code>\l@dcsnote</code> .....	1
<code>\l@dcsnotetext</code> .....	1
<code>\l@dcsnotetext@l</code> .....	1
<code>\l@dcsnotetext@r</code> .....	1
<code>\l@ddodoreinextrafeet</code> .....	1
<code>\l@ddoxtrafeet</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@dfetbeginmini</code>	1
<code>\l@dfetendmini</code>	1
<code>\l@getline@margin</code>	1
<code>\l@getlock@disp</code>	1
<code>\l@getref@num</code>	1
<code>\l@getsidenote@margin</code>	1
<code>\l@gobbleoptarg</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@nullfills</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@dstoreforedtext</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>\l@noexpands@edgl</code>	1
<code>\l@secondmandarg</code>	1
<code>\l@wrap@ifnotemptybox</code>	1
<code>\l@wrapcs@ifnotemptybox</code>	1
<code>\label</code>	65
<code>\labelpstartfalse</code>	1
<code>\labelpstarttrue</code>	1, 20
<code>\labelref@list</code>	1
<code>\labelrefsparseline</code>	1
<code>\labelrefsparsesubline</code>	1
<code>\last@page@num</code>	1
Lavagnino, John	13
<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@PendInEdtext</code>	1
<code>\led@err@PendNoPstart</code>	1
<code>\led@err@PendNotNumbered</code>	1
<code>\led@err@PstartInEdtext</code>	1
<code>\led@err@PstartInPstart</code>	1
<code>\led@err@PstartNotNumbered</code>	1
<code>\led@err@ReverseColumns</code>	1
<code>\led@err@samewordRTL</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@addtocounter</code>	1
<code>\led@error@fail@patch@endminipage</code>	1
<code>\led@error@fail@patch@setcounter</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@BadHookSeries</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>	62
<code>ledgroupsize (environment)</code>	63
<code>\ledinnernote</code>	1, 68
<code>\ledinnote</code>	1
<code>\ledinnotemark</code>	1
<code>\ledleftnote</code>	1, 68
<code>\ledlinenum</code>	1
<code>\ledllfill</code>	1
<code>\ledlsnotefontsetup</code>	1, 69
<code>\ledlsnotesep</code>	1, 69
<code>\ledlsnotewidth</code>	1, 69
<code>\lednopb</code>	1, 77
<code>\lednopbinversetrue</code>	78
<code>\lednopbnum</code>	1
<code>\ledouternote</code>	1, 68
<code>\ledpb</code>	1, 77
<code>\ledpbnum</code>	1

<code>\ledpbsetting</code>	1, 78
<code>\ledrightnote</code>	1, 68
<code>\ledrlfill</code>	1
<code>\ledrsnotefontsetup</code>	1, 69
<code>\ledrsnotesep</code>	1, 69
<code>\ledrsnotewidth</code>	1, 69
<code>\ledsectnomark</code>	1
<code>\ledsectnotoc</code>	1
<code>\ledsetnormalparstuff@common</code>	1
<code>\ledsetnormalparstuffX</code>	1
<code>\ledsidenote</code>	1, 68
<code>\leftctab</code>	1
<code>\leftlinenum</code>	1, 23
<code>\leftltab</code>	1
<code>\leftnoteupfalse</code>	69
<code>\leftpstartnum</code>	1
<code>\leftrtab</code>	1
Leibniz	14
<code>\lemma</code>	1, 32
<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>\lineannot</code>	1, 26
<code>\lineannot@set</code>	1
<code>\lineation</code>	1, 22
<code>\linenum</code>	1, 32
<code>\linenum@out</code>	1
<code>\linenumannotation</code>	1, 25
<code>\linenumberlist</code>	1, 22
<code>\linenumberstyle</code>	1, 24
<code>\linenumincrement</code>	1, 22
<code>\linenummargin</code>	1, 23
<code>\linenumr@p</code>	1
<code>\linenumrep</code>	1
<code>\linenumsep</code>	1, 23
<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, 24
Lorch, Richard	14
<code>\ltab</code>	1
<code>\ltabtext</code>	1

Luecking, Dan . . . . . 85

## M

<code>\m@mmf@check</code> . . . . .	1
<code>\m@mmf@prepare</code> . . . . .	1
<code>\M@sect</code> . . . . .	1
<code>\makehboxofhboxes</code> . . . . .	1
<code>\managestanza@modulo</code> . . . . .	1
<code>\maxhnotesX</code> . . . . .	55
Mayer, Gyula . . . . .	14
<code>\measurembody</code> . . . . .	1
<code>\measuremcell</code> . . . . .	1
<code>\measuremrow</code> . . . . .	1
<code>\measuretbody</code> . . . . .	1
<code>\measuretcell</code> . . . . .	1
<code>\measuretrow</code> . . . . .	1
Middleton, Thomas . . . . .	14, 111
<code>minipage</code> (environment) . . . . .	62
Mittelbach, Frank . . . . .	13, 14
<code>\morenoexpands</code> . . . . .	1, 80
<code>\mp@append@notesX</code> . . . . .	1
<code>\mp@append@Xnotes</code> . . . . .	1
<code>\mpfnpos</code> . . . . .	1, 42
<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, 39
<code>\msdata@c</code> . . . . .	1
<code>\msdata@cR</code> . . . . .	1
<code>\multfootsep</code> . . . . .	1, 41
<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>\next@line@list@stuff</code> .....	1
<code>\no@expands</code> .....	1
<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> .....	51
<code>\notenumfontX</code> .....	51
<code>\noteschanged@false</code> .....	1
<code>\noteschanged@true</code> .....	1
<code>\noteswidthliketwocolumnsX</code> .....	56
<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> .....	21
<code>\numberlinetrue</code> .....	21
<code>\numberpstartfalse</code> .....	1, 20
<code>\numberpstarttrue</code> .....	1, 20
<code>\numberstanzafalse</code> .....	61
<code>\numberstanzatrue</code> .....	61
<code>\numlabfont</code> .....	1, 57

## O

<code>\old@hsize</code> .....	1
<code>\one@line</code> .....	1
<code>optionafter</code> .....	27
<code>optionantilabe</code> .....	61, 347
<code>optionauxdir</code> .....	16, 480
<code>optionbefore</code> .....	27
<code>optioncontinuousnumberingwithcolumns</code> .....	96, 164, 171, 479, 484
<code>optioninnnote</code> .....	478
<code>optioninnote</code> .....	478
<code>optionlinrangesep</code> .....	288
<code>optionnocritical</code> .....	478
<code>optionnoeledsec</code> .....	77, 388, 480
<code>optionnoend</code> .....	478
<code>optionnopentialties</code> .....	80
<code>optionnoprefix</code> .....	68, 485
<code>optionnoresetlinenumannotation</code> .....	25, 44, 484
<code>optionnotenumber</code> .....	478
<code>optionprefixmore</code> .....	68, 485
<code>optionsameparallelpagenum</code> .....	484
<code>optionswcaseinsensitive</code> .....	34, 483

## P

<code>\page@action</code> .....	1
<code>\page@num</code> .....	1
<code>\pagelinesep</code> .....	1, 71
<code>\pageref</code> .....	65
<code>\par@line</code> .....	1
<code>\par@patch@pagenumbering</code> .....	1
<code>\par@patch@thepage</code> .....	1
<code>\para@footgroupX</code> .....	1
<code>\para@footsetup</code> .....	1
<code>\para@footsetupX</code> .....	1
<code>\para@vfootnoteX</code> .....	1
<code>\parafootfmt</code> .....	1
<code>\parafootfmtX</code> .....	1
<code>\parafootgroup</code> .....	1
<code>\parafootsepX</code> .....	54
<code>\parafootstart</code> .....	1
<code>\parafootstartX</code> .....	1
<code>\paravfootnote</code> .....	1
<code>\parindentX</code> .....	52
<code>\parse@annot</code> .....	1
<code>\pausenumbering</code> .....	1, 21
<code>\pausenumbering@page@num</code> .....	1
<code>\pend</code> .....	1, 18
<code>Plato of Tivoli</code> .....	14
<code>\postbodyfootmark</code> .....	1
<code>\prebodyfootmark</code> .....	1
<code>\prenotesX</code> .....	55

<code>\prepare@edindex@fornote</code> .....	1
<code>\prepare@prenotesX</code> .....	1
<code>\prepare@Xgroupbyline</code> .....	1
<code>\prepare@Xprenotes</code> .....	1
<code>\prev@ncpb</code> .....	1
<code>\prev@pb</code> .....	1
<code>\prevpage@num</code> .....	1
<code>\print@eledsection</code> .....	1
<code>\print@footnoteXrule</code> .....	1
<code>\print@leftmargin@eledsection</code> .....	1
<code>\print@lemma</code> .....	1
<code>\print@line</code> .....	1
<code>\print@notesX</code> .....	1
<code>\print@rightmargin@eledsection</code> .....	1
<code>\print@Xfootnoterule</code> .....	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>\printlinefootnotenumbers</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, 18
<code>\pstarteref</code> .....	1
<code>\pstartnum</code> .....	1
<code>\pstartref</code> .....	63
<code>\pstarts@read@L</code> .....	1
<code>\pstarts@typeset@L</code> .....	1

## Q

<code>\quotation</code> .....	1
<code>\quote</code> .....	1

## R

<code>\raggedX</code> .....	54
<code>\raw@text</code> .....	1
<code>\rbracket</code> .....	1
<code>\read@linelist</code> .....	1
<code>\ref</code> .....	65
<code>\ref@reg@later</code> .....	1
<code>\Relax</code> .....	1
<code>\reledmac@error</code> .....	1
<code>\reledmac@warning</code> .....	1



<code>\removeboxes</code> .....	1
<code>\reset@current@annot</code> .....	1
<code>\reset@msd@options@iffullpage</code> .....	1
<code>\resetprevline@</code> .....	1, 114
<code>\resetprevpage@</code> .....	1
<code>\resetprevpage@num</code> .....	114
<code>\restore@endnotes</code> .....	1
<code>\restore@familiarnotes</code> .....	1
<code>\restore@notes</code> .....	1
<code>\restore@sidenotes</code> .....	1
<code>\resumenumbering</code> .....	1, 21
<code>\rightctab</code> .....	1
<code>\rightlinenum</code> .....	1, 23
<code>\rightltab</code> .....	1
<code>\rightnoteupfalse</code> .....	69
<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 .....	14
<code>\sameword</code> .....	1, 33
<code>\sameword@inedtext</code> .....	1
Schöpf, Rainer .....	14
<code>\section@num</code> .....	1
<code>\select@@lemmafont</code> .....	1
<code>\select@lemmafont</code> .....	1, 57
<code>\SEref</code> .....	1, 65
<code>\SErefonlypage</code> .....	65
<code>\SErefwithpage</code> .....	1, 65
<code>\series</code> .....	1
<code>\seriesatbegin</code> .....	1, 42
<code>\seriesatend</code> .....	1, 42
<code>\set@continuousnumberingforL</code> .....	1
<code>\set@line</code> .....	1
<code>\set@line@action</code> .....	1
<code>\set@this@c@page</code> .....	1
<code>\set@txtbeforenotesX</code> .....	1
<code>\set@Xtxtbeforenotes</code> .....	1
<code>\setapprefprefixmore</code> .....	66
<code>\setapprefprefixsingle</code> .....	66
<code>\setcommand@series</code> .....	1
<code>\sethangindentX</code> .....	1
<code>\sethangingsymbol</code> .....	1, 60
<code>\setistwofollowinglines</code> .....	1
<code>\setl@dlp@rbox</code> .....	1
<code>\setl@drpr@box</code> .....	1

<code>\setline</code>	1, 24
<code>\setlinenum</code>	1, 24
<code>\setlinenumannotationsep</code>	1, 28
<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, 40
<code>\setmsdataposition</code>	1, 40
<code>\setmsdataseries</code>	1, 40
<code>\setnoteswidthliketwocolumnsX@</code>	1
<code>\setnotesXpositionliketwocolumns@</code>	1
<code>\setparindentX</code>	1
<code>\setprintendlines</code>	1
<code>\setprintlines</code>	1
<code>\setSErefonlypageprefixmore</code>	66
<code>\setSErefonlypageprefixsingle</code>	66
<code>\setSErefprefixmore</code>	66
<code>\setSErefprefixsingle</code>	66
<code>\setsidenotessep</code>	69
<code>\setstanzaindents</code>	1, 57
<code>\setstanzapenalties</code>	1, 59
<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, 78
<code>\showwordrank</code>	1, 37
<code>\sidenote@margin</code>	1
<code>\sidenotemargin</code>	1, 68
<code>\sidepstartnumtrue</code>	20
<code>\skip@lockoff</code>	1
<code>\skipnumbering</code>	1, 25
<code>\splitoff</code>	1
<code>\spreadmath</code>	1, 74
<code>\spreadtext</code>	1, 74
<code>\stanza</code>	1, 57
<code>\stanza@count</code>	1
<code>\stanza@hang</code>	1
<code>\stanza@indent</code>	1
<code>\stanza@line</code>	1
<code>\stanzaindent</code>	1, 59

<code>\stanzaindent*</code>	1, 59
<code>\stanzaindentbase</code>	1, 57
<code>\stanzanumwrapper</code>	1, 61
<code>\startlock</code>	1, 24
<code>\startsub</code>	1, 23
<code>\stepl@dcolcount</code>	1
<code>\stopmsdata</code>	1, 39
<code>\store@cannot@to@absline</code>	1
<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, 24
<code>\sublinenumincrement</code>	1, 22
<code>\sublinenumr@p</code>	1
<code>\sublinenumrep</code>	1
<code>\sublineref</code>	1, 63
<code>\sublines@false</code>	1
<code>\sublines@true</code>	1
<code>\sublock@disp</code>	1
<code>\sublockdisp</code>	1
Sullivan, Wayne	14, 80, 95, 100, 197, 198, 297, 339
<code>\sw@noexpand</code>	1
<code>\swnoexpands</code>	37
<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>	40
Theodosius	14
<code>\thepageline</code>	1
<code>\thepar@page</code>	1
<code>\thepstart</code>	1, 20
<code>\thestanza</code>	1, 61
<code>\thestartpageline</code>	1
<code>\this@line@list@version</code>	1
<code>\this@section@last@page@num</code>	1
<code>\this@section@page@num</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, 31
<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
<code>\txtbeforenotesonlyonceX</code> .....	54
<code>\txtbeforenotesX</code> .....	54

## U

<code>\unvxhX</code> .....	1
----------------------------	---

## V

Vamana .....	14
<code>\variab</code> .....	1
<code>\vbfnoteX</code> .....	1
<code>\vl@dbfnote</code> .....	1
<code>\vl@dcnote</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 .....	13
<code>\widthX</code> .....	56
<code>\wrap@edcrossref</code> .....	1
<code>\wrapcontentX</code> .....	52
<code>\wrapped@bodyfootmarkX</code> .....	1
<code>\wrapped@footfootmarkX</code> .....	1
Wujastyk, Dominik .....	13

## X

<code>\X@atbegininsertion</code> .....	1
<code>\X@beforeinsertion</code> .....	1
<code>\X@doreinfeet</code> .....	1
<code>\xabslineref</code> .....	1
<code>\Xafterlemmaseparator</code> .....	50
<code>\Xafternote</code> .....	53
<code>\Xafternumber</code> .....	48

\Xafterrule .....	55
\Xaftersymlinenum .....	48
\xannotationref .....	1, 64
\Xarrangement .....	1, 43
\Xarrangement@normal .....	1
\Xarrangement@paragraph .....	1
\Xarrangement@threecol .....	1
\Xarrangement@twocol .....	1
\Xbeforeinserting .....	52
\Xbeforelemmaseparator .....	50
\Xbeforenotes .....	55
\Xbeforenumber .....	48
\Xbeforesymlinenum .....	48
\Xbhookgroup .....	55
\Xbhooknote .....	52
\Xboxlinenum .....	48
\Xboxlinenumalign .....	49
\Xboxsymlinenum .....	49
\Xcolalign .....	53
\xedindex .....	1
\xedlabel .....	1
\xedtext .....	1
\Xendafterlemmaseparator .....	50
\Xendafternote .....	56
\Xendafterpagenumber .....	46
\Xendaftersymlinenum .....	48
\Xendaftrenumber .....	48
\Xendahookinplaceofnumber .....	50
\Xendahooklinenumber .....	49
\Xendbeforelemmaseparator .....	50
\Xendbeforenumber .....	48
\Xendbeforepagenumber .....	46
\Xendbeforesymlinenum .....	48
\Xendbhookinplaceofnumber .....	50
\Xendbhooklinenumber .....	49
\Xendbhooknote .....	52
\Xendboxendlinenumalign .....	49
\Xendboxlinenum .....	49
\Xendboxlinenumalign .....	49
\Xendboxstartlinenumalign .....	49
\Xendboxsymlinenum .....	49
\Xendhangindent .....	52
\Xendinplaceofflemmaseparator .....	50
\Xendinplaceofnumber .....	48
\Xendinplaceofpagenumber .....	45
\Xendlemmadisablefontselection .....	51
\Xendlemmafont .....	51
\Xendlemmaseparator .....	50
\Xendlinenumannotationonlyfirst .....	45
\Xendlinenumannotationonlyfirstintwo .....	45

\Xendlinenumannotationposition	26
\Xendlineprefixmore	46
\Xendlineprefixsingle	46
\Xendlinerangeseparator	46
\Xendmorethantwolines	47
\Xendnoidenticallylinenumannotation	28
\Xendnonumber	47
\Xendnotefontsize	51
\Xendnotenumfont	51
\Xendnumberonlyfirstinline	44
\Xendnumberonlyfirstintwolines	44
\Xendpagenumberonlyfirst	45
\Xendparagraph	56
\Xendsep	56
\Xendstorelineinfo	1
\Xendsublinesep	48
\Xendsymlinenum	44
\Xendsymlinenumannotation	45
\Xendsympagenum	45
\Xendtwolines	47
\Xendtwolinesbutnotmore	47
\Xendtxtbeforenotes	54
\Xendwrapcontent	52
\xflagref	1
\Xgroupbyline	54
\Xgroupbylineseparetwolines	54
\Xhangindent	52
\Xhsizethreecol	53
\Xhsizetwocol	53
\Xinnotemark	70
\Xinplaceoflemmaseparator	50
\Xinplaceofnumber	48
\Xinsertparafootsep	1
\Xledsetnormalparstuff	1
\xleft@appenditem	1
\Xlemmadisablefontselection	51
\Xlemmafont	51
\Xlemmaseparator	50
\Xlinenumannotationonlyfirst	44
\Xlinenumannotationonlyfirstintwo	44
\Xlinenumannotationposition	26
\Xlinenumannotationposition@side	1
\Xlinerrangeseparator	46
\xlineref	1, 64
\Xmaxhnotes	55
\Xmorethantwolines	46
\Xnoidenticallylinenumannotation	28
\Xnolemmaseparator	1, 50
\Xnonbreakableafternumber	48
\Xnonumber	47

\Xnotefontsize	51
\Xnotenumfont	51
\Xnoteswidthliketwocolumns	56
\Xnumberonlyfirstinline	44
\Xnumberonlyfirstintwolines	44
\Xonlypstart	47
\Xpagelinesep	48
\xpageref	1, 64
\Xparafootsep	54
\Xparindent	52
\Xprenotes	1, 55
\Xprenotes@	1
\Xpstart	47
\Xpstarteverytime	47
\Xpstartonlyfirst	47
\xpstartref	1, 64
\Xpstartseparator	47
\XR@test	1
\XR@test@mac	1
\XR@test@mac@test	1
\Xragged	54
\xright@appenditem	1
\Xrigidbalance	1
\Xsethangindent	1
\Xsetparindent	1
\Xstanza	47
\Xstanzaonlyfirst	47
\Xstanzaseparator	47
\Xstorelineinfo	1
\xsublineref	1, 64
\Xsublinesep	23, 47
\Xsublineleftside	23
\Xsymlinenum	44
\Xsymlinenumannotation	45
\Xtoendnotes	31
\Xtwolines	46
\Xtwolinesonlyinsamepage	46
\Xtxtbeforenotes	54
\Xtxtbeforenotesonlyonce	54
\Xtxtbeforenumber	46
\Xunvxh	1
\Xwidth	56
\Xwrapcontent	52
\Xwrapendlemma	52
\Xwraplemma	51
\Xwraplinenumannotation@ref	1
\Xwraplinenumannotation@side	1
\xxref	1, 64

**Z**

\zz@@@	1
\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	86
\morenoexpands: Added \l@dtabnoexpands to \no@expands	141
\reledmac@error: Added \eledmac@error and replaced error messages	87
General: Added tabmac code, and extended indexing	1
v0.2.1.	
\@lab: Removed page setting from \@lab	301
\edlabel: Tweaked \edlabel to get correct page numbers	298
\l@ddodoreintrafeet: Renamed \dodoreintrafeet to \l@ddodoreintrafeet	291
\l@ddoxtrafeet: Renamed \doxtrafeet to \l@ddoxtrafeet	289
\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	141
General: Added text about normal labeling	65
Bug fixes and match with mempatch v1.8	1
Major changes to insert code when memoir is loaded	292
v0.2.2.	
\footfudgefiddle: Added \footfudgefiddle	195
\next@line@list@stuff: Added initial write of page number in \line@list@stuff	131
\para@footsetup: Added \footfudgefiddle to \para@footsetup	196
\para@footsetupX: Added \footfudgefiddle to \para@footsetupX	241
General: Improved paragraph footnotes	1
New Dekker example	1
Used \providecommand for \@gobblethree to avoid clash with the amsfonts package	94
v0.3.0.	
\@lab: Replaced \the\line@num by \linenumr@p\line@num in \@lab, and similar for sub-lines	301
\@n@reg: Added a bunch of code to \@n@ for handling \setlinenum	118
\ledlinenum: Added \linenumr@p and \sublinenumr@p to \leftlinenum and \rightlinenum	109
\linenumberlist: Added \linenumberlist mechanism	95
\printendlines: Added \linenumr@p and \sublinenumr@p to \printendlines	260
\printlines: Added \linenumr@p and \sublinenumr@p to \printlines	223
\sublinenumr@p: Added \linenumberstyle and \sublinenumberstyle	108
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	324
\Xarrangement@normal: Added minpage footnote setup to \footnormal	190
\edtext: Added \showlemma to \edtext	142
\l@dfeetendmini: Added \l@dfeetbeginmini, \l@dfeetendmini and all their supporting code	322
\mpnormalfootgroup: Added \mpnormalfootgroup	194

\mpnormalvfootnote: Added \mpnormalvfootnote	192
\showlemma: Added \showlemma	95
General: Added final/draft options	83
Added ledgroup environment	325
Added ledgroupsize environment	326
Added minipage, etc., support	1
v0.4.1.	
\doXfeet: Changed \doXfeet code for easier extensions	290
\edindex: Let eledmac take advantage of memoir's indexing	331
\printXnotes: Added \opXfeet	290
General: Added code for changing \doclearpage	293
Not released. Minor editorial improvements and code tweaks	1
Only change \@footnotetext and \@footnotemark if memoir not used	226
v0.5.0.	
\@footnotetext: Enabled regular \footnote in numbered text	227
\@xympar: Eliminated \marginpar disturbance	314
General: Added left and right sidenotes	315
Added sidenotes, familiar footnotes in numbered text	1
v0.5.1.	
\affixline@num: Changed \affixline@num to cater for sidenotes	175
\l@getsidenote@margin: Added \sidenotemargin and \sidenote@margin	315
General: Added moveable sidenote	315
Fixed right line numbers killed in v0.5	1
Only change \hsize in ledgroupsize environment otherwise page number can be in wrong place	326
v0.6.0.	
\@lopR: Added \@pend, \@pendR, \@lopL and \@lopR in anticipation of parallel processing	121
\@nl@reg: Extended \@nl to include the page number	118
\get@thisfootnote: Changed \l@dbfnote and \v1@dbfnote as originals could give incorrect markers in the footnotes	227
\set@this@page: Extended \new@line to output page numbers	132
General: Fixed long paragraphs looping	1
Fixed minor typos	1
Prepared for eledpar package	1
v0.7.0.	
\@nl@reg: Added \@nl@reg	118
\@ref@reg: Added \@ref@reg	128
\affixline@num: Added skipnumering to \affixline@num	175
\do@actions@fixedcode: Added \do@actions@fixedcode	171
\do@actions@next: Added number skipping to \do@actions	169
\do@insidelinehook: Added \do@linehook for use in \do@line	166
\endnumbering: Changed \endnumbering for eledpar	99
\fx@l@cks: Added \ch@cksub@l@ck, \ch@ck@l@ck and \fx@l@cks	177
\footsplitskips: Added \footsplitskips for use in many footnote styles	188
\get@linelistfile: Added \get@linelistfile	116
\initnumbering@reg: Added \initnumbering@reg	98
\l@advance@parledgroup@beforenormalnotes: Added \l@dunboxmpfoot containing some common code	325
\l@dcsnotetext@r: Added \l@emptyd@ta	167

\l@dgetline@margin: Added \l@dgetline@margin	105
\l@dgetlock@disp: Added \l@dgetlock@disp	107
\l@dgetsidenote@margin: Added \l@dgetsidenote@margin	315
\l@dnumpstartsL: Added \l@dnumpstartsL, \ifl@dpairing and \ifpst@rted for/from eledpar	95
\l@drsn@te: Added \l@dlsn@te and \l@drsn@te for use in \do@line	167
\l@dzeropenalties: Added \l@dzeropenalties	161
\ledlinenum: Added \ledlinenum for use by \leftlinenum and \rightlinenum	109
\list@clearing@reg: Added \list@clearing@reg	116
\n@num: Added \n@num	126
\next@line@list@stuff: Deleted \page@start from \line@list@stuff	131
\normalbfnoteX: Removed extraneous space from \normalbfnoteX	233
\pausenumbering@page@num: Changed \resumenumbering for eledpar	101
\setprintendlines: Added \setprintendlines for use by \printendlines	257
\setprintlines: Added \setprintlines for use by \printlines	217
\skipnumbering: Added \skipnumbering and supports	138
\sublinenumincrement: Added \firstlinenum, \linenumincrement, \firstsublinenum and \linenumincrement	106
\sublinenumr@p: Using \linenumrep instead of \linenumr@p	108
Using \sublinenumrep instead of \sublinenumr@p	108
\vnumfootnoteX: Removed extraneous space from \vnumfootnoteX	235
General: eledmac having been available for 2 years, deleted the commented out original edmac texts	1
Maïeul Rouquette new maintainer	1
Made macros of all messages	87
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	95
General: For compatibility with eledpar, possibility to use \autopar on the right side.	1
Possibility to number \pstart.	20
Possibility to number the pstart with the commands \numberpstarttrue.	1
v0.12.1.	
General: Don't number \pstarts 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.	341
General: New <code>stanzaindentsrepetition</code> counter to repeat stanza indents every $n$ verses.	58
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 <code>memoir</code> 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.	298
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.	175
<code>\ifinserthangingsymbol</code> : New management of <code>hangingsymbol</code> insertion, preventing undesirable insertions.	340
<code>\printlines</code> : Line numbering can be reset at each <code>pstart</code> .	222
General: Line numbering can be reset at each <code>pstart</code> .	103
Possibility to print <code>\pstart</code> number at the sides of the text.	20
v0.17.0.	
<code>\ifinserthangingsymbol</code> : New new management of <code>hangingsymbol</code> insertion, preventing undesirable insertions.	340
v1.0.0.	
<code>\morenoexpands</code> : Change to be compatible with new features	141
General: <code>\lemma</code> can contain commands.	32
Debug in lineation command	22
New generic commands to customize footnote display.	43, 278
Options <code>nonum</code> and <code>nosep</code> in <code>\Xfootnote</code> .	30
Options of <code>\Xfootnotes</code> .	186
Possibility to have commands in sidenotes.	68
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.	44
v1.1.0.	
<code>\Xprenotes</code> : New skip <code>\Xprenotes@</code>	249
<code>\settoggle@series</code> : <code>\settoggle@series</code> switch the global value of the toggle, not only the local value.	279
General: Add <code>\labelpstarttrue</code> .	20
Add <code>\Xnumberonlyfirstintwolines</code>	44
Add <code>\Xpstart</code> and <code>\Xonlypstart</code>	47
New hook to add arbitrary code at the beginning of the notes	52
New options for block of notes.	54
New package option: <code>parapparatus</code> .	1
New tools to change order of series	277
Sectioning commands.	76
v1.2.0.	
<code>\Xprenotes</code> : Debug in familiar footnotes (bug introduced by v1.1).	249
<code>\endquote</code> : Compatibility of <code>\ledchapter</code> with the <i>memoir</i> class.	381

v1.3.0.		
\endquote:	Quotation and quote environment inside numbered sections. . . . .	381
v1.4.0.		
\edtext:	Compatibility of \edtext with the right-to-left direction (with Polyglossia). . . . .	142
\ledsetnormalparstuffX:	Direction of footnotes with polyglossia. . . . .	246
\newseries@:	Remembers the language of the lemma, in order to create a correct direction for the footnote separator. . . . .	267
\rbracket:	Switch the right bracket to a left bracket when the lemma is RTL (needs polyglossia or LuaTeX). . . . .	210
General:	Compatibility with LuaTeX of RTL notes. . . . .	1
v1.4.1.		
\affixside@note:	Remove spurious spaces. . . . .	321
\endquote:	New option <i>noquotation</i> . . . . .	381
\get@thisfootnote:	Compatibility of standard footnotes with eledmac when these footnotes contain any commands. . . . .	227
\labelrefsparsesubline:	Fix a bug with \edlabel. . . . .	299
v1.4.2.		
General:	Debug with some special classes. . . . .	1
v1.4.3.		
General:	Add \Xnonbreakableafternumber. . . . .	48
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. . . . .	381
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. . . . .	233
v1.5.0.		
\do@insidelinehook:	Added \do@insidelinehook for use in \do@line . . . . .	166
\edindex:	Compatibility with imakeidx package, and possibility to use multiple index with \edindex. . . . .	331
General:	Correct indexing when the call is made in critical notes. . . . .	327
v1.5.1.		
\managestanza@modulo:	Correct stanzaindentsrepetition counter . . . . .	341
\normalvfootnoteX:	Fix a bug with normal familiar footnotes when mixing RTL and LTR text. . . . .	230
v1.6.0.		
\newverse:	Add \falseverse macro. . . . .	345
v1.6.1.		
\AtStartEveryPstart:	Spurious space in \pstart. . . . .	156
\ifinserthangingsymbol:	Hang verse is now not automatically flush right. . . . .	340
\l@dunhbox@line:	Move the call to \inserthangingsymbol to allow use \hfill inside. . . . .	162

\pend: Spurious space in \pend. . . . .	158
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. . . . .	77
v1.8.0.	
\endquote: Correction of sectioning commands in parallel texts. . . . .	381
\get@index@command: Debug \get@index@command and compatibility with hyperref package. . . . .	330
\newhookcommand@series@reload: Debug \beforenotesX and \maxnotesX which did not work. . . . .	282
\prevpage@num: Correct \parafootsep when using with ledgroup. . . . .	201
General: Compatibility with parledgroup option of eledpar package. . . . .	1
If imakeidx and hyperref are loaded, adds hyperref in the index. . . . .	327
v1.8.1.	
General: Debug endnotes when more than one series is used (change the position where tools for endnotes are defined). . . . .	251
v1.8.2.	
General: Debug compatibility problem with hebrew option of babel package. . . . .	1
v1.8.3.	
General: Fixes spurious spaces added by v1.7.0. . . . .	1
v1.8.5.	
General: Debug indexing in right column, with eledpar. . . . .	327
v1.9.0.	
\l@ddoxtrafeet: Add \fnpos to choice the order of footnotes. . . . .	289
\l@dfeetendmini: Add \mpfnpos to choice the order of footnotes in minipage / ledgroup. . . . .	322
v1.10.0.	
\endquote: Correction of sectioning commands in parallel texts. . . . .	381
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. . . . .	51
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 \@1. Consequently, \@1@reg becomes \@nl@reg. . . . .	118

\AtStartEveryPstart: New optional argument for \pstart, to execute code before it.	156
\edindex: Use correctly default index when imakeidx is loaded.	331
\endquote: \ledxxx sectioning commands are deprecated and replaced by \eledxxx commands.	381
\initnumbering@reg: \beginnumbering is defined only on eledmac, not on eledpar.	98
\l@dgetsidenote@margin: \sidenotemargin is now directly defined in eledmac to be able to manage eledpar.	315
\l@disnote: \l@dlsnote, \l@drsnote and \l@dcnote defined only one time, in eledmac, including needs for eledpar case.	316
\l@dnumbstartsL: Add \ifledRcol@ for eledpar	95
\l@dunhbox@line: \do@line is split in more little commands.	163
\newhookcommand@series@reload: Debug \beforenotesX and \maxhnotesX which did not work when called after \footparagraphX.	282
Debug \Xbeforenotes and \Xmaxhnotes which did not work when called after \footparagraph.	282
\pend: New optional argument for \pend, to execute code after it.	158
\stanza: &can have an optional argument: content to be printed after.	345
\Stanza can have an optional argument: content to be printed before.	345
Add \newverse macro, \falseverse deprecated.	345
General: Add \ledinnernote and \ledouternote commands.	68
Add \Xendparagraph and related settings.	56
Add hyperlink to crossref (needs hyperref package).	63
Compatibility with musixtex.	1
Debug eledmac sectioning command after using \resumenumbering.	1
Ensure that imakeidx is loaded <i>before</i> eledmac	327
New hooks: \Xafterrule and \afterruleX	55
New options for ragged-paragraph notes	54
New sectioning commands.	76
Optional arguments for \pstart and \pend.	19
v1.12.1.	
\wrap@edcrossref: Fix spurious spaces.	303
v1.12.2.	
\l@dunhbox@line: Fix a bug with critical notes at the tops of pages (added by v12.0.0)	162
v1.12.3.	
\flag@end: \flag@start and \flag@end are now defined only one time for eledmac and eledpar	133
\flag@start send a error message when a \edtext is done without insert (note)	133
\reledmac@error: Replaced error messages	87
General: Add macros for new messages since v0.7	87
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.	68
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	333
\@ssect: Debug \eledchapter in parallel with memoir	384
\doinsidelinehook: Added \dolinehook and \doinsidelinehook	166

\endnumbering: Allow to mix parallel columns and normal text when using	
\pausenumbering	99
\l@dgobbleoptarg: \l@dgobblearg becomes \l@dgobbeloptyarg	364
\l@drestoreforedtext: Debug optional arguments of \Xfootnote in tabular context	365
\pausenumbering@page@num: Debug \resumenumbering	101
v1.12.7.	
\wrap@edcrossref: \wrap@edcrossref is now robust	303
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	133
v1.13.0.	
\newhooktoggle@series@reload: Add \newhookcommand@toggle@reload	282
\para@footsetupX: In \para@footsetupX, use \columnwidth instead of \hsize	241
\settoggle@series: \settoggle@series can take an optional arguments to reload series setup.	279
General: Add \Xnoteswidthliketwocolumns and \noteswidthliketwocolumnsX	56
Added widthliketwocolumns option	83
v1.13.1.	
\ifat@every@pstart@star@: Add \l@dzeropenalties in \pstart	156
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@dpaging for eledpar	95
General: Fix a 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 a bug with index when memoir class is used without hyperref	1
v1.14.0.	
\edindex: Let eledmac take advantage of imakeidx even when memoir class is used	331
General: Debug spurious characters before endnotes.	251
Delete previous override of \l@d@@wrindexhyp at the beginning of a document when hyperref is not loaded.	335
Move gobbling command	94
Provide \@gobblefour	94
v1.14.1.	
\@ssect: Debug sectioning commands when using both handout and hyperref package.	387
v1.14.2.	
\@ssect: Debug \edtext after starred sectioning commands when using memoir class.	384
v1.15.0.	
\@edtext@level: New boolean \if@edtext@.	141
\arrangementX@paragraph: Correct bug with paragraphed familiar footnotes setting.	240
\endsub: Restore subline feature (disabled by mistake in v1.8.0).	136
\if@lemmacommand@: New boolean \iflemmacommand@.	147
General: Fix a bug with footnotes layout when using some options of the geometry package (bug add by v1.13.0).	1
New commands \AtEveryPstart and \AtEveryPend.	20
New tools to prevent ambiguous references in lemma	33



v1.15.1.	
\next@line@list@stuff:	Revert modification of 1.5.2 which makes bug with numbering. Leave vertical mode to solve spurious space before minipage. . . . .
	131
v1.16.0.	
General:	\edtext is now defined only in eledmac, not in eledpar. Debug wrong numbering when using \sameword + eledpar + \tag command. . . . .
	142
Compatibility of standard footnotes with some biblatex styles.	. . . . .
	1
New \stanzaindent command.	. . . . .
	1
v1.16.1.	
\xlineref:	\lineref is not defined if defined by some other package, like lineno. Eledmac provides \edlineref instead. . . . .
	303
v1.17.0.	
\edtext:	Error message when calling \edtext outside of a numbered paragraph. . . .
	142
v1.18.0.	
\@edindex@hyperref:	Fix spurious space with \edindex when using imakeidx/indextools + hyperref. . . . .
	333
\edlabel:	\edlabel is now defined only one time for both eledmac and eledpar . . . .
	298
\l@d@section:	Option parapparatus works for endnotes. . . . .
	252
\l@dnumpstartsl:	Add \ifl@dprintingpages and \@dprintingcolumns for eledpar . . . . .
	95
\print@line:	Compatibility with Lua $\TeX$ RTL languages. . . . .
	163
\printlinefootnote:	Code refactoring in \printlinefootnote: the printing of the numbers are factorized in \printlinefootnotearea . . . . .
	212
\printpstart:	Debug \Xpstart with parallel pages and columns (eledpar) . . . . .
	211
General:	Add \Xpstarteverytime . . . . .
	47
Compatibility with Lua $\TeX$ RTL languages.	. . . . .
	1
Debug \Xonlypstart when using \Xnumberonlyfirstinline and the current line number differs from the previous.	. . . . .
	47
v1.19.0.	
\footplitskips:	\footplitskips doesn't set \floatingpenalty to \@MM when processing parallel pages. . . . .
	188
\xxref:	\xxref works also with right side numbers, when \@Rlineflag is not empty. . . .
	306
General:	\Xmaxhnotes and \maxhnotesX work now for both two-columns and three-columns setting. . . . .
	1
Compatibility with eledpar v1.13.0.	. . . . .
	1
v1.19.1.	
General:	Call \correct@footinsX@box and \correct@Xfootins@box directly in \print@notesX@forpages and \print@Xnotes@forpages, that is in eledpar. . . .
	1
v1.20.0.	
\printlines:	Added \ifl@dXmorethantwolines and \ifl@dXmorethantwolines to \printlines . . . . .
	223
\stanza:	& and \& can be preceded by spaces. . . . .
	345
\xxref:	Debug \xxref when not loading eledpar (Fix a bug added in 1.19.0). . . . .
	306
General:	Add \Xendboxlinenum . . . . .
	49
Add \Xtwolines and \Xmorethantwolines hooks	. . . . .
	46
Add series option.	. . . . .
	1
Correct \Xinplaceofnumber hook.	. . . . .
	1
Explicit error message when calling \Xfootnote outside of \edtext.	. . . . .
	1
Fix a bug with line number typesetting direction when using \eledsection and similar commands for RTL texts with Lua $\TeX$ .	. . . . .
	1

Fix issues with RTL text in notes when using Lua $\TeX$ .	1
Options fulllines in $\backslash$ Xfootnote.	30
The $\backslash$ newifs are not followed by boolean values set to false, because it is the $\TeX$ default setting.	1
v1.21.0.	
$\backslash$ @edindex@hyperref: Look at the hyperindex option of hyperref before inserting hyperref	333
$\backslash$ l@d@section: $\backslash$ endnotes take five arguments.	252
$\backslash$ ledinnotemark: Add $\backslash$ ledinnotemark.	330
$\backslash$ ledsetnormalparstuffX: $\backslash$ ledsetnormalparstuff is deprecated and becomes $\backslash$ ledsetnormalparstuffX and $\backslash$ Xledsetnormalparstuff.	246
$\backslash$ n@num: $\backslash$ n@num@ref deleted	126
$\backslash$ n@num defined only one time for both Eledmac and Eledpar	126
$\backslash$ newhookcommand@series: $\backslash$ newhookcommand@series can take an optional argument.	281
$\backslash$ newhooktoggle@series: $\backslash$ newhooktoggle@series can take an optional argument.	281
$\backslash$ print@footnoterule: Code refactoring: the spaces after the footnote rules are directly managed in $\backslash$ print@Xfootnoterule and $\backslash$ print@footnoterule	248
$\backslash$ seriesatend: Fix spurious space in $\backslash$ seriesatend	278
$\backslash$ skipnumbering: $\backslash$ skipnumbering defined only one time for both Eledmac and Eledpar.	138
Correct $\backslash$ skipnumbering for stanza.	138
Delete $\backslash$ skipnumbering@reg.	138
General: $\backslash$ AtEveryPstart and $\backslash$ AtEveryPend are now compatible with $\backslash$ autopar	1
$\backslash$ Xafterrule and $\backslash$ afterruleX features no longer create problems of overflowing at the bottom of the page.	1
$\backslash$ chapter inside optional argument of $\backslash$ pstart works when typesetting parallel pages	1
$\backslash$ preXnotes and $\backslash$ prenotesX features no longer create problems of overflowing at the bottom of the page.	1
$\backslash$ seriesatbegin and $\backslash$ seriesatbegin more efficient	277
Add $\backslash$ applabel and related	66
Add $\backslash$ beforenotesX and $\backslash$ Xbeforenotes features for notes set in two and three column.	1
Add $\backslash$ hidenummering	25
Add $\backslash$ Xcolalign and $\backslash$ colalignX hooks	53
Add $\backslash$ Xendtwolines, $\backslash$ Xendmoreethantwolines, $\backslash$ Xendtwolinesbutnotmore and $\backslash$ Xendtwolinesonlyinsamepage.	47
Add $\backslash$ Xparindent and $\backslash$ hangindentX	52
Add $\backslash$ Xtwolinesbutnotmore and $\backslash$ Xtwolinesonlyinsamepage.	1
Add nocritical, noend, nofamiliar and noledgroup options.	1
Add noeledsec package option	1
Debug $\backslash$ beforenotesX $\backslash$ maxhnotesX $\backslash$ noteswidthliketwocolumnsX and $\backslash$ afterruleX with footnotes set in two and three columns.	1
Fix a bug when a $\backslash$ Xfootnote follows a $\backslash$ Xendnote in the second argument of $\backslash$ edtext (bug added in eledmac 1.0.0).	1
Fix a bug with $\backslash$ maxhnotesX when using $\backslash$ foottwocolX or $\backslash$ foottwocolX.	1
Fix a 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> . . . . .	94
v1.22.0.	
<code>\ledinnote</code> : <code>\ledinnote</code> takes a first optional argument, which is the label for hyperlinks. . . . .	330
General: Add <code>\doendnotesbysection</code> command. . . . .	31
Add option for lemma separator inside endnotes . . . . .	50
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. . . . .	31
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> . . . . .	201
General: Fix a 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> . . . . .	141
<code>\Serefwithpage</code> : Debug <code>\Xendtwolines</code> , <code>\Xendmoreethantwolines</code> , <code>\Xendtwolinesbutnotmore</code> and <code>\Xendtwolinesonlyinsamepage</code> when using <code>\apprefwithpage</code> . . . . .	310
<code>\lemma</code> : Fix spurious space after <code>\lemma</code> command . . . . .	147
<code>\newsseries@</code> : Prevent spurious spaces when <code>\Afootnote</code> and similar commands are followed by spaces (bug added on 1.0.0). . . . .	267
<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. . . . .	152
General: Add <code>\Xboxlinenumalign</code> and <code>\Xendboxlinenumalign</code> . . . . .	49
Add <code>\Xboxstartlinenum</code> , <code>\Xendboxstartlinenum</code> , <code>\Xboxendlinenum</code> , <code>\Xendboxendlinenum</code> . . . . .	49
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</code> ... <code>\endnumbering</code> without <code>\pstart</code> . . . . .	1
Fix a bug with <code>\sameword</code> when the lemma overlaps multiple line. . . . .	33
Fix a bug with <code>\sameword</code> when the same lemma is used for multiple notes or for nested <code>\edtexts</code> . . . . .	33
Fix a 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. . . . .	33
v1.23.1.	
General: Fix a bug with <code>\lemma</code> command in the right side. . . . .	1
v1.23.2.	
General: Compatibility with $\text{\LaTeX}$ '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 a 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. . . . .	324
<code>\@xympar</code> : Patching <code>\@xympar</code> instead of redefining it . . . . .	314
<code>\endminipage</code> : Patch <code>\endminipage</code> instead of redefining it. . . . .	324
<code>\initnumbering@quote</code> : <code>\initnumbering@sectcmd</code> becomes <code>\initnumbering@quote</code> . . . . .	381
<code>\l@advance@parledgroup@beforenormalnotes</code> : Some code of <code>\l@dumboxmpfoot</code> moved to <code>\l@advance@parledegroupp@beforenormalnotes</code> . . . . .	325
<code>\newseries@</code> : One endnotes file by series. . . . .	274
General: <code>\@makecol</code> , <code>\@reinserts</code> and <code>\@doclearpage</code> are patched instead of begin redefined . . . . .	292
<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> . . . . .	292
Add <code>\Xendinplaceofnumber</code> hook. . . . .	1
Add <code>\Xendnonumber</code> hook. . . . .	1
Add <code>nonum</code> option for endnotes. . . . .	1
Fix a 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 $\text{\LaTeX}$ ’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 . . . . .	226
Suppress indexing command specific to <code>memoir</code> . . . . .	331
v2.0.1.	
General: Fix a 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 a bug with <code>\advanceline</code> at the beginning of a <code>\pstart</code> . . . . .	1
Fix a bug with <code>\chapter</code> in optional argument of <code>\pstart</code> in parallel typesetting with <code>scrbook</code> . . . . .	1
Fix a bug with <code>\eledchapter</code> in parallel typesetting with <code>scrbook</code> . . . . .	1
Fix a 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 a bug with <code>\ledpbsetting{before}</code> . . . . .	1
v2.1.2.	
General: Fix a 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. . . . .	246
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 a bug which inserted double space before paragraphed familiar notes. . . . .	1
Fix a bug with <code>\edindex</code> when using not-Latin characters without UTF-8 engines . . . . .	1
v2.2.0.	
General: Fix a bug with combination of <code>\onehalfspacing</code> and two columns and three	
columns notes typeset. . . . .	1
Fix a 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 $\text{TeX}$ format 2015/10/01. . . . .	1
v2.2.2.	
General: Fix a bug in <code>\sethangingsymbol</code> . . . . .	1
Fix a bug with old version of <code>etex</code> . . . . .	1
v2.3.0.	
General: Disable empty lines as paragraph in stanza. . . . .	1
Fix incompatibility 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. . . . .	186
General: <code>\Xbhooknote</code> and <code>\bhooknoteX</code> work with notes in columns. . . . .	1
Fix a bug of <code>\parindentX</code> and <code>\Xparindent</code> with two columns and three columns	
notes. . . . .	1
Fix a 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>\Xlinerrangeseparator</code> and <code>\Xendlinerrangeseparator</code> . . . . .	46

Option linerangesep for critical footnotes and endnotes. . . . .	46
v2.4.1.	
General: Fix a bug with \appref and \apprefwithpage (introduced in v2.4.0). . . . .	1
Fix a bug with tabular environments when using babel or polyglossia languages that override $\LaTeX$ \roman command, like Greek language. . . . .	1
Fix a bug with tabular environments when using babel or polyglossia languages that override $\LaTeX$ \roman command, like Greek. . . . .	1
v2.5.0.	
\Serefwithpage: Debug \setapprefprefixsingle . . . . .	310
\edlabel: Fix a bug when calling \edlabel in a footnotes of the right side . . . . .	298
\l@d@section: \endnotes take six arguments. . . . .	252
\printlines: \printlines takes an eighth argument: the line flag . . . . .	222
\xlineref: \xlineref does not include anymore the side flag. Use \xflagref to get it. Not that \edlineref still contains the flag. . . . .	303
General: \apprefwithpage and \appref print double quotation mark when the label was not defined. . . . .	1
\apprefwithpage and \appref work with right side crossref. . . . .	1
\apprefwithpage works also when noend option is enabled. . . . .	1
\appref and \apprefwithpage can take linerangesep optional argument. . . . .	1
\edlabel works now in \Xfootnote. . . . .	1
\lemma can be used even when the nocritical is enabled. . . . .	1
Compatibility with new hook and tools of reledpar 2.6.0. . . . .	1
Fix spurious vertical space in astanza environment (reledpar) . . . . .	1
Log now states ‘There were undefined references’ when using wrong references in \edlineref or edpageref. . . . .	1
New hooks to customize page and line number appearance in endnotes. . . . .	1
New hooks: \Xhookgroup and \hookgroupX. . . . .	1
New tools to easily make cross-reference to a passage defined by a start and an end line	65
v2.6.0.	
General: Adds compatibility with innnote and notenumber options of indextools package. . . . .	1
Fix a bug with footnote counter in ledgroup (added in v2.5.0). . . . .	1
Fix bug, introduced in v2.5.0, with footnote numbering in parallel typesetting when using perpage package. . . . .	1
v2.7.0.	
\@k: \rigidbalance is split in \Xrigidbalance and \rigidbalanceX. . . . .	202
\l@d@section: \endnotes take seven arguments. . . . .	252
General: Add dash as default page range separator for \SEonlypage . . . . .	1
Debug \SErefonlypage when referring to only one page. . . . .	1
Delete parenthesis after \SErefonlypage. . . . .	1
Fix (again) bugs with footnote numbering in parallel typesetting while using ledgroup environments (bug added in v2.5.0). . . . .	1
Fix a bug with \SErefwithpage. . . . .	1
Fix bugs in compatibility with innote and notenumber options of indextools package, when indexing outside of a ledgroup. . . . .	1
New commands to make glossaries connected to page and linenumber with the glossaries package . . . . .	1
New hooks: \Xhsize and \hsizeX . . . . .	56
New hooks: \Xlemmfont and \xendlemmfont . . . . .	51

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 related, <code>\appref</code> and related. . . . .	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> . . . . .	56
Fix problem of hyphenation when using <code>hyperref</code> package (added in v2.7.0). . . . .	1
v2.8.0.	
<code>\l@d@section: \Xendhangindent</code> and <code>\Xendafternote</code> can take values which are relative to the font size of the endnote. . . . .	252
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 a bug concerning indent in a paragraph immediately following a sectioning command (bug NOT fixed on <code>reledpar</code> ) . . . . .	1
Fix a bug with <code>\AtEveryPstart</code> added in version 2.0.0. . . . .	1
Fix a 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 a bug when using <code>\lineneation{page}</code> and <code>\pausenumbering... \resumenumbering</code> . . . . .	1
Fix a 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 a bug when notes start with “plus” or “minus”. . . . .	1
v2.9.2.	
General: Fix a 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 a bug with line number position and reset added by v. 2.9.0 . . . . .	1

v2.10.0.	
\print@lemma: Code refactoring between \parafootfmt, \twocolfootfmt, \threecolfootfmt and \normalfootfmt. . . . .	211
General: Add \AtEveryStanza and \AtEveryStopStanza. . . . .	1
Fix a bug in \ledlsnotefontsetup and \ledrsnotefontsetup which could not handle \color command properly. . . . .	1
More specific error messages. . . . .	1
New hooks: \Xwrapcontent, \Xendwrapcontent and \wrapcontentX. . . . .	52
New hooks: \Xwraplemma and \Xendwraplemma . . . . .	51
v2.10.1.	
General: Add ‘nopenalties’ option. . . . .	1
Fix a bug introduced in v. 1.4: not paragraphed critical footnotes could prevent marginal line number from being displayed . . . . .	1
v2.11.0.	
\do@actions@fixedcode: Add action 1010 . . . . .	171
General: Add new tools to produce an apparatus of manuscripts . . . . .	1
Fix a bug in \Xparafootsep in parallel typesetting . . . . .	1
Make \parafootsepX work . . . . .	1
Prevent \Xtxtbeforenotes hook from causing notes to go beyond the bottom margin . . . . .	1
v2.12.0.	
General: \preXnotes becomes \Xprenotes (naming convention) . . . . .	1
Add \hidenumberingonleftpage and \hidenumberingonrightpage . . . . .	1
Add \toendnotes and related. . . . .	1
Add auxdir option. . . . .	1
Fix a bug in critical and familiar footnotes when using uppercase letters with accent mark . . . . .	1
Fix a bug when using \chapter in optional argument of \pstart in parallel typesetting in combination with the noeledsec option. . . . .	1
Fix a bug with \ledinnernote and \ledouternote in parallel typesetting . . . . .	1
Fix a bug with familiar footnote number in optional argument of \pstart or \pend in parallel typesetting . . . . .	1
Fix spurious vertical space in \chapter when used as optional argument of \pstart in parallel typesetting. . . . .	1
Make endnote compatible with \sameword mechanism . . . . .	1
More accurate message to control the position of \Xfootnote and \applabel in the $\LaTeX$ 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 \Xlineflag (bug added in v. 2.5.0). . . . .	1
v2.13.2.	
General: Fix a bug added in v. 11.2 which could make parallel typesetting not work. . . . .	1
v2.13.3.	
General: Makes \Xendafterpagenumbe affecting \Serefwithpage . . . . .	1
v2.14.0.	
General: Hyperref with the line number inside critical footnotes is correct when using \xxref . . . . .	1
Some internal changes for new features of reledpar. . . . .	1



v2.14.1.	
General: Fix a 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 a bug with <code>fancyhdr</code> package 3.8 and later.	1
v2.15.3.	
General: Fix a 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>	292
Add <code>\Xendpagenumberonlyfirst</code> , <code>\Xendpagenumberonlyfirstifsingle</code> , <code>\Xendpagenumberonlyfirstintwo</code> , <code>\Xendinplaceofpagenumber</code> and <code>\Xendsympagenum</code> hooks.	1
Add <code>\Xpagelinesep</code> hook.	48
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 a bug with <code>\Xgroupbylines</code> for notes in two columns	1
Fix a bug with <code>\Xtxtbeforenotes</code> for notes in three or two columns	1
Fix a 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.	292
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 a 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 a 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 a bug with <code>\Seref</code> (bug added in v2.7.0).	1
v2.16.4.	
General: Fix a 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 a bug with the line number style in <code>\doenotes</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 a bug with <code>\msdata</code> when using multiple <code>\beginnumbering... \endnumbering</code> . . . . .	1
Fix a bug with <code>\numberpstarttrue</code> when using multiple <code>\beginnumbering... \endnumbering</code> . . . . .	1
v2.16.8.	
General: Fix a 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 a bug with <code>\doendnotesbysection</code> and <code>\doendnotes</code> . . . . .	1
v2.17.4.	
General: Fix a bug with <code>\setSErefonlypageprefixsingle</code> and <code>\setSErefonlypageprefixmore</code> . . . . .	1
v2.17.5.	
General: Fix a bug with <code>\pstartref</code> when referring to the left side in parallel typesetting. . . . .	1
v2.18.0.	
General: Fix a 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 a 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 a 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 a 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 to <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 to <code>\newverse</code> . . . . .	1
Fix a 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> commands.	1
Fix a bug when using formatting command in the argument of <code>\edindex</code> inside <code>\edtext</code> .	1
Now, as explained in the handbook, a <code>\edindex</code> inside <code>\edtext</code> only creates an index reference to the main text, and not to the critical footnote.	1
v2.22.0.	
General: Add <code>\txtbeforenotesonlyonceX</code> and <code>\Xtxtbeforenotesonlyonce</code> hooks.	1
Add <code>\txtbeforenotesX</code> hook.	1
Fix a bug added in v2.16.0 when using <code>\Xtxtbeforenotes</code> with paragraphed or normal footnotes.	1
Fix a bug with three and two columns critical footnotes, broken in v. 2.17.6.	1
v2.22.1.	
General: Compatibility with new version of <code>reledpar</code> .	1
Fix a bug with some commands inside <code>\sameword</code> .	1
v2.22.2.	
General: Do not print footnotes at the first run.	1
v2.23.0.	
General: Add <code>swcaseinsensitive</code> option	1
Optimisation of the code added on v2.22.2 to not print footnotes at the first run.	1
v2.24.0.	
<code>\next@line@list@stuff</code> : Add <code>\next@line@list@stuff</code> internal hook.	131
General: Add <code>\swnoexpands</code> macro to avoid problems with not fully expandable macro inside <code>\sameword</code> .	1
Fix a bug with <code>\edlabel</code> at the beginning of a <code>\pstart</code> .	1
Fix spurious space with <code>\labelpstarttrue</code>	1
When a <code>\setlinenum</code> is used, it is stronger than the <code>\lineation{page}</code> setting.	1
v2.24.1.	
General: Compatibility with <code>reledpar 2.20.2</code>	1
v2.24.2.	
General: Fix bug between tabular environments and endnotes	1
v2.25.0.	
General: Add <code>\Xendtxtbeforenotes</code> hook	1
v2.26.0.	
General: Add a mechanism for annotation of stanza line numbering	1
Fix bug in <code>\xpstartref</code>	1
v2.26.1.	
General: Fix bug with <code>\linenumannotation</code>	1
v2.26.10.	
General: Add error message when insertion a <code>\pstart</code> or <code>\pend</code> inside <code>\edtext</code> arguments	1
Really fix bug with page number in footnote after <code>\mainmatter</code> (bug introduced in v2.26.7)	1
v2.26.11.	
General: Allow to use <code>\edtext</code> inside <code>\sameword</code>	1
v2.26.12.	
General: Precise minimal version of <code>xparse</code> package required.	1
v2.26.2.	
General: Fix bug with <code>\linenumannotation</code> in parallel typesetting	1
Fix bug with <code>\Xgroupbyline</code> and <code>\Xparafootsep</code> .	1

Fix bug with <code>\Xgroupbyline</code> and optional argument of critical footnotes. . . . .	1
Fix bug with <code>\Xgroupbyline</code> for the series after the first one. . . . .	1
v2.26.3.	
General: Fix bug added by v2.26.0 with notes in parallel typesetting . . . . .	1
v2.26.4.	
General: Fix bug when using <code>\numberpstarttrue</code> and alternating between manual <code>\pstart</code> and automatic <code>\pstart</code> produced by <code>\autopar</code> . . . . .	1
v2.26.5.	
General: Check that users have a enough recent version of <code>xparse</code> . . . . .	1
Fix bug when using <code>\eledsection</code> with the <code>continuousnumberingwithcolumns</code> of <code>reledpar</code> . . . . .	1
Fix bug with <code>continuousnumberingwithcolumns</code> . . . . .	1
Fix bug with using <code>\eledsection</code> and related when alternate normal typesetting and parallel typesetting . . . . .	1
v2.26.6.	
General: Fix bug with <code>\lineation</code> by page and nexted <code>\edtext</code> between two pages (added in v. 2.24.0) . . . . .	1
v2.26.7.	
General: Fix page number in footnote and endnotes when using <code>sameparallelpagenumber</code> option of <code>reledpar</code> and typesetting critical text in single mode after critical text in double page mode . . . . .	1
v2.26.8.	
General: Fix bug in some specific cases, introduced by v.2.27.7 . . . . .	1
v2.26.9.	
General: Fix bug with critical footnotes line number when alternating between <code>\numberlinefalse</code> and <code>\numberlinetrue</code> . . . . .	1
Fix bug with page number in footnote after <code>\mainmatter</code> (bug introduced in v2.26.7) . . . . .	1
Workaround with a bug of <code>TeX2018/04/01</code> patch level 3 to generate the .sty file from a .ins file . . . . .	1
v2.27.0.	
General: Add <code>\doinsidethislinehook</code> . . . . .	1
v2.27.1.	
General: Fix spurious space in <code>\hidenumbering</code> . . . . .	1
v2.28.0.	
General: Familiar footnotes can take an optional argument to manually define the footnote mark. . . . .	1
v2.28.1.	
General: Fix bug with the alignment of the columns of two and three columns critical footnotes. . . . .	1
v2.29.0.	
General: Add <code>noresetlinenumannotation</code> . . . . .	1
Fix bug with <code>\linenumannotation</code> and ‘ <code>nofamiliar</code> ’ option. . . . .	1
Fix little bug with <code>\msdata</code> . . . . .	1
Really reset line number annotation at each line . . . . .	1
v2.30.0.	
General: Add <code>\Xnoidenticallinenumannotation</code> and <code>\Xendnoidenticallinenumannotation</code> hooks . . . . .	1
Add separator between line number annotations in margin. Add <code>\setlinenumannotationsep</code> . . . . .	1

v2.31.0.	
General: Add <code>noprefix</code> and <code>prefixmore</code> to <code>\appref</code> and <code>\Seref</code> .	1
v2.31.1.	
General: Fix bug in vertical spacing with <code>\Xgroupbyline</code> .	1
Fix bug with <code>\Xgroupbyline</code> , which could produce spurious blank lines.	1
Fix bug with <code>\Xhangindent</code> and <code>\hangindentX</code> for right-to-left texts.	1
Fix bug with indentation with <code>\Xgroupbyline</code> .	1
Fix bug with multiple footnote layer and <code>\Xgroupbyline</code> .	1
Fix incompatibility between <code>\Xgroupbyline</code> and <code>hyperref</code> package.	1
Fix incompatibility between <code>\Xgroupbyline</code> and optional argument of critical footnotes.	1
v2.31.2.	
General: Fix bug with familiar footnote containing non-default font (introduced with v. 2.20.0).	1
v2.31.3.	
General: Fix bug with paragraphed footnote in RTL context.	1
v2.32.0.	
General: Add <code>\Xlinenumannotationonlyfirst</code> , <code>\Xlinenumannotationonlyfirstintwo</code> , <code>\Xsymlinenumannotation</code> , <code>\Xendlinenumannotationonlyfirst</code> , <code>\Xendlinenumannotationonlyfirstintwo</code> and <code>\Xendsymlinenumannotation</code> .	1
v2.32.1.	
General: Do not print <code>\Xendsep</code> if we print <code>\Xendsymlinenum</code> .	1
When using <code>\parindentX</code> or <code>\Xparindent</code> and stanza, use the normal indentation in the footnote, not the verse indentation.	1
v2.32.2.	
General: Fix spurious vertical space in familiar footnotes with Lua $\TeX$ .	1
v2.32.3.	
General: Fix bug with critical footnotes when typesetting Arabic text with <code>polyglossia</code> .	1
v2.32.4.	
General: <code>\footnoteXmark</code> and <code>\footnoteXtext</code> can now have an optional argument, setting the footnote number.	1
Fix bug <code>\footnoteXmark</code> and <code>\footnoteXtext</code> .	1
Fix bug when using <code>\Xgroupbyline</code> and twocol arrangement.	1
Fix bug with three and two columns footnotes when using <code>bid</code> .	1
v2.32.5.	
General: Fix bug with <code>\edindex</code> inside <code>\eledsection</code> and related.	1
Fix bug with macro inside <code>\sameword</code> with no- $\mathcal{X}\TeX$ engines.	1
Fix bug with page number in index with <code>\edindex</code> when indexing a critical notes.	1
v2.32.6.	
General: Add <code>hyperref</code> link when inserting glossary entries.	1
Fix bug with glossary entries inside the first argument of <code>\edtext</code> .	1
v2.32.7.	
General: Fix bug with <code>\lineation{pstart}</code> and <code>\edtext</code> in the very first line of a <code>\pstart</code> .	1
Fix bug with <code>\parafootsepX</code> .	1
Fix spurious spaces at the very first line of <code>edtabular</code> environments.	1
v2.32.8.	
General: Add subline number in manuscript apparatus.	1

v2.33.0.	
\ledinnotemark: \ledinnotemark redefinition is deprecated, use \Xinnotemark and \innotemarkX instead. . . . .	330
\newhookcommand@series: \newhookcommand@series can take an second optional argument (third argument). . . . .	281
General: Add \Xinnotemark and \innotemarkX hooks. . . . .	1
Add antilabe feature. . . . .	1
Add startstanzaindentsrepetition feature. . . . .	1
v2.33.1.	
General: Fix a spacing bug when the argument of \antilabe contains more than one word. . . . .	1
v2.34.0.	
General: Add \Xpstartonlyfirst and \Xstanzaonlyfirst. . . . .	1
Add \Xpstartseparator. . . . .	1
v2.34.1.	
General: Fix bug with page number when using stanazindentsrepetition, bug introduced in v2.33.0 . . . . .	1
v2.34.2.	
General: Fix bug with tabular environment, in some specific use case, added in v2.32.7 . . .	1
v2.34.3.	
General: Fix bug with \parafootsepX when the footnotes are called outside of a numbered section. . . . .	1
Fixes incompatibility between Fixes incompatibility between lineation by \pstart and (a)stanza. . . . .	1
Makes command like \hidenumbering working at the very beginning of a verse when the first argument of \setstanzaindents is equal to 0. . . . .	1
v2.34.4.	
General: Fix bugs with \parafootsepX outside of numbered section. . . . .	1
Fix some bugs with lineation by page, when a numbered section starts at the very beginning of page. . . . .	1
v2.34.5.	
General: Fix incompatibility between \apprefwithpage / \Serefwithpage and line number annotation mechanism. . . . .	1
v2.34.6.	
General: Fix incompatibility with calc package (add with v.2.34.4 . . . . .	1