

# 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 v2.32.0, last revised 2019/07/15.

<sup>†</sup>maieul at maieul dot net

1.2.4 <code>reledmac</code> . . . . .	15
1.3 Bibliography . . . . .	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>15</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>17</b>
5.1 Text lines numbering . . . . .	17
5.2 Paragraphs . . . . .	18
5.2.1 Basics . . . . .	18
5.2.2 Automatically producing <code>\pstart</code> ... <code>\pend</code> . . . . .	18
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> . . . . .	19
5.2.5 Numbering paragraphs ( <code>\pstart</code> ) . . . . .	20
5.2.6 Languages written in Right to Left . . . . .	20
5.2.7 Memory limits . . . . .	20
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 . . . . .	22
5.3.5 Other settings . . . . .	23
5.4 Changing the line numbers . . . . .	23
5.4.1 Sublineation . . . . .	23
5.4.2 Locking lineation . . . . .	23
5.4.3 Setting and changing line number . . . . .	23
5.4.4 Line number style . . . . .	24
5.4.5 Skipping and hiding number . . . . .	24
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 . . . . .	27
5.5.5 Separator between annotations . . . . .	27
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>28</b>
6.1 Terminology	28
6.2 Critical notes	29
6.2.1 The lemma	29
6.2.2 Footnotes	29
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 Use with <code>\lemma</code> command	34
6.3.5 Sameword for a group of words	36
6.3.6 Customizing	37
6.3.7 Problems with some macros	37
6.3.8 Automatic sameword annotation	38
6.4 Apparatus of manuscripts	38
6.4.1 Marking sections of text	38
6.4.2 Layout of the apparatus of manuscripts	39
6.4.3 Settings	39
6.5 Familiar notes	40
6.5.1 Basic use	40
6.5.2 Customizing mark	40
6.5.3 Separator for multiple footnotes	40
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	41
<b>7 Apparatus customization</b>	<b>42</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 at first time	44
7.2.3 Print page number only at first time	45
7.2.4 Arbitrary text before line number	45
7.2.5 Separator for line range	46
7.2.6 Abbreviate line range	46
7.2.7 Disable line number	46
7.2.8 Printing pstart number	46
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 . . . . .	47
7.2.12 Space around number . . . . .	47
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 . . . . .	49
7.4.1 For footnotes . . . . .	49
7.4.2 For endnotes . . . . .	50
7.5 Font style . . . . .	50
7.5.1 For line number . . . . .	50
7.5.2 For the lemma . . . . .	50
7.5.3 For all notes . . . . .	51
7.6 Wrapping notes . . . . .	51
7.6.1 Wrapping lemmas . . . . .	51
7.6.2 Wrapping contents . . . . .	51
7.7 Indent of notes content . . . . .	51
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 . . . . .	52
7.10.1 Alignment . . . . .	52
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 . . . . .	53
7.12 Options for block of notes . . . . .	53
7.12.1 Grouping notes by line . . . . .	53
7.12.2 Text before notes . . . . .	54
7.12.3 Code before notes . . . . .	54
7.12.4 Spacing . . . . .	54
7.12.5 Rule . . . . .	55
7.12.6 Maximum height . . . . .	55
7.12.7 Width . . . . .	55
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>56</b>
<b>9 Verse</b>	<b>57</b>
9.1 Basic . . . . .	57
9.2 Define stanza indents . . . . .	57
9.3 Repeating stanza indents . . . . .	57
9.4 Manual stanza indent . . . . .	58
9.5 Stanza breaking . . . . .	58
9.6 Hanging symbol . . . . .	59

9.7 Long verse and page break . . . . .	59
9.8 Content before/after verses . . . . .	59
9.9 Numbering stanza . . . . .	60
9.10 Various tools . . . . .	61
9.11 Notes on empty lines . . . . .	61
<b>10 Grouping</b>	<b>61</b>
<b>11 Cross referencing</b>	<b>62</b>
11.1 Basic use . . . . .	62
11.2 Cross-referencing to a critical note . . . . .	62
11.3 Cross-referencing which return a number in any case . . . . .	62
11.3.1 Cross-referencing in order to define line number of a critical note . . . . .	63
11.4 Not automatic cross-referencing . . . . .	63
11.5 Normal L <sup>A</sup> T <sub>E</sub> X cross-referencing . . . . .	64
11.6 References to start and end lines . . . . .	64
11.6.1 Reference to main text lines . . . . .	64
11.6.2 References to lines that are commented on in the apparatus . . . . .	64
11.6.3 Settings . . . . .	64
11.6.4 Combining multiple references . . . . .	66
11.7 Compatibility with xr package . . . . .	67
<b>12 Sidenotes</b>	<b>67</b>
12.1 Basics . . . . .	67
12.2 Setting . . . . .	67
12.2.1 Width . . . . .	67
12.2.2 Vertical position . . . . .	68
12.2.3 Distance to the main text . . . . .	68
12.2.4 Font . . . . .	68
12.2.5 Separator between notes . . . . .	68
<b>13 Indexing</b>	<b>68</b>
13.1 Basics . . . . .	68
13.2 Use with imakeidx or indextools . . . . .	68
13.3 Referring to critical notes . . . . .	69
13.4 Separator between page and line numbers . . . . .	69
13.5 Using xindy . . . . .	70
13.6 Advanced setting . . . . .	70
<b>14 Glossary</b>	<b>70</b>
14.1 Preamble setting . . . . .	71
14.2 Commands . . . . .	71
<b>15 Tabular material</b>	<b>71</b>

<b>16 Sectioning commands</b>	<b>74</b>
16.1 Sectioning commands without line numbers or critical notes . . . . .	74
16.2 Sectioning commands with line numbering and critical notes . . . . .	75
16.3 Optimization . . . . .	75
<b>17 Quotation environments</b>	<b>76</b>
<b>18 Page breaks</b>	<b>76</b>
18.1 Control page breaking . . . . .	76
18.2 Prevent page break in a long verses . . . . .	76
<b>19 Miscellaneous</b>	<b>77</b>
19.1 Known and suspected limitations . . . . .	77
19.1.1 Non-standard geometry . . . . .	77
19.1.2 floatrow package compatibility . . . . .	77
19.1.3 ‘No room for a new’ . . . . .	77
19.1.4 Marginal notes . . . . .	78
19.1.5 Paragraph shape . . . . .	78
19.1.6 Paragraphed footnotes . . . . .	78
19.1.7 Use with other packages . . . . .	79
19.2 Parallel typesetting . . . . .	80
<b>I Implementation overview</b>	<b>81</b>
<b>II Preliminaries</b>	<b>81</b>
II.1 Links with original edmac . . . . .	81
II.2 Package declaration . . . . .	81
II.3 Package options . . . . .	82
II.3.1 Options of reledpar . . . . .	82
II.3.2 Options of reledmac . . . . .	82
II.4 Loading packages . . . . .	84
II.5 Compatibility with Lua $\TeX$ . . . . .	85
II.6 Boolean flags . . . . .	85
II.7 Messages . . . . .	86
II.8 Gobbling . . . . .	92
II.9 Wrapping content . . . . .	92
II.10 Miscellaneous commands . . . . .	93
II.11 Prepare reledpar . . . . .	93
II.12 Booleans provided by other optional packages which are required in any case . . . . .	94
<b>III Sectioning commands</b>	<b>94</b>
<b>IV List macros</b>	<b>100</b>

<b>V Line counting</b>	<b>101</b>
V.1 Choosing the system of lineation . . . . .	101
V.2 Line number margin . . . . .	103
V.3 Line number initialization and increment . . . . .	104
V.4 Line number locking . . . . .	105
V.5 Line number style . . . . .	106
V.6 Line number printing . . . . .	106
V.7 Line number counters and lists . . . . .	108
V.8 Line number locking counter . . . . .	109
V.9 Line number associated to lemma . . . . .	109
V.10 Reading the line-list file . . . . .	112
V.11 Commands within the line-list file . . . . .	115
V.12 Writing to the line-list file . . . . .	128
<b>VI Marking text for notes</b>	<b>136</b>
VI.1 <code>\edtext</code> itself . . . . .	137
VI.2 Substitute lemma . . . . .	144
VI.3 Substitute line numbers . . . . .	145
VI.4 Lemma disambiguation . . . . .	146
<b>VII Paragraph decomposition and reassembly</b>	<b>153</b>
VII.1 Boxes, counters, <code>\pstart</code> and <code>\pend</code> . . . . .	154
VII.2 Processing one line . . . . .	160
VII.2.1 General process . . . . .	160
VII.2.2 Process for “normal” line . . . . .	161
VII.2.3 Process for line containing <code>\eledsection</code> command . . . . .	163
VII.2.4 Hooks . . . . .	164
VII.2.5 Sidenotes and marginal line number initialization . . . . .	165
<b>VIII Line and page number computation</b>	<b>166</b>
VIII.1 Continuous line numbering between parallel typesetting and normal typesetting . . . . .	169
<b>IX Line number annotation</b>	<b>170</b>
<b>X Line number printing</b>	<b>172</b>
<b>XI Pstart number printing inside</b>	<b>176</b>
<b>XII Restoring footnotes and penalties</b>	<b>178</b>
XII.1 Add insertions to the vertical list . . . . .	178
XII.2 Penalties . . . . .	180
XII.3 Printing leftover notes . . . . .	181
XII.4 Text before notes . . . . .	182

<b>XIII Critical footnotes</b>	<b>183</b>
XIII.1 Fonts . . . . .	184
XIII.2 Individual note options . . . . .	184
XIII.3 Notes language . . . . .	184
XIII.4 General survey of the way we manage notes . . . . .	186
XIII.5 General setup . . . . .	186
XIII.6 Footnotes arrangement . . . . .	187
XIII.6.1 User level macro . . . . .	187
XIII.6.2 Normal footnote . . . . .	187
XIII.6.3 Paragraphed footnotes . . . . .	192
XIII.6.4 Columnar footnotes . . . . .	200
XIII.7 Footnote hanging indent . . . . .	207
XIII.8 Critical notes presentation . . . . .	207
XIII.8.1 Font tools . . . . .	207
XIII.8.2 Pstart number in footnote . . . . .	208
XIII.8.3 Lemma printing . . . . .	208
XIII.8.4 Line number printing . . . . .	209
XIII.8.5 Footnote grouped by line . . . . .	220
<b>XIV Familiar footnotes</b>	<b>222</b>
XIV.1 Adjacent footnotes . . . . .	222
XIV.2 Regular footnotes for numbered texts . . . . .	223
XIV.3 Footnote formats . . . . .	225
XIV.4 Footnote arrangement . . . . .	226
XIV.4.1 User level macro . . . . .	226
XIV.4.2 Normal footnotes . . . . .	226
XIV.4.3 Two columns footnotes . . . . .	232
XIV.4.4 Three columns footnotes . . . . .	234
XIV.4.5 Paragraphed footnotes . . . . .	237
XIV.5 Wrapping footnote marks in hyperlink . . . . .	241
<b>XV Code common to both critical and familiar footnote in normal arrangement</b>	<b>242</b>
<b>XVI Footnotes' width for two columns</b>	<b>242</b>
<b>XVII Footnotes' order</b>	<b>244</b>
<b>XVIII Footnotes' rule</b>	<b>244</b>
<b>XIX Specific skip for first series of footnotes</b>	<b>244</b>
XIX.1 Overview . . . . .	244
XIX.2 User level command . . . . .	245
XIX.3 Internal commands . . . . .	245



<b>XX Endnotes</b>	<b>247</b>
XX.1 Internal commands	247
XX.2 User level commands	251
XX.2.1 Inserting contents to endnotes	251
XX.2.2 Printing endnotes	251
<b>XXI Generate series of notes</b>	<b>260</b>
XXI.1 Test if series is still existing	260
XXI.2 Init specific to <code>reledpar</code>	260
XXI.3 For critical footnotes	261
XXI.3.1 Options	261
XXI.3.2 Create inserts, needed to add notes in foot	263
XXI.3.3 Create commands for critical apparatus, <code>\Afootnote</code> , <code>\Bfootnote</code> etc.	263
XXI.3.4 Set standard display	265
XXI.4 For familiar footnotes	266
XXI.4.1 Options	266
XXI.4.2 Create tools for familiar footnotes ( <code>\footnoteX</code> )	266
XXI.5 The endnotes	269
XXI.5.1 The auxiliary file	269
XXI.5.2 The main macro	269
XXI.5.3 Tools	270
XXI.5.4 Internal commands	270
XXI.5.5 The options	271
XXI.6 Init standards series (A,B,C,D,E)	273
<b>XXII Setting series display</b>	<b>273</b>
XXII.1 Change series order	273
XXII.2 Test series order	273
XXII.2.1 Get the first series	273
XXII.3 Series setting	274
XXII.3.1 General way of working	274
XXII.3.2 Tools to set options	274
XXII.3.3 Tools to generate options commands	276
XXII.3.4 Options for critical notes	277
XXII.3.5 Options for familiar notes	279
XXII.3.6 Options for endnotes	280
XXII.4 Hooks for a particular footnote	282
XXII.5 Alias	284
<b>XXIII Output routine</b>	<b>284</b>
XXIII.1 Extra footnotes output	284
XXIII.2 Patching standard output's commands	287
<b>XXIV Page numbering in parallel typesetting</b>	<b>290</b>

<b>XXV Cross referencing</b>	<b>291</b>
XXV.1 Compatibility with xref . . . . .	308
<b>XXVI Sidenotes</b>	<b>309</b>
<b>XXVII Minipages and such</b>	<b>316</b>
<b>XXVIII Indexing</b>	<b>321</b>
XXVIII.1 Looking on package order . . . . .	322
XXVIII.2 Auxiliary macros for \edindex . . . . .	322
XXVIII.3 Code specific to \edindexin critical footnotes . . . . .	323
XXVIII.4 Analysis of command in indexed text . . . . .	324
XXVIII.5 Code for the formatted index . . . . .	325
XXVIII.6 Main code . . . . .	325
XXVIII.7 Hyperlink . . . . .	327
XXVIII.8 ‘innote’ and ‘notenumber’ option of indextols package . . . . .	330
<b>XXIX Glossaries</b>	<b>330</b>
<b>XXX Verse</b>	<b>332</b>
XXX.1 Hanging symbol management . . . . .	333
XXX.2 Using & character . . . . .	333
XXX.3 Code category setting . . . . .	333
XXX.4 Stanza count and indent . . . . .	334
XXX.5 Numbering stanza . . . . .	335
XXX.6 Stanza number in note . . . . .	336
XXX.7 Main work . . . . .	337
XXX.8 Restore catcode and penalties . . . . .	340
<b>XXXI Apparatus of Manuscripts</b>	<b>340</b>
XXXI.1 User level macro . . . . .	340
XXXI.2 Setting macro . . . . .	341
XXXI.3 Counters and lists . . . . .	342
XXXI.4 Auxiliary file macros . . . . .	343
XXXI.5 Action macro . . . . .	343
XXXI.6 Inserting footnote . . . . .	349
XXXI.7 Other . . . . .	349
<b>XXXII Arrays and tables</b>	<b>349</b>
XXXII.1 Preamble: macro as environment . . . . .	349
XXXII.2 Tabular environments . . . . .	353
XXXII.2.1 Disabling and restoring commands . . . . .	353
XXXII.2.2 Counters, boxes and lengths . . . . .	357
XXXII.2.3 Tabular typesetting . . . . .	361
XXXII.2.4 Environments . . . . .	372
<b>XXXIII Quotation’s commands</b>	<b>372</b>

<b>XXXIV Section's title commands</b>	<b>373</b>
XXXIV.1 Commands to disable some feature . . . . .	373
XXXIV.2 General overview . . . . .	374
XXXIV.3 <code>\beforeeledchapter</code> command . . . . .	374
XXXIV.4 Auxiliary commands . . . . .	375
XXXIV.5 Patching standard commands . . . . .	376
XXXIV.6 Main code of <code>\eledxxx</code> commands . . . . .	381
XXXIV.7 Macros written in the auxiliary file . . . . .	383
<b>XXXV Page breaking or no page breaking depending of specific lines</b>	<b>386</b>
<b>XXXVI Long verse: prevents being separated by a page break</b>	<b>387</b>
<b>XXXVII Tools for hyperref package</b>	<b>388</b>
<b>XXXVIII Compatibility with <code>eledmac</code></b>	<b>389</b>
<b>Appendix A Things to do when changing versions</b>	<b>391</b>
A.1 Migrating from <code>edmac</code> to <code>ledmac</code> . . . . .	391
A.2 Migration from <code>ledmac</code> to <code>eledmac</code> . . . . .	392
A.3 Migration to <code>eledmac</code> 1.5.1 . . . . .	393
A.4 Migration to <code>eledmac</code> 1.12.0 . . . . .	393
A.5 Migration to <code>eledmac</code> 17.1 . . . . .	394
A.6 Migration to <code>eledmac</code> 1.21.0 . . . . .	394
A.6.1 <code>\Xledsetnormalparstuff</code> and <code>\ledsetnormalparstuffX</code> . . .	394
A.6.2 Endnotes . . . . .	394
A.7 Migration to <code>eledmac</code> 1.22.0 . . . . .	394
A.8 Migration to <code>eledmac</code> 1.23.0 . . . . .	394
A.9 Migration from <code>eledmac</code> to <code>reledmac</code> . . . . .	395
A.9.1 Risk of 'no room for a new' . . . . .	395
A.9.2 Multiple indices with memoir . . . . .	395
A.9.3 Deprecated commands and options . . . . .	395
A.9.4 <code>\renewcommand</code> replaced by <code>command</code> . . . . .	396
A.9.5 Commands the names of which have been changed . . . . .	396
A.9.6 Endnotes . . . . .	398
A.9.7 Z Series . . . . .	398
A.9.8 Internal commands . . . . .	398
A.10 Migration to <code>reledmac</code> 2.1.0 . . . . .	398
A.11 Migration to <code>reledmac</code> 2.1.3 . . . . .	398
A.12 Migration to <code>reledmac</code> 2.3.0 . . . . .	398
A.13 Migration to <code>reledmac</code> 2.4.0 . . . . .	399
A.14 Migration to <code>reledmac</code> 2.5.0 . . . . .	399
A.15 Migration to <code>reledmac</code> 2.7.0 . . . . .	399
A.16 Migration to <code>reledmac</code> 2.7.2 . . . . .	399
A.17 Migration to <code>reledmac</code> 2.8.0 . . . . .	399
A.18 Migration to <code>reledmac</code> 2.13.1 . . . . .	399

A.19 Migration to <code>reledmac</code> 2.18.0 . . . . .	400
A.20 Migration to <code>reledmac</code> 2.21.0 . . . . .	400
A.21 Migration to <code>reledmac</code> 2.24.0 . . . . .	400
A.22 Migration to <code>reledmac</code> 2.26.0 . . . . .	400
A.23 Migration to <code>reledmac</code> 2.27.1 . . . . .	400
A.24 Migration to <code>reledmac</code> 2.30.0 . . . . .	400
A.25 Migration to <code>reledmac</code> 2.31.1 . . . . .	401
<b>Appendix B Auxiliary softwares</b>	<b>401</b>
B.1 <i>samewords</i> . . . . .	401
B.2 critical-keys for <i>Emacs</i> . . . . .	401
B.3 Import from TEI . . . . .	401
<b>References</b>	<b>402</b>
<b>Index</b>	<b>402</b>
<b>Change History</b>	<b>454</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 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

---

<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).

combination enables you to recover to some extent the text of each line as ASCII code, facilitating the creation of concordances, an *index verborum*, etc.

As of 1994, we were pleased to be able to say that `edmac` was being used for the real-life book production of several interesting editions, such as the Latin texts of Euclid's *Elements*,<sup>2</sup> an edition of the letters of Nicolaus Copernicus,<sup>3</sup> Simon Bredon's *Arithmetica*,<sup>4</sup> a Latin translation by Plato of Tivoli of an Arabic astrolabe text,<sup>5</sup> a Latin translation of part II of the Arabic *Algebra* by Abū Kāmil Shujā' b. Aslam,<sup>6</sup> the Latin *Rithmarchia* of Werinher von Tegernsee,<sup>7</sup> a middle-Dutch romance epic on the Crusades,<sup>8</sup> a seventeenth-century Hungarian politico-philosophical tract,<sup>9</sup> an anonymous Latin compilation from Hungary entitled *Sermones Compilati in Studio Generali Quinqueecclesiensi in Regno Ungarie*,<sup>10</sup> the collected letters and papers of Leibniz,<sup>11</sup> Theodosius's *Spherics*, the German *Algorismus* of Sacrobosco, the Sanskrit text of the *Kāśikāvṛtti* of Vāmana and Jayāditya,<sup>12</sup> and the English texts of Thomas Middleton's collected works.

### 1.2.2 `ledmac`

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

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

In March 2003 Peter Wilson started an attempt to port `edmac` from TeX to LaTeX. The starting point was `edmac` version 3.16 as documented on 19 July 1994 (available from CTAN). In August 2003 the `tabmac` functions were added; the starting point for these being version 1.0 of October 1996. The `edstanza` (v0.01) functions were added in 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.

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

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

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

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

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

<sup>7</sup>Menso Folkerts, 'Die *Rithmarchia* des Werinher von Tegernsee', *ibid.*

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

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

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

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

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

### 1.2.3 eledmac

Important changes were put in version 1.0, to make ledmac more easily extensible (see 7 p. 42). 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. 392.

### 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. 395.

## 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/>.

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

The reledmac package is a three-pass package like  $\text{\LaTeX}$  itself. Although your textual apparatus and line numbers will be printed on the first run, it takes two more compilations by  $\text{\LaTeX}$  to be sure that everything is correctly placed, and one more if you typeset right-to-left text with  $\text{\XeLaTeX}$ . If you make any subsequent changes altering the number of lines or notes, the input file may similarly require three passes to get everything to the right place. reledmac will tell you that you need to make more runs when it detects changes, but it does not expend the labor to check this thoroughly. If you have problems with a line or two misnumbered at the top of a page, try running  $\text{\LaTeX}$  once or twice more.

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

## 3 Compatibility warning

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

Please read 7.12.6 p. 55.

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  $\TeX$ auxiliary files (`.aux`, `.toc`, ...).

**eledmac-compat** help to migrate from `eledmac` to `reledmac` (see A.9.5 p. 396).

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

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

**noquotation** by default, the quotation environment is redefined within numbered text. You can disable this redefinition with `noquotation` (see 17 p. 76).

**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. 70).



## 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. 75).

**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. 61). 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`.

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. 80).

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  
`\AtEveryPend`    printed before every `\pstart` begins / after every `\pend` ends.

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:

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

```
\AtStartEveryPstart{*}
```

Instead of manually doing

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

### 5.2.5 Numbering paragraphs (\pstart)

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

```
\setcounter{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 reledmac stores a lot of information about line numbers and footnotes in memory  
 \resumenumbering 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

\resumenummering
\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\resumenummering}
```

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

## 5.3 Lineation commands

### 5.3.1 Disabling lineation

`\numberlinefalse` 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 just not want to have the line number in the margins, but want to use critical footnote. In this case, set 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. 23).

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

## 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` `\endsub` You insert the `\startsub` and `\endsub` commands in your text to turn sub-lineation 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` `Xsublinesepside` You can change the separator between the line number and the subline number either 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` `\endlock` 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` `\advanceline` In some cases you may want to modify the line numbers that are automatically cal-

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

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

#### 5.4.4 Line number style

`\linenumberstyle` Line numbers are normally printed as arabic numbers. You can use `\linenumberstyle{<style>}`  
`\sublinenumberstyle` to change the numbering style. `<style>` must be one of:

`Alph` Uppercase letters (A ... Z).

`alph` Lowercase letters (a ... z).

`arabic` Arabic numerals (1, 2, ...)

`Roman` Uppercase Roman numerals (I, II, ...)

`roman` Lowercase Roman numerals (i, ii, ...)

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

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

#### 5.4.5 Skipping and hiding number

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

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

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



## 5.5 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`.
- 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}&
```

---

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

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.
- `\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.

`\Xlinenumannotationposition`  
`\Xendlinenumannotationposition`

### 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. 62).
- `\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. 62). 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
```

### 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 `\Xnoidenticallylinenumannotation[⟨s⟩]`, for critical footnotes or `\Xendnoidenticallylinenumannotation[⟨s⟩]`

### 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⟩}`

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

### 5.5.6 Annotations in the apparatus

Some others options for the annotation in the apparatus are described on 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  $\TeX$ .

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.		
		1 Smith] Jones C, D.

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

The `⟨lemma⟩` may contain further `\edtext` commands. Nesting makes it possible to print an explanatory note on a long passage together with notes on variants for individual words within the passage. For example:

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

However, `\edtext` cannot handle overlapping but unnested notes—for example, one note covering lines 10–15, and another covering 12–18; 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.

`\Bfootnote` These are the main macros that you will use to construct the critical apparatus of your text.

`\Cfootnote`

`\Dfootnote`

`\Efootnote`

If you need more series of critical notes, please look at 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).
- `linangesep=⟨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`

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

`\Eendnote` `\Aendnote[⟨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 `\Xendinplaceoflemmaseparator` to set it (7.4.2 p. 50).
- `linangesep=⟨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

---

<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.

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:

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

`\linenum` You can use `\linenum{<arg>}` to change the line numbers passed to the notes. `<arg>` actually consist of seven parameters: the page, line, and sub-line number for the start of the lemma; the same three numbers for the end of the lemma; and the font specifier for the lemma. As the argument to `\linenum`, you specify those seven parameters in that order, separated by vertical bars (the `|` character). I.e.

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

However, you can retain the value computed by `reledmac` for any number by simply omitting it; and you can omit a sequence of vertical bars at the end of the argument. For example, `\linenum{|||23}` changes only the ending page number of the current lemma.

This command does not change the marginal line numbers in any way; it just changes the numbers passed to the notes. Its use comes in situations that `\edtext` has trouble dealing with for whatever reason. If you need notes for overlapping passages that aren't nested, for instance, you can use `\lemma` and `\linenum` to generate such notes despite the limitations of `\edtext`. If the `<lemma>` argument to `\edtext` is extremely long, you may run out of memory; here again you can specify a note with an abbreviated lemma using `\lemma` and `\linenum`. The numbers used in `\linenum` need



not be entered manually; you can use the ‘x-’ symbolic cross-referencing commands below (11 p. 62) 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
```

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.

<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`.

### 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 numbers:

- 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 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}}.%
```

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

`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`.

The diagram consists of nested rectangular boxes. The outermost box contains the text `1inlemma` followed by a smaller box. This inner box contains `23` followed by the text `1...3 4 51`. The box containing `23` is itself a smaller box, representing a nested `\edtext` level. The subscripts (`inlemma`, `3`, `1`) represent the content of the `\lemma` command for each respective `\edtext` level.

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)
}%
{%
  1      A (1) A (2) A (3) A (4) A (5)
  \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]%
}

```

### 6.3.5 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] causam rei B

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

### 6.3.6 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}%
}
```

### 6.3.7 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
```

### 6.3.8 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  $\text{\LaTeX}$  argumentless macro, it will gobble the following space. To keep that space, you have to either append a backslash followed by a space or `{}` to `\stopmsdata`, like so:

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

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

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

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

#### 6.4.2 Layout of the apparatus of manuscripts

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

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

#### 6.4.3 Settings

As the apparatus of manuscripts technically consists of first-level critical notes (‘A’ series), any setting available for critical notes can be applied (7 p. 42). 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`

`\footnoteB`

`\footnoteC`

`\footnoteD`

`\footnoteE`

As well as the standard  $\text{\LaTeX}$  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`

`\bodyfootmarkA`

`\footfootmarkA`

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:

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

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

`\newcommand*{\bodyfootmarkA}{%`

`\hbox{\textsuperscript{\normalfont\@nameuse{@thefnmarkA}}}}`

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

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

There are similar command triples for the other series.

### 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  $\text{\LaTeX}$ 's command `\footnotemark` and `\footnotetext`. For example, to use with `\uline`, do:

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

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

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

## 6.7 Changing series

### 6.7.1 Create a new series

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

### 6.7.2 Delete series

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

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

### 6.7.3 Series order

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

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

## 6.8 Position of critical and familiar footnotes

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

```
\fnpos{critical-familiar}
```

Or, if you want a minipage or ledgroup to have critical footnotes after familiar footnotes, use:

```
\mpfnpos{familiar-critical}
```

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

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

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

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

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

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

## 7 Apparatus customization

Some commands can be used to change the display of the footnotes. All can have an optional argument  $[\langle s \rangle]$ , which is the letter of the series — or a list of letters separated by comma — depending on which option is applied. If the optional argument is omitted or empty, the setting will apply to the entire series.

When a length, noted  $\langle l \rangle$ , is used, it can be stretchable:  $a$  plus  $b$  minus  $c$ . The final length  $m$  is calculated by  $\LaTeX$  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.

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

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[⟨s⟩]{⟨a⟩}` to change the arrangement of the  $\langle s \rangle$  series of critical footnotes and `\arrangementX[⟨s⟩]{⟨a⟩}` 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. 78.
- `twocol` formats them as separate paragraphs, but in two columns;
- `threecol`, in three columns.
- `normal`, restore normal arrangement.

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

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

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

`\hsize` has been set for the pages that use this series of notes; otherwise T<sub>E</sub>X will try to put too many or too few of these notes on each page. If you need to change the `\hsize` within the document, call the arrangement macro again afterwards to take account of the new value.

## 7.2 Control line number printing

### 7.2.1 Print line number only at first time

`\Xnumberonlyfirstinline`

By default, the line number is printed in every note. If you want to print it only the first time for a given line number (i.e., 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 `⟨symbol⟩` 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 at first time

`\Xlinenumannotationonlyfirst`

By default, the line number annotation (?? p. ??) 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 followings points:

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

`\Xlinenumannotationonlyfirstintwo`

Suppose you have a lemma on annoation 1 and a lemma between annotation 1 and annotation 3. With `\Xlinenumannotationonlyfirst`, the second lemma is considered to be with the same line annotation as the first lemma. But if you use both `\Xlinenumannotationonlyfirst[⟨s⟩]` and `\Xlinenumannotationonlyfirstintwo[⟨s⟩]`, a distinction is made.

`\Xsymlinenumannotation`

For setting 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.

`\Xendlinenumannotationonlyfirst`

`\Xendlinenumannotationonlyfirstintwo`

`\Xendsymlinenumannotation`

For endnotes, `\Xendlinenumannotationonlyfirst`; `\Xendlinenumannotationonlyfirstintwo` and `\Xendsymlinenumannotation` are the equivalents of

`\Xlinenumannotationonlyfirst`; `\Xlinenumannotationonlyfirstintwo` and `\Xsymlinenumannot`.

### 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:

- `\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

`\Xbeforenumber`

`\Xbeforenumber[⟨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`

`\Xendlineprefixsingle`

`\Xendafterpagenumber[⟨s⟩]{⟨text⟩}` defines the text after the page number in endnotes. Default value is ) (open parenthesis followed by a single space). `\Xendlineprefixsingle[⟨s⟩]{⟨text⟩}` defines the text before the line number in endnotes when there is only one line. Default value is empty.

`\Xendlineprefixmore`

`\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[ $\langle s \rangle$ ]{ $\langle text \rangle$ }`, and with `\Xendlinerangeseparator[ $\langle s \rangle$ ]{ $\langle text \rangle$ }` 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[ $\langle s \rangle$ ]{ $\langle text \rangle$ }` and `\Xmorethantwolines[ $\langle s \rangle$ ]{ $\langle text \rangle$ }`. The  $\langle text \rangle$  argument of `\Xtwolines` will be printed if the lemma is on two lines, and the  $\langle text \rangle$  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  $\langle text \rangle$  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[ $\langle series \rangle$ ]`.

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[ $\langle series \rangle$ ]`.

You can disable `\Xtwolines` and related for a specific note by using the ‘[fulllines]’ argument in the note macro cf. 6.2.2 p. 30.

`\Xendtwolines`  
`\Xendmorethantwolines`  
`\Xendtwolinesbutnotmore`

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

### 7.2.7 Disable line number

`\Xnonumber`  
`\Xendnonumber`

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

### 7.2.8 Printing pstart number

`\Xpstart`

You can use `\Xpstart[ $\langle s \rangle$ ]` if you want to print the pstart number in the footnote, before the line and subline number. Note that when you change the lineation system, the option is automatically switched :

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

`\Xpstarteverytime` By default, the pstart number is printed only in the part of text where you have called `\numberpstarttrue`. We don't know why you would like to print the pstart number in the notes and not in the main text. However, if you want to do it, you can call `\Xpstarteverytime[⟨s⟩]`. In this case, the pstart number will be printed every time in footnote.

`\Xonlypstart` In combination with `\Xpstart`, you can use `\Xonlypstart[⟨s⟩]` if you want to print only the pstart number in the footnote, and not the line and subline number.

### 7.2.9 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.

### 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` `\Xendafternumber` are the equivalents of `\Xbeforenumber`  
`\Xendafternumber`

and `\Xafterenumber` for endnotes.

`\Xnonbreakableafternumber`

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

### 7.2.13 Space around line symbol

`\Xbeforemylinenum`

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

`\Xaftersymylinenum`

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

`\Xendbeforemylinenum`

`\Xendaftersymylinenum`

`\Xendbeforemylinenum` and `\Xendaftersymylinenum` are the equivalents of `\Xbeforemylinenum` and `\Xaftersymylinenum` 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` [ $\langle s \rangle$ ] { $\langle l \rangle$ }. The default value is 1 em.

`\Xendinplaceofnumber`

`\Xendinplaceofnumber` [ $\langle s \rangle$ ] { $\langle l \rangle$ } 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` [ $\langle s \rangle$ ] { $\langle l \rangle$ } 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}
```

`\Xboxsymylinenum`

`\Xboxsymylinenum` [ $\langle s \rangle$ ] { $\langle l \rangle$ } is the same as `\Xboxlinenum` but for the line number symbol.

`\Xendboxsymylinenum`

`\Xendboxsymylinenum` [ $\langle s \rangle$ ] { $\langle l \rangle$ } is the same as `\Xboxsymylinenum` 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` [ $\langle s \rangle$ ] { $\langle text \rangle$ } where  $\langle text \rangle$  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  $\langle l \rangle$ . 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  $\langle l \rangle$ . 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⟩}`, `\Xendboxlinenumalign` `\Xendboxendlinenum[⟨s⟩]{⟨l⟩}` are the same as, respectively, `\Xboxlinenum` and `\Xboxendlinenumalign`, `\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 `\Xendnotenumfont` font setting.

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

`\Xendbhookinplaceofnumber` `\Xendbhookinplaceofnumber[⟨s⟩]{⟨code⟩}` is used to execute code before space or symbol which replace line number in endnotes. The code is executed before the `\Xendbeforelymlinenumber` space and before the `\Xendnotenumfont` font setting.

`\Xendahookinplaceofnumber` `\Xendahookinplaceofnumber[⟨s⟩]{⟨code⟩}` is used to execute code after space or symbol which replace line number in endnotes. The code is executed after the `\Xendaftersymlymlinenumber` space.

## 7.4 Separator between the lemma and the note

### 7.4.1 For footnotes

`\Xlemmaseparator` By default, in a footnote, the separator between the lemma and the note is a right bracket (`\rbracket`)<sup>20</sup>. You can use `\Xlemmaseparator[⟨s⟩]{⟨Xlemmaseparator⟩}` to change it. The optional argument can be used to specify the series in which it is used. Note that there is a non-breakable space between the lemma and the separator, but a **breakable** space between the separator and the following text.

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

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

`\Xnolemmaseparator`

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

`\Xinplaceoflemmaseparator`

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

### 7.4.2 For endnotes

`\Xendlemmaseparator`

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

Note that there is a non-breakable space between the lemma and the separator, but a **breakable** space between the separator and the following text.

`\Xendbeforelemmaseparator`

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

`\Xendafterlemmaseparator`

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

`\Xendinplaceoflemmaseparator`

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

## 7.5 Font style

### 7.5.1 For line number

`\Xnotenumfont`

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

`\Xendnotenumfont`

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

`\notenumfontX`

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

### 7.5.2 For the lemma

`\Xlemmadisablefontselection`

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 `\Xlemmadisablefontselection[⟨s⟩]` command allows to disable it for a specific series.

`\Xendlemmadisablefontselection`

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

---

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

command allows `\Xendlemmadisablefontselection[⟨s⟩]` to disable it for a specific series.

`\Xlemmafont`      Use `\Xlemmafont[⟨s⟩]{⟨cmd⟩}` to apply a  $\TeX$  font command to the lemma. For  
`\Xendlemmafont`    example, to have boldface lemma:

`\Xlemmafont{\bfseries}`

`\Xendlemmafont[⟨s⟩]{⟨cmd⟩}` is the same for endnotes.

### 7.5.3 For all notes

`\Xnotefontsize`    `\Xnotefontsize[⟨s⟩]{⟨command⟩}` is used to define the font size of critical footnotes of the series. The default value is `\footnotesize`. The `⟨command⟩` must not be a size in pt, but a standard  $\TeX$  size, like `\small`.

`\notefontsizeX`    `\notefontsizeX[⟨s⟩]{⟨command⟩}` is used to define the font size of familiar footnotes of the series. The default value is `\footnotesize`. The `⟨command⟩` must not be a size in pt, but a standard  $\TeX$  size, like `\small`.

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

## 7.6 Wrapping notes

### 7.6.1 Wrapping lemmas

`\Xwraplemma`      `\Xwraplemma[⟨s⟩]{⟨cmd⟩}` is used to wrap, in the footnote, the lemma in a  $\TeX$  command. For example, with the  `bidi`  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  $\TeX$  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`  
`beforeinsertingX`

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

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`

`\colalignX`

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

`<code>` must be one of the following command:

`\justifying` to have text justified, as usual with  $\text{\LaTeX}$ . You can also let `<code>` empty.

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

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

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

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

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

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

### 7.10.2 Size of the columns

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

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

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

### 7.11.1 Mark separation of notes

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

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

### 7.11.2 Ragged text

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

## 7.12 Options for block of notes

### 7.12.1 Grouping notes by line

<code>\Xgroupbyline</code>	If you do not use <code>\Xarrangement{paragraph}</code> , you may want to group all the crit-
----------------------------	---

ical 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  $\TeX$  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  $\TeX$  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  $\LaTeX$  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  $\LaTeX$  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 *todolist*.

### 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 with of the block of notes. If you want to change the width of each column when typesetting notes in columns, use `\Xhsizetwocol`, `\Xhsizethreecol`, `\hsizetwocolX`, `\hsizethreecolX`, see 7.10.2 p. 53.



### 7.13 Footnotes and the `reledpar` columns

`\Xnoteswidthliketwocolumns`  
`\noteswidthliketwocolumnsX`

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

- `\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 a endnote series by using `\Xendafternote[⟨s⟩]{⟨l⟩}`. The default value is `1em plus .4em minus .4em`.

`\Xendsep`

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

## 8 Fonts

One of the most important features of the appearance of the notes, and indeed of your whole document, will be the fonts used. We will first describe the commands that give you control over the use of fonts in the different structural elements of the document, especially within the notes, and then in subsequent sections specify how these commands are used.

`\numlabfont`

Line numbers for the main text are usually printed in a smaller font in the margin. The `\numlabfont` macro is provided as a standard name for that font: it is initially defined as

```
\newcommand{\numlabfont}{\normalfont\scriptsize}
```

You might wish to use a different font if, for example, you preferred to have these line numbers printed using old-style numerals.

`\select@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. 9.8 p. 59)

### 9.2 Define stanza indents

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

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

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

The numerical entries must be whole numbers, 0 or greater, separated by commas without embedded spaces. The first entry gives the hanging indentation to be used if the stanza line requires more than one print line.

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

If you want the hanging verse to be flush right, you can use `\sethanginsymbol:` see p. 9.6 p. 59.

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 `stanzaindentsrepetition` counter at  $n$ . For example:

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

is like

```
\setstanzaindents{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. 393.**

If you don't use the `stanzaindentsrepetition` 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{stanzaindentsrepetition}{0}
```

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

## 9.4 Manual stanza indent

```
\stanzaindent
\stanzaindent*
```

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

If you use the mechanism of indent repetition, the next verse will be printed as it should be even if the current verse would have its normal indent value. In other words, using `\stanzaindent` in a verse does not shift the indent repetition.

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

## 9.5 Stanza breaking

```
\setstanzapenalties
```

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

The command

```
\setstanzapenalties{1,5000,10100,5000,0}
```

results in a penalty of 5000 being placed after the first and third lines of the stanza, and a penalty of  $-100$  after the second.

The first entry “1” is a control value. If it is zero, then no penalties are passed on to  $\TeX$ , which is the default. Values between 0 and 10000 are penalty values; values between 10001 and 20000 have 10000 subtracted and the result is given as a negative penalty. The mechanism used for indentations and penalties requires unsigned values

less than 32768. No penalty is placed after the last line, so the final ,0 in 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.

## 9.6 Hanging symbol

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

`\sethangingsymbol`

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

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

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

## 9.7 Long verse and page break

If you want to prevent page breaks inside long verses, use the option `nopbinverse` when loading package, or use `\lednopbinversetrue`. Read 18.2 p. 76 for further details.

## 9.8 Content before/after verses

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

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

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

`\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).

## 9.9 Numbering stanza

`\numberstanzatrue`  
`\numberstanzafalse`

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

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}}%
}
```

`\stanzanumwrapper`

You can change the value of the `stanza` counter with the usual commands of  $\TeX$ .

You can redefine `\stanzanumwrapper` in order to modify the way the stanza number is inserted in the flow of text. Default value is:

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

## 9.10 Various tools

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

## 9.11 Notes on empty lines

Since v2.3.0 of `reledmac`, empty lines when typesetting verses no longer produce new paragraphs, and consequently, do not insert vertical spaces. Use optional argument of `\stanza` or `\newverse` to insert vertical space (9.8 p. 59).

## 10 Grouping

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

`minipage` You can put numbered text with critical footnotes in a `minipage` and the footnotes are set at the end of the `minipage`.

You can also put familiar footnotes (see section 6.5) in a `minipage` but unlike with `\footnote` the numbering scheme is unaltered.

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

`ledgroupsize` The `ledgroupsize` environment is similar to `ledgroup` except that you must specify a width for the environment, as with a `minipage`.

`\begin{ledgroupsize}[⟨pos⟩]{⟨width⟩}`.

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

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

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

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

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

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

## 11 Cross referencing

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

### 11.1 Basic use

`\edlabel` First you place a label in the text using the command `\edlabel{<lab>}`. `<lab>` can be almost anything you like, including letters, numbers, punctuation, or a combination—anything but spaces; you might type `\edlabel{toves-3}`, for example.<sup>21</sup>

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

`\edlineref`

`\sublineref`

`\pstartref`

`\annotationref`

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

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

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

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

### 11.2 Cross-referencing to a critical note

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

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

If you want to refer to starting and ending lines, you should use `\appref` and related tools (11.6.2 p. 64).

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

`\xpageref` 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 `ℒTEX` is looking for a number, such a warning

`\xlineref`

`\xsublineref`

`\xpstartref`

`\xannotationref`

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

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 $\text{\LaTeX}$ cross-referencing

`\label`      The normal `\label`, `\ref` and `\pageref` macros may be used within numbered text,  
`\ref`      and operate in the familiar fashion.  
`\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 clas-  
sical `\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  
`\apprefwithpage` lines. The first one will print the lines as they are printed in the critical footnotes, while  
the second will print the lines as they are printed in endnotes.

### 11.6.3 Settings

`\setapprefprefixsingle`      **Specific to these tools** If you use `\apprefprefixsingle{<prefix>}`, `<prefix>` will be  
`\setapprefprefixmore`



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 and \setSerefprefixmore are similar for \Seref command.

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 \XendbeforepagenumberSerefonly (see 11.6.3 p. 65). 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`      The package's default setting is  
`\ledsidenote`      `\sidenotemargin{right}`  
`\sidenotemargin`

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  $\TeX$  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 them to be placed to align with the first line of the note to which it refers, use `\leftnoteupfalse` (for left note) and/or `\rightnoteupfalse` (for right note).

### 12.2.3 Distance to the main text

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

### 12.2.4 Font

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

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

These can of course be changed to suit.

### 12.2.5 Separator between notes

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

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

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. You can redefine this suffix by redefining the command `\ledinnotemark`. Its actual definition is:

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

Where `#1` stands for the reference.

### 13.4 Separator between page and line numbers

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

- is the default separator used by the `MAKEINDEX` program.

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

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

However, you also have to configure your `.ist` index style file. For example, if you use `:` as separator<sup>22</sup>.

```
page_compositor ":"
```

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

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

### 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:

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

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

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

## 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. 69).

## 14.2 Commands

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

`\edgls[⟨options⟩]{⟨label⟩}[⟨insert⟩]`

## 15 Tabular material

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

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 environments are centered with respect to the surrounding text.

```

edarrayl
edarrayc
edarrayr
edtabularl
edtabularc
edtabularr
\begin{edtabularc}
1 & 2 & 3 \\
a & bb & ccc \\
AAA & BB & C
\end{edtabularc}

```

1	2	3
a	bb	ccc
AAA	BB	C

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

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

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  
`\spreadtext` calculation of column widths. `\spreadtext{<text>}` is the analagous command for use  
in edtabular environments.

<pre> \begin{edarrayl} 1 &amp; 2 &amp; &amp; 3 &amp; &amp; 4 &amp; \\ &amp; \spreadmath{F+G+C} &amp; &amp; &amp; &amp; &amp; \\ a &amp; &amp; bb &amp; &amp; ccc &amp; &amp; dddd \end{edarrayl} </pre>	<pre> 1 2 3 4 F + G + C a bb ccc dddd </pre>
---	--

`\edrowfill` The macro `\edrowfill{<start>}{<end>}{<fill>}` fills columns number `<start>` to `<end>` inclusive with `<fill>`. The `<fill>` 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          & 2      & 3      & 4      & 5 \\
Q          & & fd & h      & qwertziohg \\
v          & wptz   & x      & y      & vb \\
g          & nnn    & \edrowfill{3}{5}{\upbracefill} & & \\
\edrowfill{1}{3}{\downbracefill} & & & pq & dgh \\
k          & & l      & co     & ghweropjklmnbvcxys \\
1          & 2 & 3 & \edrowfill{4}{5}{\hrulefill} & 
\end{edtabularr}

```



1	2	3	4	5
Q		fd	h	qwertziohg
v	wptz	x	y	vb
g	nnn	$\overbrace{\hspace{10em}}$		
k	$\overbrace{\hspace{10em}}$		pq	dgh
1	2	3	co	ghweropjklmnbvcxys
			$\underbrace{\hspace{10em}}$	

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 `\upbracketfill` 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}
```

1	2	3	4
$a$	$\lrcorner$		$d$
$A$	$B$	$C$	$D$

`\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, `\edaftertab` typesets  $\langle text \rangle$  left justified before the  $\langle entry \rangle$ . Similarly `\edaftertab{ $\langle entry \rangle$ }{ $\langle text \rangle$ }`,

where  $\langle entry \rangle$  is an entry in the rightmost column, typesets  $\langle text \rangle$  right justified after the  $\langle entry \rangle$ .

For example:

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

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

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

`\edvertdots`

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

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

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

## 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[⟨text⟩]{⟨critical text⟩}`,
- `\eledchapter*`,
- `\eledsection[⟨text⟩]{⟨critical text⟩}`,
- `\eledsection*`,
- `\eledsubsection[⟨text⟩]{⟨critical text⟩}`,
- `\eledsubsection*`,
- `\eledsubsubsection[⟨text⟩]{⟨critical text⟩}`,
- `\eledsubsubsection*`.

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

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

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

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

## 16.3 Optimization

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

## 17 Quotation environments

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

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

## 18 Page breaks

### 18.1 Control page breaking

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

`\ledpb`  
`\lednopb`

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

**These commands have effect only at the second run.**

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.

`\ledpbsetting`

If you are using `reledpar` to typeset parallel pages, you must use `\lednopb` on both sides in the two corresponding lines. This is especially important when you are using stanzas; otherwise, the pages will be out of sync.

### 18.2 Prevent page break in a long verses

`\lednopbinversetrue`

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

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

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

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

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

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

### 19.1 Known and suspected limitations

#### 19.1.1 Non-standard geometry

If you use classes other than `article` or `book`, or if you use the `geometry` package, you should use `maxnotesX` and/or `\Xmaxnotes` 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. 75 in order to know which are the available options.

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

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

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

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

#### 19.1.4 Marginal notes

In general, `reledmac`'s system for adding marginal line numbers breaks anything that makes direct use of the  $\LaTeX$  insert system, which includes `marginpars`, footnotes and floats.

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

#### 19.1.5 Paragraph shape

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

`\ballast`

$\LaTeX$  is a three-pass system, but even after a document has been processed three times, there are some tricky situations in which the page breaks decided by  $\TeX$  never settle down. At each successive run, `reledmac` may oscillate between two different sets of page decisions. To stop this happening, should it arise, Wayne Sullivan suggested the inclusion of the quantity `\ballast`. The amount of `\ballast` will be subtracted from the penalties which apply to the page breaks calculated on the *previous* run through  $\TeX$ , thus reinforcing these breaks. So if you find your page breaks oscillating, insert `\setcounter{ballast}{100}`

or some such figure, and with any luck the page breaks will settle down. Luckily, this problem does not crop up at all often.

#### 19.1.6 Paragraphed footnotes

The restriction on explicit line-breaking in paragraphed footnotes, mentioned on 7.1 p. 43, and described in more detail on XIII.6.3 p. 195, really is a nuisance if that is something you need to do. There are some possible solutions, described by Michael Downes, but this area remains unsatisfactory.

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

`\footfudgefiddle`

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

```
\renewcommand{\footfudgefiddle}{68}
```

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

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

Finally, in many cases you should use `\Xmaxhnotes` and / or `\maxhnotesX` (7.12.6 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
```

---

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

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

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

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

## 19.2 Parallel typesetting

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

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



## I Implementation overview

We present the `reledmac` code in roughly the order in which it is used during a run of  $\TeX$ . 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}[2019/07/15 v2.32.0 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.

`\ifsameparallelpagenumbers`

`\ifprevpgnotnumbered`

`\ifledfinal`

`\ifnocritical@`

`\if@noeled@sec`

`\ifnoend@`

`\ifnofamiliar@`

`\ifnoledgroup@`

`\ifparapparatus@`

`\ifnoquotation@`

`\iflednopbinverse`

`\ifwidthliketwocolumns`

`\ifxindy@`

`\ifxindyhyperref@`

`\ifeledmaccompat@`

`\ifnoresetlinenumannotation@`

### 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\ifsameparallelpagenumbers%
9 \newif\ifprevpgnotnumbered%%
10 %

```

### II.3.2 Options of reledmac

```

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 }%

```

```

33 \newif\ifnoquotation@
34 \DeclareOptionX{noquotation}{\noquotation@true}
35
36
37 \newif\ifledfinal
38 \DeclareOptionX{final}{\ledfinaltrue}
39 \DeclareOptionX{draft}{\ledfinalfalse}
40 \ExecuteOptionsX{final}
41
42 \newif\ifparapparatus@
43 \DeclareOptionX{parapparatus}{\parapparatus@true}
44
45 \newif\iflednopbinverse
46 \DeclareOptionX{nopbinverse}{\lednopbinversetrue}
47
48 \newif\ifwidthliketwocolumns%
49 \DeclareOptionX{widthliketwocolumns}{\widthliketwocolumnstrue}%
50
51 \newif\ifcontinuousnumberingwithcolumns
52 \DeclareOptionX{continuousnumberingwithcolumns}{\
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 %

```

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.

```

89 \ProcessOptionsX*\relax
90
91 %

```

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

```

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

```

## 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.

```

110 \ifx\directlua\undefined\else%
111   \directlua{tex.enableprimitives("",{"texdir","pardir","bodydir"})}
112 \fi
113 %

```

## II.6 Boolean flags

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

```

114 \newif\ifl@dmemoir
115 \@ifclassloaded{memoir}{\l@dmemoirtrue}{\l@dmemoirfalse}
116
117 %

```

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

```

118 \newif\if@ledgroup%
119 %

```

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

```

120 \newif\ifl@imakeidx
121 \@ifpackageloaded{imakeidx}{\l@imakeidxtrue}{}%False is the default value
122 %

```

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

```

123 \newif\ifl@indextools%
124 \@ifpackageloaded{indextools}{%
125   \l@indextoolstrue%
126   \l@imakeidxtrue%
127   \let\imki@wrindexentry\indtl@wrindexentry%
128 }{}%
129 %

```

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.

```

130 \newif\ifl@footmisc
131 \@ifpackageloaded{footmisc}{\l@footmisctrue}{}%False is the default value
132 %

```

`\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.

```

133 \ifdef{\if@RTL}{\newif\if@RTL}
134 %

```

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

```

135 \newif\if@firstlineofpage%
136 \newif\if@firstlineofpageR%
137 %

```

## II.7 Messages

All the messages are grouped here as macros. This saves T<sub>E</sub>X's memory when the same message is repeated and also lets them be edited easily.

**\reledmac@warning** Write a warning message.

```

138 \newcommand{\reledmac@warning}[1]{\PackageWarning{reledmac}{#1}}
139 %

```

**\reledmac@error** Write an error message.

```

140 \newcommand{\reledmac@error}[2]{\PackageError{reledmac}{#1}{#2}}
141 %

```

```

\led@err@NumberingStarted 142 \newcommand*\led@err@NumberingStarted{%

```

```

\led@err@NumberingNotStarted 143 \reledmac@error{Numbering has already been started}{\@ehc}}

```

```

\led@err@NumberingShouldHaveStarted 144 \newcommand*\led@err@NumberingShouldHaveStarted{%

```

```

145 \reledmac@error{Numbering was not started}{\@ehc}}

```

```

146 \newcommand*\led@err@NumberingShouldHaveStarted{%

```

```

147 \reledmac@error{Numbering should already have been started}{\@ehc}}

```

```

148 %

```

```

\led@err@edtextoutsidestart 149 \newcommand*\led@err@edtextoutsidestart{%

```

```

\led@err@PstartInEdtext 150 \reledmac@error{\string\edtext\space outside numbered paragraph (\pstart\

```

```

\led@err@PendInEdtext \space ... \space\pend)}{\@ehc}}%

```

```

151

```

```

152 \newcommand{\led@err@PstartInEdtext}[1]{%

```

```

153 \reledmac@error{\string\pstart\space in \string\edtext\space #1 argument
154 }{\@ehc}%

```

```

155 %

```

```

156

```

```

157 \newcommand{\led@err@PendInEdtext}[1]{%

```

```

158 \reledmac@error{\string\pend\space in \string\edtext\space #1 argument}{\
159 \@ehc}%

```

```

158 %

```

```

159 %

```

```

\led@mess@NotesChanged60 \newcommand*{\led@mess@NotesChanged}{%
161   \typeout{reledmac reminder: }%
162   \typeout{ The number of the footnotes in this section
163     has changed since the last run.}%
164   \typeout{ You will need to run LaTeX two more times
165     before the footnote placement}%
166   \typeout{ and line numbering in this section are
167     correct.}}
168   %

\led@mess@SectionContinued69 \newcommand*{\led@mess@SectionContinued}[1]{%
170   \message{Section #1 (continuing the previous section)}}
171   %

\led@err@LineationInNumbered72 \newcommand*{\led@err@LineationInNumbered}{%
173   \reledmac@error{You can't use \string\lineation\space within
174     a numbered section}{\@ehc}}
175   %

\led@warn@BadLineation76 \newcommand*{\led@warn@BadLineation}{%
\led@warn@BadLinenummargin77   \reledmac@warning{Bad \string\lineation\space argument}}
\led@warn@BadLockdisp78 \newcommand*{\led@warn@BadLinenummargin}{%
\led@warn@BadSublockdisp79   \reledmac@warning{Bad \string\linenummargin\space argument}}
180   \newcommand*{\led@warn@BadLockdisp}{%
181     \reledmac@warning{Bad \string\lockdisp\space argument}}
182   \newcommand*{\led@warn@BadSublockdisp}{%
183     \reledmac@warning{Bad \string\sublockdisp\space argument}}
184   %

\led@warn@NoFile85 \newcommand*{\led@warn@NoFile}[1]{%
186   \reledmac@warning{File `#1' not found}}
187   %

\led@warn@LineFileObsolete88 \newcommand*{\led@warn@Obsolete}[1]{%
189   \reledmac@warning{Line-list file #1 was obsolete. We have not read it.
Please run LaTeX again.}}
190   %

\led@warn@BadAdvancelineSubline91 \newcommand*{\led@warn@BadAdvancelineSubline}{%
\led@warn@BadAdvancelineLine92   \reledmac@warning{\string\advanceline\space produced a sub-line
193     number less than zero.}}
194   \newcommand*{\led@warn@BadAdvancelineLine}{%
195     \reledmac@warning{\string\advanceline\space produced a line
196     number less than zero.}}
197   %

```

```

\led@warn@BadSetline 98 \newcommand*\led@warn@BadSetline}{%
\led@warn@BadSetlinenum 99 \reledmac@warning{Bad \string\setline\space argument}}
200 \newcommand*\led@warn@BadSetlinenum}{%
201 \reledmac@warning{Bad \string\setlinenum\space argument}}
202 %

\led@err@PstartNotNumbered 203 \newcommand*\led@err@PstartNotNumbered}{%
\led@err@PstartInPstart 204 \reledmac@error{\string\pstart\space must be used within a
\led@err@PendNotNumbered 205 numbered section %
\led@err@PendNoPstart 206 (\string\beginnumbering\space ...\space\string\
\led@err@AutoparNotNumbered endnumbering)}{\@ehc}}%
\led@err@NumberingWithoutPstart 207 \newcommand*\led@err@PstartInPstart}{%
208 \reledmac@error{\string\pstart\space encountered while another
209 \string\pstart\space was in effect}{\@ehc}}
210 \newcommand*\led@err@PendNotNumbered}{%
211 \reledmac@error{\string\pend\space must be used within a
212 numbered section}{\@ehc}}
213 \newcommand*\led@err@PendNoPstart}{%
214 \reledmac@error{\string\pend\space must follow a \string\pstart}{\@ehc}}
215 \newcommand*\led@err@AutoparNotNumbered}{%
216 \reledmac@error{\string\autopar\space must be used within a
217 numbered section}{\@ehc}}
218 \newcommand*\led@err@NumberingWithoutPstart}{%
219 \reledmac@error{\string\beginnumbering...\string\endnumbering\space
without \string\pstart}{\@ehc}}%
220 %

\led@warn@BadAction 221 \newcommand*\led@warn@BadAction}{%
222 \reledmac@warning{Bad action code, value \next@action.}}
223 %

\led@warn@DuplicateLabel 224 \newcommand*\led@warn@DuplicateLabel}[1]{%
\led@warn@AppLabelOutSecondArgEdtext 225 \reledmac@warning{Duplicate definition of label `#1'\@gobble}%
\led@warn@RefUndefined 226 \@latex@warning@no@line{Label `#1' multiply defined}%
\led@warn@RefUndefined 227 }%
228 \newcommand*\led@warn@AppLabelOutSecondArgEdtext}[1]{%
229 \reledmac@warning{\string\applabel\space outside of the second argument
of an \string\edtext\space `#1' on page \thepage.}}%
230 \newcommand*\led@warn@RefUndefined}[1]{%
231 \G@refundefinedtrue%
232 \reledmac@warning{Reference `#1' on page \thepage\space undefined.%
233 Using `000'.}%
234 \@latex@warning{Reference `#1' undefined\on@line}%
235 }%
236 \newcommand*\led@warn@pairRefUndefined}[1]{%
237 \G@refundefinedtrue%

```



```

238 \reledmac@warning{Reference `#1:start' and/or `#1:end' on page \thepage\
space undefined.
239         Using `??'.}%
240 \@latex@warning{Reference `#1:start' and/or `#1:end' undefined\on@line}%
241 }
242 %

```

```

\led@warn@NoMarginpars43 \newcommand*\led@warn@NoMarginpars{%
244 \reledmac@warning{You can't use \string\marginpar\space in numbered text
}}
245 %

```

```

\led@warn@BadSidenotemargin46 \newcommand*\led@warn@BadSidenotemargin{%
247 \reledmac@warning{Bad \string\sidenotemargin\space argument}}
248 %

```

```

\led@warn@NoIndexFile49 \newcommand*\led@warn@NoIndexFile[1]{%
250 \reledmac@warning{Undefined index file #1}}
251 %

```

```

\led@warn@SeriesStillExist52 \newcommand*\led@warn@SeriesStillExist[1]{%
253 \reledmac@warning{Series #1 is still existing !}}%
254 }%
255 %

```

```

\led@err@BadAction56 \newcommand*\led@err@StanzaIndentNotDefined{%
257 \reledmac@error{You have not defined the indentation for the line \number
\stanza@count}\@ehc}}%
258 %

```

```

\led@err@ManySidenotes59 \newcommand*\led@err@ManySidenotes{%
\led@err@ManyLeftnotes60 \ifledRcol%
\led@err@ManyRightnotes61 \reledmac@warning{\itemcount@\space sidenotes on line \the\line@numR\
space p. \the\page@numR}%
262 \else%
263 \reledmac@warning{\itemcount@\space sidenotes on line \the\line@num\
space p. \the\page@num}%
264 \fi%
265 }%
266 \newcommand*\led@err@ManyLeftnotes{%
267 \ifledRcol%
268 \reledmac@warning{\itemcount@\space leftnotes on line \the\line@numR\
space p. \the\page@numR}%

```

```

269 \else%
270 \reledmac@warning{\itemcount@\space leftnotes on line \the\line@num\
space p. \the\page@num}%
271 \fi%
272 }%
273 \newcommand{\led@err@ManyRightnotes}{%
274 \ifledRcol%
275 \reledmac@warning{\itemcount@\space rightnotes on line \the\line@numR\
space p. \the\page@numR}%
276 \else%
277 \reledmac@warning{\itemcount@\space rightnotes on line \the\line@num\
space p. \the\page@num}%
278 \fi%
279 }%
280 %

```

```

\led@err@TooManyColumns81 \newcommand*\led@err@TooManyColumns{%
\led@err@UnequalColumns82 \reledmac@error{Too many columns}{\@ehc}}
\led@err@LowStartColumn83 \newcommand*\led@err@UnequalColumns{%
\led@err@HighEndColumn84 \reledmac@error{Number of columns is not equal to the number
\led@err@ReverseColumns85 in the previous row (or \protect\\ \space forgotten?)}{\@ehc}}
286 \newcommand*\led@err@LowStartColumn{%
287 \reledmac@error{Start column is too low}{\@ehc}}
288 \newcommand*\led@err@HighEndColumn{%
289 \reledmac@error{End column is too high}{\@ehc}}
290 \newcommand*\led@err@ReverseColumns{%
291 \reledmac@error{Start column is greater than end column}{\@ehc}}
292 %

```

```

\led@err@toendnotes@outsidenumbering93 \newcommand{\led@err@toendnotes@outsidenumbering}{%
294 \reledmac@error{\string\toendnotes\space and related commands must be
called inside a numbered text (\string\beginnumbering\space ... \space\
string\endnumbering)}{\@ehc}%
295 }%
296 %

```

```

\led@err@EdtextWithoutFootnote97 \newcommand{\led@err@EdtextWithoutFootnote}{%
298 \reledmac@error{edtext without Xfootnote. Check syntax}{\@ehc}%
299 }%
300 %

```

```

\led@err@FootnoteNotInSecondArgEdtext101 \newcommand{\led@err@FootnoteNotInSecondArgEdtext}[1]{%
302 \reledmac@error{#1footnote outside of the second argument of an edtext.
Check syntax}{\@ehc}%
303 }%
304 %

```

```

error@PackageAfterEledmac 305 \newcommand{\led@error@PackageAfterEledmac}[1]{%
306   \reledmac@error{#1 must be loaded before reledmac}{\@ehc}%
307 }%
308 %

error@fail@patch@@makecol 309 \newcommand{\led@error@fail@patch@@makecol}{%
310   \reledmac@error{Fail to patch \string\@makecol\space command}{\@ehc}%
311 }%
312 %

ror@fail@patch@@reinserts 313 \newcommand{\led@error@fail@patch@@reinserts}{%
314   \reledmac@error{Fail to patch \string\@reinserts\space command}{\@ehc}%
315 }%
316 %

r@fail@patch@@docclearpage 317 \newcommand{\led@error@fail@patch@@docclearpage}{%
318   \reledmac@error{Fail to patch \string\@docclearpage\space command}{\@ehc}%
319 }%
320 %

r@fail@patch@@iiiminipage 321 \newcommand{\led@error@fail@patch@@iiiminipage}{%
322   \reledmac@error{Fail to patch \string\@iiiminipage\space command}{\@ehc}%
323 }%
324 %

or@fail@patch@endminipage 325 \newcommand{\led@error@fail@patch@endminipage}{%
326   \reledmac@error{Failed to patch the \string\endminipage\space command}{\@ehc}%
327 }%
328 %

or@fail@patch@endminipage 329 \newcommand{\led@error@fail@patch@makeindex}{%
330   \reledmac@error{Failed to patch the \string\makeindex\space command}{\@ehc}%
331 }%
332 %

n@edinde@outsidenumbering 333 \newcommand{\led@warn@edinde@outsidenumbering}{%
334   \reledmac@warning{\string\edindex\space called outside of \string\beginnumbering\space ... \space \string\endnumbering. \MessageBreak Automatically switched to \string\index.}%
335 }%
336 %

```

```
\led@warning@hsizeX@deprecated37 \newcommand{\led@warning@hsizeX@deprecated}{%
338 \reledmac@warning{\string\hsizeX\space command deprecated, use \string\
widthX\space instead.}%
339 }%
340 %
```

```
\led@warning@Xhsize@deprecated41 \newcommand{\led@warning@Xhsize@deprecated}{%
342 \reledmac@warning{\string\Xhsize\space command deprecated, use \string\
Xwidth\space instead.}%
343 }%
344 %
```

```
\led@warning@msdatawithoutstop45 \newcommand{\led@warning@msdatawithoutstop}{%
346 \reledmac@warning{\string\msdata\space without corresponding \string\
stopmsdata}%
347 }%
348 %
```

```
\led@warning@preXnotes@deprecated49 \newcommand{\led@warning@preXnotes@deprecated}{%
350 \reledmac@warning@preXnotes@deprecated%
351 }%
352 %
```

## II.8 Gobbling

Here, we define some commands which gobble their arguments.

```
\@gobblethree53 \providecommand*\@gobblethree}[3]{}
\@gobblefour54 \providecommand*\@gobblefour}[4]{}
\@gobbleseven55 \providecommand*\@gobbleseven}[7]{}
356 %
```

## 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).

```

357 \newcommand{\l@wrap@ifnotemptybox}[2]{%
358   \setbox0=\hbox{#2}%
359   \ifdim\wd0=\z@ \else%
360     #1{#2}%
361   \fi%
362 }%
363 \newcommand{\l@wrapcs@ifnotemptybox}[2]{%
364   \l@wrap@ifnotemptybox{\csname #1\endcsname}{#2}%
365 }%
366 %

```

## II.10 Miscellaneous commands

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

```

367 \ifl@dfinal
368   \newcommand*{\showlemma}[1]{#1}
369 \else
370   \newcommand*{\showlemma}[1]{\underline{#1}}
371 \fi
372
373 %

```

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

Initialize it as `\empty`.

```

374 \let\linenumberlist=\empty
375
376 %

```

`\@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).

```

377 \newcount\@l@tempcnta \newcount\@l@tempcntb
378 %

```

## 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%`

```

382 \newif\ifl@dprintingcolumns%
383 \newif\ifpst@rtedL
384 \newcount\l@dnumstartsL
385 %

```

`\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`.

```

386 \newif\ifledRcol
387 \newif\ifledRcol@
388 %

```

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

```

389 \newif\ifnumberingR
390 %

```

`\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`.

```

391 \newif\ifafterendnumberingR%
392 %

```

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

```

393 \newif\ifXnote@%
394 %

```

## 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.

```

395 \providebool{indtl@innote}%
396 \providebool{indtl@notenumber}%
397 %

```

## III Sectioning commands

`\section@num` You use `\beginnumbering` and `\endnumbering` to begin and end a line-numbered section of the text; the pair of commands may be used as many times as you like within one document to start and end multiple, separately line-numbered sections.  $\TeX$  will maintain and display a 'section number' as a count named `\section@num` that counts how many `\beginnumbering` and `\resumenummering` commands have appeared; it need not be related to the logical divisions of your text.

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

```
398 \newcount\section@num
399 \section@num=0
400 \let\extensionchars=\empty
401 %
```

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

```
402 \newif\ifnumbering
403 %
```

`\beginnumbering` `\initnumbering@reg` `\beginnumbering` begins a section of numbered text. When it is executed we increment the section number, initialize our counters, send a message to your terminal, and call macros to start the lineation machinery and endnote files.

The initializations here are trickier than they look. `\line@list@stuff` will use all of the counters that are zeroed here when it assembles the line-list and other lists of information about the lineation. But it will do all of this locally and within a group, and when it is done the lists will remain but the counters will return to zero. Those same counters will then be used as we process the text of this section, but the assignments will be made globally. These initializations actually apply to both uses, though in all other respects there should be no direct interaction between the use of these counters and variables in the two processing steps. For parallel processing :

- zero `\l@dnumpstartsL` — the number of chunks to be processed.
- set `\ifpst@rtedL` to FALSE.

```
404 \newcommand*{\beginnumbering}{%
405   \ifnumbering
406     \led@err@NumberingStarted
407   \endnumbering
408   \fi
409   \global\numberingtrue
410   \global\afterendnumberingRfalse%
411   \global\advance\section@num \@ne
412   \initnumbering@reg
413   \message{Section \the\section@num }%

```

```

414 \line@list@stuff{\jobname.\extensionchars\the\section@num}%
415 \ifcontinuousnumberingwithcolumns%
416   \unless\ifafterendnumberingR%
417     \unless\ifl@dpairing%
418       \ledRcoltrue%
419       \global\advance\section@numR \@ne%
420       \message{Section \the\section@numR R (continuousnumbering)}%
421       \line@list@stuffR{\jobname.\extensionchars\the\section@numR R}%
422       \ledRcolfalse%
423     \fi%
424   \fi%
425 \fi%
426 \l@dend@stuff
427 \setcounter{pstart}{1}
428 \ifl@dpairing
429   \global\l@dnumpstartsL \z@
430   \global\pst@rtedLfalse
431 %

```

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.

```

432 \else
433   \begingroup
434   \global\@afterindenttrue%In order to reestablish normal feature if the \
435   \initnumbering@quote
436   \ifwidthliketwocolumns%
437     \setwidthliketwocolumns%
438     \csuse{setpositionliketwocolumns@\columns@position}%
439   \fi%
440 \fi
441 \gdef\eled@sections@{ }%
442 \if@noeled@sec\else%
443   \makeatletter%
444   \InputIfFileExists%
445     {\l@auxdir\jobname.eledsec\the\section@num}%
446     {}%
447     {\led@warn@NoFile{\l@auxdir\jobname.eledsec\the\section@num}}%
448   \makeatother%
449   \immediate\openout\eled@sectioning@out=\l@auxdir\jobname.eledsec\the\
450   section@num\relax%
451   \fi%
452 \fi
453 \newcommand*{\initnumbering@reg}{%
454   \global\pst@rtedLfalse

```



```

454 \global\l@dnumpstartsL \z@
455 \global\absline@num \z@
456 \gdef\normal@page@break{}
457 \gdef\l@prev@pb{}
458 \gdef\l@prev@nopb{}
459 \global\line@num \z@
460 \global\subline@num \z@
461 \global\@lock \z@
462 \global\sub@lock \z@
463 \global\sublines@false
464 \global\let\next@page@num=\relax
465 \global\let\this@section@next@page@num=\relax%
466 \global\let\sub@change=\relax
467 \global\last@page@num=-10000%
468 \ifdefined\line@numR%
469   \line@numR=\z@%
470   \last@page@numR=\z@%
471 \fi%
472 \resetprevline@
473 \resetprevpage@num
474 \global\stopmsdata@inserted@true%
475 \global\let\@msdata@list\relax%
476 \global\csundef{\@msdata@\add@msd@c @data}%
477 }
478
479 %

```

`\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.

```

480 \def\endnumbering{%
481   \ifnumbering
482     \global\numberingfalse
483     \normal@pars
484     \ifnum\l@dnumpstartsL=0%
485       \led@err@NumberingWithoutPstart%
486     \fi%
487 %

```

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`.

```

488   \global\page@num=\this@section@page@num%
489   \global\last@page@num=\this@section@last@page@num%
490   \global\let\next@page@num\this@section@next@page@num%
491   \ifl@dpairing

```

```

492 \global\pst@rtedLfalse
493 \else
494 \ifx\insertlines@list\empty\else
495 \global\noteschanged@true
496 \fi
497 \ifx\line@list\empty\else
498 \global\noteschanged@true
499 \fi
500 \fi
501 \ifnoteschanged@
502 \led@mess@NotesChanged
503 \fi
504 \else
505 \led@err@NumberingNotStarted
506 \fi
507 \autoparfalse
508 \if@noeled@sec\else%
509 \immediate\closeout\eled@sectioning@out%
510 \fi%
511 \ifl@dpairing\else
512 \global\l@dnumpststartsl=\z@%
513 \endgroup
514 \fi
515 }
516 %

```

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

```

517 \newcount\pausenumbering@page@num%
518 \newcommand{\pausenumbering}{%
519 \ifx\this@section@next@page@num\relax%
520 \global\pausenumbering@page@num=0%
521 \else%
522 \global\pausenumbering@page@num=\this@section@next@page@num%
523 \fi%
524 \ifautopar\global\autopar@pausetrue\fi%
525 \endnumbering\global\numberingtrue}
526 %

```

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.

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

```

527 \newif\ifresumenumbering@start%
528 \newcommand*{\resumenumbering}{%
529   \ifnumbering
530     \ifautopar@pause\autopar\fi
531     \global\pst@rtedLtrue
532     \global\advance\section@num \@ne
533     \global\resumenumbering@starttrue%
534     \led@mess@SectionContinued{\the\section@num}%
535     \set@continuousnumberingforL%
536     \line@list@stuff{\jobname.\extensionchars\the\section@num}%
537     \ifcontinuousnumberingwithcolumns%
538       \unless\ifafterendnumberingR%
539         \unless\ifl@dpairing%
540           \ledRcoltrue%
541           \global\advance\section@numR \@ne%
542           \message{Section \the\section@numR R (continuousnumbering)}%
543           \line@list@stuffR{\jobname.\extensionchars\the\section@numR R}%
544           \ledRcolfalse%
545         \fi%
546       \fi%
547     \fi%
548     \l@dend@stuff
549     \ifl@dpairing\else%
550       \begingroup%
551       \initnumbering@quote%
552       \ifwidthliketwocolumns%
553         \setwidthliketwocolumns%
554         \csuse{setpositionliketwocolumns@}\columns@position}%
555       \fi%
556     \fi%
557   \else
558     \led@err@NumberingShouldHaveStarted
559     \endnumbering
560     \beginnumbering
561   \fi}
562
563
564 %

```

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

```

565 \newcommand{\set@continuousnumberingforL}{%
566   \ifcontinuousnumberingwithcolumns%
567     \ifl@dpairing%
568       \unless\ifl@dpadding%
569         \global\c@pstartL=\c@pstart%
570       \fi%
571     \fi%
572   \fi%

```

```

573 }%
574 %

```

## 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 L<sup>A</sup>T<sub>E</sub>X3 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.

```

575 \newcommand*{\list@create}[1]{%
576   \global\let#1=\empty%
577 }%
578 %

```

**\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 .

```

579 \newcommand*{\list@clear}[1]{%
580   \global\let#1=\empty%
581 }
582 %

```

**\xright@appenditem** `\xright@appenditem` expands an item and appends it to the right end of a list macro. 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`.

```

583 \newtoks\led@toksa \newtoks\led@toksb
584 \global\let@toksa={\}
585 \long\def\xright@appenditem#1\to#2{%
586   \global\led@toksb=\expandafter{#2}%
587   \xdef#2{\the\led@toksb\the\led@toksa\expandafter{#1}}%
588   \global\led@toksb={}}
589 %

```

`\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`.

```

590 \long\def\xleft@appenditem#1\to#2{%
591   \global\led@toksb=\expandafter{#2}%
592   \xdef#2{\the\led@toksa\expandafter{#1}\the\led@toksb}%
593   \global\led@toksb={}}
594 %

```

`\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.

```

595 \def\gl@p#1\to#2{\expandafter\gl@poff#1\gl@poff#1#2}
596 \long\def\gl@poff\#1#2\gl@poff#3#4{\gdef#4{#1}\gdef#3{#2}}
597
598 %

```

## V Line counting

### V.1 Choosing the system of lineation

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

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

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

`\bypstart@true`

`\ifbypage@`

`\bypage@true`

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

```

599 \newif\ifbypage@
600 \newif\ifbypstart@
601 %

```

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@R62 \newif\ifbypage@R
\ifbypstart@R63 \newif\ifbypstart@R
604 %

```

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

```

605 \newcommand*{\lineation}[1]{\{
606 %

```

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

```

607 \ifnumbering
608 \led@err@LineationInNumbered
609 \else
610 %

```

If the argument is page.

```

611 \def\@tempa{#1}\def\@tempb{page}%
612 \ifx\@tempa\@tempb
613 \global\bypage@true
614 \global\bypstart@false
615 \unless\ifnocritical@%
616 \Xpstart[] [false]%
617 \fi%
618 %

```

If the argument is pstart.

```

619 \else
620 \def\@tempb{pstart}%
621 \ifx\@tempa\@tempb
622 \global\bypage@false
623 \global\bypstart@true
624 \unless\ifnocritical@%
625 \Xpstart%
626 \fi%
627 %

```

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

```

628 \else
629 \def\@tempb{section}
630 \ifx\@tempa\@tempb
631 \global\bypage@false
632 \global\bypstart@false
633 \unless\ifnocritical@%
634 \Xpstart[] [false]%
635 \fi%
636 %

```

In other case, it is an error.

```

637 \else
638 \led@warn@BadLineation
639 \fi
640 \fi
641 \fi
642 \fi}}
643 %

```

## V.2 Line number margin

`\linenummargin` `\linenummargin{<word>}` specify which margin line numbers are in; it takes one argument, a string, which value can be left ; right; inner or outer.  
`\line@margin`  
`\l@dgetline@margin` The selection is recorded in the count `\line@margin`: 0 for left, 1 for right, 2 for outer, and 3 for inner.

```

644 \newcount\line@margin%
645 \newcount\line@margin@columns%Only for parallel typesetting
646 \line@margin@columns=\m@ne%
647
648 \newcommand*{\linenummargin}[1]{%
649   \l@dgetline@margin{#1}%
650   \ifnum\@l@tempcntb>\m@ne
651     \ifledRcol
652       \global\line@marginR=\@l@tempcntb
653       \led@warn@setting@in@rightside{\linenummargin}%
654     \else
655       \global\line@margin=\@l@tempcntb
656     \fi
657   \fi}}
658
659 \newcommand*{\l@dgetline@margin}[1]{%
660   \def\@tempa{#1}\def\@tempb{left}%
661   \ifx\@tempa\@tempb
662     \@l@tempcntb \z@
663   \else
664     \def\@tempb{right}%
665     \ifx\@tempa\@tempb
666       \@l@tempcntb \@ne
667     \else
668       \def\@tempb{outer}%
669       \ifx\@tempa\@tempb
670         \@l@tempcntb \tw@
671       \else
672         \def\@tempb{inner}%
673         \ifx\@tempa\@tempb
674           \@l@tempcntb \thr@@
675         \else
676           \led@warn@BadLinenummargin
677           \@l@tempcntb \m@ne
678         \fi
679       \fi
680     \fi
681   \fi}
682
683 %

```

### 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.

```
684 \newcounter{firstlinenum}
685   \setcounter{firstlinenum}{5}
686 \newcounter{linenumincrement}
687   \setcounter{linenumincrement}{5}
688 %
```

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

```
689 \newcounter{firstsublinenum}
690   \setcounter{firstsublinenum}{5}
691 \newcounter{sublinenumincrement}
692   \setcounter{sublinenumincrement}{5}
693
694 %
```

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

```
\linenumincrement
\firstsublinenum
\sublinenumincrement
695 \newcommand*{\firstlinenum}[1]{%
696   \ifledRcol%
697     \setcounter{firstlinenumR}{#1}%
698     \led@warn@setting@in@rightside{\firstlinenum}%
699   \else%
700     \setcounter{firstlinenum}{#1}%
701   \fi%
702 }
703 \newcommand*{\linenumincrement}[1]{%
704   \ifledRcol%
705     \setcounter{linenumincrementR}{#1}%
706     \led@warn@setting@in@rightside{\linenumincrement}%
707   \else%
708     \setcounter{linenumincrement}{#1}%
709   \fi%
710 }
711 \newcommand*{\firstsublinenum}[1]{%
712   \ifledRcol%
713     \setcounter{firstsublinenumR}{#1}%
714     \led@warn@setting@in@rightside{\firstsublinenum}%
715   \else%
716     \setcounter{firstsublinenum}{#1}%
717   \fi%
718 }
```



```

719 }
720 \newcommand*{\sublinenumincrement}[1]{%
721   \ifledRcol%
722     \setcounter{sublinenumincrementR}{#1}%
723     \led@warn@setting@in@rightside{\sublinenumincrement}%
724   \else%
725     \setcounter{sublinenumincrement}{#1}%
726   \fi%
727 }
728
729 %

```

## 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.

```

730 \newcount\lock@disp
731 \newcommand{\lockdisp}[1]{%
732   \l@dgetlock@disp{#1}%
733   \ifnum\l@dtempcntb>\m@ne
734     \global\lock@disp=\l@dtempcntb
735   \else
736     \led@warn@BadLockdisp
737   \fi}}
738 \newcommand*{\l@dgetlock@disp}[1]{
739   \def\@tempa{#1}\def\@tempb{first}%
740   \ifx\@tempa\@tempb
741     \l@dtempcntb \z@
742   \else
743     \def\@tempb{last}%
744     \ifx\@tempa\@tempb
745       \l@dtempcntb \@ne
746     \else
747       \def\@tempb{all}%
748       \ifx\@tempa\@tempb
749         \l@dtempcntb \tw@
750       \else
751         \l@dtempcntb \m@ne
752       \fi
753     \fi
754   \fi}
755
756 %

```

`\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.

```

757 \newcount\sublock@disp
758 \newcommand{\sublockdisp}[1]{%
759   \l@getlock@disp{#1}%
760   \ifnum\l@dttempcntb>\m@ne
761     \global\sublock@disp=\l@dttempcntb
762   \else
763     \led@warn@BadSublockdisp
764   \fi}}
765
766 %

```

## 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.

```

767 \newcommand*{\linenumberstyle}[1]{%
768   \def\linenumrep##1{\@nameuse{@#1}{##1}}}
769 \newcommand*{\sublinenumberstyle}[1]{%
770   \def\sublinenumrep##1{\@nameuse{@#1}{##1}}}
771 %

```

Initialise the number styles to arabic.

```

772 \linenumberstyle{arabic}
773 \let\linenumr@p\linenumrep
774 \sublinenumberstyle{arabic}
775 \let\sublinenumr@p\sublinenumrep
776
777 %

```

## V.6 Line number printing

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

Whatever these macros output gets printed in a box that will be put into the appropriate margin without any space between it and the line of text. You will generally want a kern between a line number and the text, and `\linenumsep` is provided as a

standard way of storing its size. Line numbers are usually printed in a smaller font, and `\numlabfont` is provided as a standard name for that font. When called, these macros will be executed within a group, so font changes and the like will remain local.

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

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

```

778 \newlength{\linenumsep}
779 \setlength{\linenumsep}{1pc}
780 \newcommand*{\numlabfont}{\normalfont\scriptsize}
781 \newcommand*{\ledlinenum}{%
782   \bgroup%
783   \ifluatex%
784     \texdir TLT%
785   \fi%
786   \numlabfont%
787   \ifdefstring{\Xlinenumannotationposition@side}{before}{%
788     \l@wrap@ifnotemptybox{\Xwraplinenumannotation@side}{%
789       \csuse{annot@the\absline@num @the\section@num}%
790     }%
791   }{%
792     \linenumrep{\line@num}%
793     \ifsublines@
794       \ifnum\subline@num>0\relax
795         \unskip%
796         \Xsublinesep@side%
797         \sublinenumrep{\subline@num}%
798       \fi
799     \fi%
800     \ifdefstring{\Xlinenumannotationposition@side}{after}{%
801       \l@wrap@ifnotemptybox{\Xwraplinenumannotation@side}{%
802         \csuse{annot@the\absline@num @the\section@num}%
803       }%
804     }{%
805       \egroup%
806     }%
807   }%
808 \newcommand*{\leftlinenum}{%
809   \ledlinenum
810   \kern\linenumsep}
811 \newcommand*{\rightlinenum}{%
812   \kern\linenumsep
813   \ledlinenum}
814
815 %

```

## 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`.

```
816 \newcount\line@num
817 %
```

`\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.

```
818 \newcount\subline@num
819 %
```

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

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

`\sublines@false`

```
820 \newif\ifsublines@
821 %
```

`\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.

```

822 \newcount\absline@num
823 %

```

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.

```

824 \newcount\@lock
825 \newcount\sub@lock
826 %

```

## V.9 Line number associated to lemma

```

\line@list
\insertlines@list
\actionlines@list
\actions@list
\annot@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|OT1/cmr/m/n.
```

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

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

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

```
827 \list@create{\line@list}
828 \list@create{\insertlines@list}
829 \list@create{\actionlines@list}
830 \list@create{\actions@list}
831 \list@create{\annot@list}%
832 %
```

```
\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
833 \newcount\page@num
834 \newcount\endpage@num
835 \newcount\endline@num
836 \newcount\endsubline@num
837 %
```

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

```
838 \newcount\this@section@page@num%
839 %
```

```
\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 `TeX`, on this file, the information from the line-list used to place the notes will be wrong, and some notes will probably be misplaced. When this happens, we prefer to give a single error message for the whole section rather than messages at every point where we notice the problem, because we do not really know where in the section notes were added or removed, and the solution in any case is simply to run `TeX` 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.

```
840 \newif\ifnoteschanged@
841 %
```

`\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@42 \newcommand*\resetprevline@{%
843   \def\do##1{\global\csundef{prevline##1}}%
844   \dolistloop{\@series}%
845 }
846 %
```

`\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@47 \newcommand*\resetprevpage@num{%
848   \def\do##1{%
849     \ifcsdef{prevpage##1@num}{%
850       \global\csname prevpage##1@num\endcsname=\z@%
851       \global\csname prevpage##1@numR\endcsname=\z@%
852     }%
853     {}%
854     \ifcsdef{##1prevpage@num}{%
855       \global\csname ##1prevpage@num\endcsname=\z@%
856       \global\csname ##1prevpage@numR\endcsname=\z@%
857     }%
858     {}%
859   }%
860   \dolistloop{\@series}%
861 }
862 %
```

## 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.

```
863 \newread\@inputcheck
864 \newcommand*\read@linelist[1]{%
```



```

865 \ifledRcol%
866   \list@clearing@regR%
867 \else%
868   \list@clearing@reg%
869 \fi%
870 %

```

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).

```

871 \list@clear{\maxlinesinpar@list}
872 %

```

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.

```

873 \get@linelistfile{#1}%
874 \@stopmsd%Security if last \endms{} is forgotten
875 \unless\ifledRcol%Get the last line of the last page
876   \csnumgdef{@lastabsline@forpage@the\page@num}{\the\absline@num}%
877   \csnumgdef{@lastline@forpage@the\page@num}{\the\line@num}%
878 \else%
879   \csnumgdef{@lastabsline@forpageR@the\page@numR}{\the\absline@numR}%
880   \csnumgdef{@lastline@forpageR@the\page@numR}{\the\line@numR}%
881 \fi%
882 \endgroup
883 %

```

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.

```

884 \ifledRcol
885   \global\page@numR=\m@ne
886   \ifx\actionlines@listR\empty
887     \gdef\next@actionlineR{1000000}%
888   \else

```

```

889     \gl@p\actionlines@listR\to\next@actionlineR
890     \gl@p\actions@listR\to\next@actionR
891   \fi
892 \else
893   \global\page@num=\m@ne
894   \ifx\actionlines@list\empty
895     \gdef\next@actionline{1000000}%
896   \else
897     \gl@p\actionlines@list\to\next@actionline
898     \gl@p\actions@list\to\next@action
899   \fi
900 \fi
901 }
902 %

```

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

```

903 \newcommand*{\list@clearing@reg}{%
904   \list@clear{\line@list}%
905   \list@clear{\insertlines@list}%
906   \list@clear{\actionlines@list}%
907   \list@clear{\actions@list}%
908   \list@clear{\linesinpar@listL}%
909   \list@clear{\linesonpage@listL}%
910 }%
911 %

```

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

```

912 \newcommand*{\get@linelistfile}[1]{%
913   \InputIfFileExists{\l@auxdir#1}{%
914     \global\noteschanged@false
915     \begingroup
916       \catcode`\[=1 \catcode`\]=2
917       \makeatletter \catcode`\^^M=9}%
918     \led@warn@NoFile{\l@auxdir#1}%
919     \global\noteschanged@true
920     \begingroup}%
921 }
922
923 %

```

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  $\LaTeX$  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. 20 above).

## V.11 Commands within the line-list file

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

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

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

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

```

924 \newcommand{\line@list@version}[1]{%
925   \IfStrEq{#1}{\this@line@list@version}%
926   {}%
927   {\ifledRcol%
928     \led@warn@Obsolete{\jobname.\extensionchars\the\section@num}%
929     \else%
930     \led@warn@Obsolete{\jobname.\extensionchars\the\section@num}%
931     \fi%
932     \endinput%
933   }%
934 }%
935 %

```

`\@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`.

```

936 \newcommand*{\@nl}[2]{%
937   \@page{#1}%
938   \ifledRcol%
939     \@nl@regR%
940   \else%
941     \@nl@reg%
942   \fi%
943 }
944 \newcommand*{\@nl@reg}{%
945   \ifx\l@dchset@num\relax \else
946     \advance\absline@num \@ne
947     \csgdef{l@dchset@num@the\absline@num}{}%To remember this line have
948     been marked by a \setlinenum
949     \set@line@action
950     \let\l@dchset@num=\relax
951     \advance\absline@num \m@ne
952     \advance\line@num \m@ne
953   \fi
954 %

```

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

```

955 \reset@current@annot%
956 \advance\absline@num \@ne
957   \ifx\next@page@num\relax \else
958     \page@action
959     \let\next@page@num=\relax
960   \fi
961   \ifx\sub@change\relax \else
962     \ifnum\sub@change>\z@
963       \sublines@true
964     \else
965       \sublines@false
966     \fi
967     \sub@action
968     \let\sub@change=\relax
969   \fi
970 %

```

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

```

971   \ifcase\@lock
972     \or
973       \@lock \tw@
974     \or \or
975       \@lock \z@

```

```

976     \fi
977     \ifcase\sub@lock
978     \or
979     \sub@lock \tw@
980     \or \or
981     \sub@lock \z@
982 \fi
983 %

```

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

```

984 \ifsublines@
985 \ifnum\sub@lock<\tw@
986 \advance\subline@num \@ne
987 \fi
988 \else
989 \ifnum\@lock<\tw@
990 \advance\line@num \@ne \subline@num \z@
991 \fi
992 \fi}
993 %
994 %

```

`\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
995 \newcount\last@page@num
996 \last@page@num=-10000
997
998 \newcount\this@section@last@page@num%
999 \this@section@last@page@num=-10000%
1000
1001 \newcommand*{\@page}[1]{%
1002 \ifl@r@col
1003 \ifnum #1=\last@page@numR
1004 \else
1005 \csnumgdef{\lastabsline@forpageR\the\page@numR}{\the\absline@numR}%
1006 \csnumgdef{\lastline@forpageR\the\page@numR}{\the\line@numR}%
1007 \ifbypage@R
1008 \ifx\l@dchset@num\relax%Not resetting if preceded by a \setlinenum
1009 \ifboolexpr{%
1010 bool{resumenumberingR@start}%
1011 and test {\ifnumequal{\last@page@numR}{-10000}}%
1012 }%
1013 {}%
1014 {%
1015 \line@numR \z@%
1016 \subline@numR \z@%
1017 \global\csdef{reset@lineR\the\numexpr\absline@numR+1\relax @\
the\section@numR}{}%
1018 }%

```

```

1019     \fi%
1020     \fi
1021     \page@numR=#1%
1022     \this@section@page@numR=#1%
1023     \last@page@numR=#1%
1024     \this@section@last@page@numR=#1%
1025     \def\next@page@numR{#1}%
1026     \gdef\this@section@next@page@numR{#1}%
1027     \fi
1028     \else
1029     \ifnum #1=\last@page@num
1030     \else
1031     \csnumgdef{\lastabsline@forpage@the\page@num}{\the\absline@num}%
1032     \csnumgdef{\lastline@forpage@the\page@num}{\the\line@num}%
1033     \ifbypage@
1034     \ifx\ldchset@num\relax%Not resetting if preceded by a \setlinenum
1035     \line@num \z@%
1036     \subline@num \z@%
1037     \global\csdef{reset@line\the\numexpr\absline@num+1\relax @the\
section@num}{}%
1038     \fi%
1039     \fi
1040     \page@num=#1%
1041     \global\this@section@page@num=#1%
1042     \last@page@num=#1%
1043     \global\this@section@last@page@num=#1%
1044     \def\next@page@num{#1}%
1045     %

```

And we set a flag that tells \@n1 that a new page number is to be set, because other associated actions shouldn't occur until the next line-start occurs.

```

1046     \gdef\this@section@next@page@num{#1}%
1047     \listxadd{\normal@page@break}{\the\absline@num}
1048     \fi
1049     \fi}
1050     %

```

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

```

1051 \@lopR \newcommand*{\@pend}[1]{}
1052 \newcommand*{\@pendR}[1]{}
1053 \newcommand*{\@lopL}[1]{}
1054 \newcommand*{\@lopR}[1]{}
1055
1056 %

```

**\sub@on** The \sub@on and \sub@off macros turn sub-lineation on and off: but not directly, since  
**\sub@off**

such changes do not really take effect until the next line of text. Instead they set a flag that notifies \@nl of the necessary action.

```

1057 \newcommand*{\sub@on}{\ifsublines@
1058     \let\sub@change=\relax
1059 \else
1060     \def\sub@change{1}%
1061 \fi}
1062 \newcommand*{\sub@off}{\ifsublines@
1063     \def\sub@change{-1}%
1064 \else
1065     \let\sub@change=\relax
1066 \fi}
1067
1068 %

```

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

```

1069
1070 \newcommand*{\@adv}[1]{%
1071     \ifsublines@
1072     \ifledRcol
1073         \advance\subline@numR by #1\relax
1074         \ifnum\subline@numR<\z@
1075             \led@warn@BadAdvancelineSubline
1076             \subline@numR \z@
1077         \fi
1078     \else
1079         \advance\subline@num by #1\relax
1080         \ifnum\subline@num<\z@
1081             \led@warn@BadAdvancelineSubline
1082             \subline@num \z@
1083         \fi
1084     \fi
1085 \else
1086     \ifledRcol
1087         \advance\line@numR by #1\relax
1088         \ifnum\line@numR<\z@
1089             \led@warn@BadAdvancelineLine
1090             \line@numR \z@
1091         \fi
1092     \else
1093         \advance\line@num by #1\relax
1094         \ifnum\line@num<\z@
1095             \led@warn@BadAdvancelineLine
1096             \line@num \z@
1097         \fi
1098     \fi
1099 \fi

```

```

1100 \set@line@action}
1101
1102 %

```

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

```

1103
1104 \newcommand*{\@set}[1]{%
1105   \ifledRcol
1106     \ifsublines@
1107       \subline@numR=#1\relax
1108     \else
1109       \line@numR=#1\relax
1110     \fi
1111     \set@line@action
1112   \else
1113     \ifsublines@
1114       \subline@num=#1\relax
1115     \else
1116       \line@num=#1\relax
1117     \fi
1118     \set@line@action
1119   \fi}
1120
1121 %

```

**\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.

```

1122
1123 \newcommand*{\l@d@set}[1]{%
1124   \ifledRcol
1125     \line@numR=#1\relax
1126     \advance\line@numR \@ne
1127     \def\l@dchset@num{#1}
1128   \else
1129     \line@num=#1\relax
1130     \advance\line@num \@ne
1131     \def\l@dchset@num{#1}
1132   \fi}
1133 \let\l@dchset@num\relax
1134
1135 %

```

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



```

1136 \newcommand*{\page@action}{%
1137   \ifledRcol
1138     \xright@appenditem{\the\absline@numR}\to\actionlines@listR
1139     \xright@appenditem{\next@page@numR}\to\actions@listR
1140   \else
1141     \xright@appenditem{\the\absline@num}\to\actionlines@list
1142     \xright@appenditem{\next@page@num}\to\actions@list
1143   \fi}
1144 %
1145 %

```

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

```

1146 \newcommand*{\set@line@action}{%
1147   \ifledRcol
1148     \xright@appenditem{\the\absline@numR}\to\actionlines@listR
1149     \ifsublines@
1150       \@l@dttempcnta=-\subline@numR
1151     \else
1152       \@l@dttempcnta=-\line@numR
1153     \fi
1154     \advance\@l@dttempcnta by -5000\relax
1155     \xright@appenditem{\the\@l@dttempcnta}\to\actions@listR
1156   \else
1157     \xright@appenditem{\the\absline@num}\to\actionlines@list
1158     \ifsublines@
1159       \@l@dttempcnta=-\subline@num
1160     \else
1161       \@l@dttempcnta=-\line@num
1162     \fi
1163     \advance\@l@dttempcnta by -5000\relax
1164     \xright@appenditem{\the\@l@dttempcnta}\to\actions@list
1165   \fi}
1166 %
1167 %

```

**\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.

```

1168 \newcommand*{\sub@action}{%
1169   \ifledRcol
1170     \xright@appenditem{\the\absline@numR}\to\actionlines@listR
1171     \ifsublines@
1172       \xright@appenditem{-1001}\to\actions@listR
1173     \else
1174       \xright@appenditem{-1002}\to\actions@listR
1175     \fi
1176   \else
1177     \xright@appenditem{-1002}\to\actions@listR
1178   \fi}

```

```

1178 \xright@appenditem{\the\absline@num}\to\actionlines@list
1179 \ifsublines@
1180 \xright@appenditem{-1001}\to\actions@list
1181 \else
1182 \xright@appenditem{-1002}\to\actions@list
1183 \fi
1184 \fi}
1185 %

```

`\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.

```

1186 \newcommand*{\lock@on}{\futurelet\next\do@lockon}
1187
1188 \newcommand*{\do@lockon}{%
1189 \ifx\next\lock@off
1190 \global\let\lock@off=\skip@lockoff
1191 \else
1192 \ifledRcol
1193 \do@lockonR
1194 \else
1195 \do@lockonL
1196 \fi
1197 \fi}
1198
1199
1200 \newcommand*{\do@lockonL}{%
1201 \xright@appenditem{\the\absline@num}\to\actionlines@list
1202 \ifsublines@
1203 \xright@appenditem{-1005}\to\actions@list
1204 \ifnum\sub@lock=\z@
1205 \sub@lock \@ne
1206 \else
1207 \ifnum\sub@lock=\thr@@
1208 \sub@lock \@ne
1209 \fi
1210 \fi
1211 \else
1212 \xright@appenditem{-1003}\to\actions@list
1213 \ifnum\@lock=\z@
1214 \@lock \@ne
1215 \else
1216 \ifnum\@lock=\thr@@
1217 \@lock \@ne

```

```

1218     \fi
1219     \fi
1220 \fi}
1221
1222 %

```

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

```

\do@lockoff
\do@lockoffL
\skip@lockoff
1223 \newcommand*{\do@lockoffL}{%
1224 \xright@appenditem{\the\absline@num}\to\actionlines@list
1225 \ifsublines@
1226 \xright@appenditem{-1006}\to\actions@list
1227 \ifnum\sub@lock=\tw@
1228 \sub@lock \thr@@
1229 \else
1230 \sub@lock \z@
1231 \fi
1232 \else
1233 \xright@appenditem{-1004}\to\actions@list
1234 \ifnum\@lock=\tw@
1235 \@lock \thr@@
1236 \else
1237 \@lock \z@
1238 \fi
1239 \fi}
1240
1241 \newcommand*{\do@lockoff}{%
1242 \reset@current@annot%
1243 \ifledRcol
1244 \do@lockoffR
1245 \else
1246 \do@lockoffL
1247 \fi}
1248 \newcommand*{\skip@lockoff}{\global\let\lock@off=\do@lockoff}
1249 \global\let\lock@off=\do@lockoff
1250
1251 %

```

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

```

1252 \newcommand*{\n@num}{%
1253 \ifledRcol%
1254 \xright@appenditem{\the\absline@numR}\to\actionlines@listR
1255 \xright@appenditem{-1007}\to\actions@listR
1256 \else%
1257 \xright@appenditem{\the\absline@num}\to\actionlines@list%
1258 \xright@appenditem{-1007}\to\actions@list%
1259 \fi%
1260 }%

```

1261  
1262

%

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

```
1263 \newcommand*\n@num@stanza}{%
1264   \ifledRcol%
1265     \xright@appenditem{\the\absline@numR}\to\actionlines@listR%
1266     \xright@appenditem{-1008}\to\actions@listR%
1267   \else%
1268     \xright@appenditem{\the\absline@num}\to\actionlines@listR%%
1269     \xright@appenditem{-1008}\to\actions@listR%
1270   \fi%
1271 }
1272 %
```

**\ifl@dhidenumber** `\hidenumbering` hides number in margin. It uses action code 1009. `\hidenumberingonleftpage` and `\hidenumberingonrightpage` are variants, using action code only conditionally

**\hidenumberingonleftpage**  
**\hidenumberingonrightpage**

```
1273 \newif\ifl@dhidenumber
1274 \newcommand*\hidenumbering{%
1275   \ifledRcol%
1276     \write\linenum@outR{\string\hide@num}%
1277   \else%
1278     \write\linenum@out{\string\hide@num}%
1279   \fi%
1280 }%
1281 \newcommand*\hide@num{%
1282   \ifledRcol%
1283     \xright@appenditem{\the\absline@numR}\to\actionlines@listR%
1284     \xright@appenditem{-1009}\to\actions@listR%
1285   \else%
1286     \xright@appenditem{\the\absline@num}\to\actionlines@listR%%
1287     \xright@appenditem{-1009}\to\actions@listR%
1288   \fi%
1289 }
1290 \newcommand*\hidenumberingonleftpage{%
1291   \ifledRcol%
1292     \write\linenum@outR{\string\hide@num@left}%
1293   \else%
1294     \write\linenum@out{\string\hide@num@left}%
1295   \fi%
1296 }%
1297
1298 \newcommand*\hide@num@left{%
1299   \ifledRcol%
1300     \ifodd\page@numR\else%
1301       \xright@appenditem{\the\absline@numR}\to\actionlines@listR%
```

```

1302 \xright@appenditem{-1009}\to\actions@listR%
1303 \fi%
1304 \else%
1305 \ifodd\page@num\else%
1306 \xright@appenditem{\the\absline@num}\to\actionlines@list%%
1307 \xright@appenditem{-1009}\to\actions@list%
1308 \fi%
1309 \fi%
1310 }%
1311
1312 \newcommand*\hidenumberingonrightpage}{%
1313 \ifledRcol%
1314 \write\linenum@outR{\string\hide@num@right}%
1315 \else%
1316 \write\linenum@out{\string\hide@num@right}%
1317 \fi%
1318 }%
1319
1320 \newcommand*\hide@num@right}{%
1321 \ifledRcol%
1322 \ifodd\page@numR%
1323 \xright@appenditem{\the\absline@numR}\to\actionlines@listR%
1324 \xright@appenditem{-1009}\to\actions@listR%
1325 \fi%
1326 \else%
1327 \ifodd\page@num%
1328 \xright@appenditem{\the\absline@num}\to\actionlines@list%%
1329 \xright@appenditem{-1009}\to\actions@list%
1330 \fi%
1331 \fi%
1332 }%
1333
1334 %

```

**\@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.

```

1335 \newcount\insert@count
1336 %

```

- #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.

```

1337 \newcommand*\dummy@ref}[2]{#2}
1338 %

```

**\@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.

```

1339 \newcommand*\@ref}[2]{%
1340   \ifledRcol%
1341     \@ref@regR{#1}{#2}%
1342   \else%
1343     \@ref@reg{#1}{#2}%
1344   \fi%
1345 }%
1346 \newcommand*\@ref@reg}[2]{%
1347   \global\insert@count=#1\relax
1348   \global\advance\@edtext@level by 1%
1349   \loop\ifnum\insert@count>\z@
1350     \xright@appenditem{\the\absline@num}\to\insertlines@list
1351     \global\advance\insert@count \m@ne
1352   \repeat
1353 %

```

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.

```

1354 \begingroup
1355   \let\@ref=\dummy@ref
1356   \let\@lopL\@gobble
1357   \let\page@action=\relax
1358   \let\sub@action=\relax
1359   \let\set@line@action=\relax
1360   \let\@lab=\relax
1361   \let\@lemma=\relax%
1362   \let\@sw\@gobblethree%
1363   \let\store@annot@to@absline\@gobble%
1364   #2
1365   \global\endpage@num=\page@num
1366   \global\endline@num=\line@num
1367   \global\endsubline@num=\subline@num
1368   \global\let\endcurrent@annot=\current@annot%
1369 \endgroup
1370 %

```

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

```

1371 \xright@appenditem%
1372   {\the\page@num|\the\line@num|}%
1373   \ifsublines@ \the\subline@num \else 0\fil%

```

```

1374 \the\endpage@num|\the\endline@num|%
1375 \ifsublines@ \the\endsubline@num \else 0\fi}\to\line@list
1376 \xright@appenditem%
1377 {\current@annot|\endcurrent@annot}\to\annot@list%
1378 %

```

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

```

1379 \@ref@reg@parse{#2}%
1380 %

```

Decrease edtext level counter.

```

1381 \global\advance\@edtext@level by -1%
1382 }
1383 %

```

**`\@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.

```

1384 \newcommand{\@ref@reg@parse}[1]{%
1385   \expandafter\list@create\expandafter{\csname sw@list@edtext@tmp@\the\
1386   \@edtext@level\endcsname}%
1387   \providebool{lemmacommand@\the\@edtext@level}%
1388   \boolfalse{lemmacommand@\the\@edtext@level}%
1389 }

```

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

```

1389 #1%
1390 %

```

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.

```

1391 \ifnum\@edtext@level>0%
1392   \def\create@this@edtext@level{\expandafter\list@create\expandafter{\
1393   \csname sw@list@edtext@\the\@edtext@level\endcsname}}%
1394   \ifcsundef{sw@list@edtext@\the\@edtext@level}{\create@this@edtext@level}%
1395   \letcs{\@tmp}{sw@list@edtext@\the\@edtext@level}%
1396   \letcs{\@tmpp}{sw@list@edtext@tmp@\the\@edtext@level}%
1397   \xright@appenditem{\expandonce\@tmpp}\to\@tmp%
1398   \global\cslet{sw@list@edtext@\the\@edtext@level}{\@tmp}%
1399   \fi%
1400 }
1401 %
1402 %

```

`\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.

```

1403 \newcommand{\@ref@later}[1]{%
1404   \global\advance\@edtext@level by \@ne%
1405   \ifledRcol%
1406     \@ref@reg@parseR{#1}%
1407   \else%
1408     \@ref@reg@parse{#1}%
1409   \fi%
1410   \global\advance\@edtext@level by -\@ne%
1411 }%
1412 %

```

## 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`.

```

1413 \newwrite\linenum@out
1414 %

```

`\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.

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`.

```

1415 \newif\iffirst@linenum@out@
1416 \iffirst@linenum@out@true
1417 %

```

`\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.

```

1418 \newcommand{\this@line@list@version}{7}%
1419 %

```



`\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`

```
1420 \let\next@line@list@stuff\relax%
1421 \newcommand*{\line@list@stuff}[1]{%
1422 %
```

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

```
1423 \global\newtoggle{notfirstrun@#1}%
1424 \IfFileExists{\l@auxdir#1}%
1425 {\global\toggletrue{notfirstrun@#1}}%
1426 {\global\togglefalse{notfirstrun@#1}}%
1427 %
```

A internal hook (not used yet).

```
1428 \next@line@list@stuff%
1429 \global\let\next@line@list@stuff\relax%
1430 %
```

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

```
1431 \read@linelist{#1}%
1432 %
```

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  $n$ .

```
1433 \iffirst@linenum@out@
1434 \global\first@linenum@out@false%
1435 \immediate\openout\linenum@out=\l@auxdir#1\relax%
1436 \immediate\write\linenum@out{\string\line@list@version{\
this@line@list@version}}%
1437 \ifl@dpaging%
1438 \immediate\write\linenum@out{\string\@par@sync@option{\
@par@this@sync@option}}%
1439 \fi%
1440 \else
1441 %
```

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.

```
1442 \if@minipage%
1443 \leavevmode%
1444 \fi%
1445 \closeout\linenum@out%
1446 \openout\linenum@out=\l@auxdir#1\relax%
```

```

1447 \write\linenum@out{\string\line@list@version{\this@line@list@version}}
1448 %
1449 \ifl@dpaging%
1450 \write\linenum@out{\string\@par@sync@option{\@par@this@sync@option}}
1451 %
1452 \fi%
1453 \fi%
1454 }%

```

`\new@line` The `\new@line` macro sends the `\@nl` command to the line-list file, to mark the start of a new text line, and its page number. 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 `\@nl`. If we use the `\sameparallelpagenumber` option of `reledpar`, we must write not the real page number (i.e. page counter, defined in standard  $\TeX$ ) 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.

```

1455 \newcommand*{\new@line}{%
1456 \ifnumberline%
1457 \IfStrEq{\led@pb@setting}{after}%
1458 {\xifinlist{\the\absline@num}{\l@prev@nopb}%
1459 {\xifinlist{\the\absline@num}{\normal@page@break}%
1460 {\numgdef{\@next@page}{\c@par@page+\@ne}%
1461 \write\linenum@out{\string\@nl[\@next@page][\@next@page]}}%
1462 }%
1463 {\write\linenum@out{\string\@nl[\the\c@par@page][\thepage]}}%
1464 }%
1465 {\write\linenum@out{\string\@nl[\the\c@par@page][\thepage]}}%
1466 {}%
1467 \IfStrEq{\led@pb@setting}{before}%
1468 {\numdef{\next@absline}{\the\absline@num+\@ne}%
1469 \xifinlist{\next@absline}{\l@prev@nopb}%
1470 {\xifinlist{\the\absline@num}{\normal@page@break}%
1471 {\numgdef{\nc@page}{\c@par@page+\@ne}%
1472 \write\linenum@out{\string\@nl[\nc@page][\nc@page]}}%
1473 }%
1474 {\write\linenum@out{\string\@nl[\the\c@par@page][\thepage]}}%
1475 }%

```

```

1476     {\write\linenum@out{\string\@nl[\the\c@par@page][\thepage]}}%
1477   }%
1478 }%
1479 \IfStrEqCase{\led@pb@setting}%
1480 {%
1481   {before}{\relax}%
1482   {after}{\relax}%
1483 }[%
1484   \write\linenum@out{\string\@nl[\the\c@par@page][\thepage]}}%
1485 ]%
1486 \fi%
1487 }
1488
1489 \newcommand{\l@the\c@page}{%
1490   \ifboolexpr{%
1491     bool{sameparallelpagenumbers}%
1492     or bool{prevpgnotnumbered}%
1493   }%
1494   {\the\c@par@page}%
1495   {\the\c@page}%
1496 }%
1497 %

```

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

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

**\flag@end**

```

1498 \newif\if@noneed@Footnote%
1499
1500 \newcommand*{\flag@start}{%
1501   \ifledRcol%
1502     \edef\next{\write\linenum@outR{%
1503       \string\@ref[\the\insert@countR][\thepage]}}%
1504     \next%
1505     \ifnum\insert@countR<1%
1506       \if@noneed@Footnote\else%
1507         \led@err@EdtextWithoutFootnote%
1508       \fi%
1509     \fi%
1510   \else%
1511     \edef\next{\write\linenum@out{%
1512       \string\@ref[\the\insert@count][\thepage]}}%
1513     \next%
1514     \ifnum\insert@count<1%
1515       \if@noneed@Footnote\else%
1516         \led@err@EdtextWithoutFootnote%
1517       \fi%

```

```

1518 \fi%
1519 \fi}%
1520
1521 \newcommand*{\flag@end}{%
1522 \ifledRcol%
1523 \write\linenum@outR{}}%
1524 \else%
1525 \write\linenum@out{}}%
1526 \fi}%
1527
1528
1529 %

```

`\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>

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

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

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

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

```

1530 \newcommand{\flag@start@RTL}{%
1531 \edlabel{edtext:start:\csuse{thisedtext@the\@edtext@level}}%
1532 \IfStrEq{\xabslineref{edtext:start:\csuse{thisedtext@the\@edtext@level}}}{%
1533 {000}%
1534 {}%
1535 {%
1536 \ifnumequal%
1537 {\xabslineref{edtext:start:\csuse{thisedtext@the\@edtext@level}}}{%
1538 {\xabslineref{edtext:end:\csuse{thisedtext@the\@edtext@level}}}%
1539 {\flag@end}%
1540 {\flag@start}%

```

<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.

```

1541 }%
1542 }%
1543
1544 \newcommand{\flag@end@RTL}{%
1545   \edlabel{edtext:end:\csuse{thisedtext@the\@edtext@level}}}%
1546   \IfStrEq{\xabslineref{edtext:start:\csuse{thisedtext@the\@edtext@level
1547     {000}}}%
1548     }{%
1549     {%
1550     \ifnumequal%
1551       {\xabslineref{edtext:start:\csuse{thisedtext@the\@edtext@level}}}%
1552       {\xabslineref{edtext:end:\csuse{thisedtext@the\@edtext@level}}}%
1553       {\flag@start}%
1554       {\flag@end}%
1555     }%
1556   }%
1557   %

```

**\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**

```

1558 \newcommand*{\flag@start@later}{%
1559   \ifledRcol%
1560     \write\linenum@outR{\string\@ref@later []}%
1561   \else%
1562     \write\linenum@out{\string\@ref@later []}%
1563   \fi%
1564 }%
1565 \newcommand{\flag@end@later}{%
1566   \ifledRcol%
1567     \write\linenum@outR{[]}%
1568   \else%
1569     \write\linenum@out{[]}%
1570   \fi%
1571 }
1572 %

```

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

We tinker with \lastskip because a command of either sort really needs to be attached to the last word preceding the change, not the first word that follows the change. This is because sub-lineation will often turn on and off in mid-line—stage directions, for example, often are mixed with dialogue in that way—and when a line is mixed we want

to label it using the system that was in effect at its start. But when sub-lineation begins at the very start of a line we have a problem, if we don't put in this code.

```

1573
1574
1575 \newcommand*{\startsub}{\dimen0\lastskip
1576   \ifdim\dimen0>0pt \unskip \fi
1577   \ifledRcol \write\linenum@outR{\string\sub@on}%
1578   \else      \write\linenum@out{\string\sub@on}%
1579   \fi
1580   \ifdim\dimen0>0pt \hskip\dimen0 \fi}
1581 \def\endsub{\dimen0\lastskip
1582   \ifdim\dimen0>0pt \unskip \fi
1583   \ifledRcol \write\linenum@outR{\string\sub@off}%
1584   \else      \write\linenum@out{\string\sub@off}%
1585   \fi
1586   \ifdim\dimen0>0pt \hskip\dimen0 \fi}
1587
1588 %

```

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

```

1589 \newcommand*{\advanceline}[1]{\leavevmode%
1590   \ifledRcol \write\linenum@outR{\string\adv[#1]}%
1591   \else      \write\linenum@out{\string\adv[#1]}%
1592   \fi%
1593 }
1594 %

```

**\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.

```

1595
1596 \newcommand*{\setline}[1]{%
1597   \leavevmode%
1598   \ifnum#1<\z@
1599     \led@warn@BadSetline%
1600   \else%
1601     \ifledRcol%
1602       \write\linenum@outR{\string\@set[#1]}%
1603     \else%
1604       \write\linenum@out{\string\@set[#1]}%
1605     \fi%
1606   \fi%
1607 }%
1608
1609 %

```

**\setlinenum** You can use `\setlinenum{<num>}` before a `\pstart` to set the visible line-number to a specified positive value. It writes a `\led@set` command to the line-list file.

```

1610
1611 \newcommand*{\setlinenum}[1]{%
1612   \ifnum#1<\z@
1613     \led@warn@BadSetlinenum
1614   \else
1615     \ifledRcol \write\linenum@outR{\string\l@d@set[#1]}
1616     \else      \write\linenum@out{\string\l@d@set[#1]} \fi
1617   \fi}
1618
1619 %

```

**\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.

```

1620
1621 \newcommand*{\startlock}{%
1622   \ifledRcol \write\linenum@outR{\string\lock@on}%
1623   \else      \write\linenum@out{\string\lock@on}%
1624   \fi}
1625 \def\endlock{%
1626   \ifledRcol \write\linenum@outR{\string\lock@off}%
1627   \else      \write\linenum@out{\string\lock@off}%
1628   \fi}
1629 %

```

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

```

\ifl@dskipversenumber
\l@dskipnumbertrue
\l@dskipnumberfalse
\skipnumbering
1630 \newif\ifl@dskipnumber
1631 \newif\ifl@dskipversenumber%
1632 \newcommand*{\skipnumbering}{%
1633   \leavevmode%
1634   \ifledRcol%
1635     \ifinstanza%
1636       \write\linenum@outR{\string\n@num@stanza}%
1637     \else%
1638       \write\linenum@outR{\string\n@num}%
1639     \fi%
1640     \advanceline{-1}%
1641   \else%
1642     \ifinstanza%
1643       \write\linenum@out{\string\n@num@stanza}%
1644     \else%
1645       \write\linenum@out{\string\n@num}%
1646     \fi%
1647     \advanceline{-1}%
1648   \fi%
1649 }%
1650

```

1651 %

## 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. 160). 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.

```
1652 \list@create{\end@lemmas}
1653 %
```

`\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.

```
1654 \newcommand{\dummy@edtext}[2]{#1}
1655 %
```

`\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`.

```
1656 \newcommand{\dummy@edtext@showlemma}[2]{\showlemma{#1}}%
1657 %
```

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

The `\new@series` command also adds `\let\footnote<X>\@gobble` to the end of the `\no@expands` macro for the series `<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 $\TeX$`  or `X $\LaTeX$` .)

```

1658 \newcommand*{\no@expands}{%
1659   \let\select@lemmafont=0%
1660   \let\startsub=\relax   \let\endsub=\relax
1661   \let\startlock=\relax  \let\endlock=\relax
1662   \let\edlabel=\@gobble
1663   \let\setline=\@gobble  \let\advanceline=\@gobble
1664   \let\sameword\sameword@inedtext%
1665   \let\edtext=\dummy@edtext

```

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

```

1666 \let\edindex\dummy@edindex%
1667 \l@dtabnoexpands
1668 \let\linenumannotation=\@gobble%
1669 \morenoexpands}
1670 \let\morenoexpands=\relax
1671
1672 %

```

**\@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.

```

1673 \newcommand{\@tag}{}
1674 %

```

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

```

1675 \newcount\@edtext@level%
1676 \@edtext@level=0%
1677 %

```

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

```

1678 \newif\if@edtext@secondarg@%
1679 %

```

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

```

1680 \newcounter{edtext}
1681 \renewcommand{\theedtext}{edtxt@arabic{edtext}}%
1682 %

```

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

```

1683 \newcommand{\edtext}[2]{\leavevmode%
1684 %

```

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

```

1685 \ifnumberedpar@%
1686 %

```

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

```

1687 \@check@edtext@args{#1}{#2}%
1688 %

```

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

```

1689 \global\advance\@edtext@level by 1%
1690 %

```

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

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

```

1691 \stepcounter{edtext}%
1692 \csxdef{thisedtext@the\@edtext@level}{\theedtext}%
1693 %

```

By default, we do not use `\lemma`

```

1694 \global\@lemmacommand@false%
1695 %

```

```

1696 \begingroup%
1697 %

```

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`.

```

1698 \ifledRcol%
1699 \ifcsvoid{sw@list@edtextR@the\@edtext@level}%
1700 {\global\let\sw@inthisedtext\empty}%
1701 {\expandafter\glp\csname sw@list@edtextR@the\
@edtext@level\endcsname\to\sw@inthisedtext}%
1702 \else%
1703 \ifcsvoid{sw@list@edtext@the\@edtext@level}%
1704 {\global\let\sw@inthisedtext\empty}%
1705 {\expandafter\glp\csname sw@list@edtext@the\@edtext@level
\endcsname\to\sw@inthisedtext}%
1706 \fi%
1707 %

```

`\@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.

```

1708 \global\renewcommand{\@tag}{%
1709 \no@expands #1%
1710 }%
1711 %

```

`\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`.

```

1712 \set@line%
1713 %

```

`\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`.

```

1714 \ifledRcol \global\insert@countR \z@%
1715 \else \global\insert@count \z@ \fi%
1716 %

```

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.

```

1717 \@edtext@secondarg@true%
1718 \ignorespaces #2\relax%
1719 \@edtext@secondarg@false%
1720 %

```

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.

```

1721 \if@RTL%
1722 \flag@start@RTL%
1723 \else%
1724 \flag@start%
1725 \fi%
1726 %

```

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

```

1727 \if@lemmacommand%
1728 \ifledRcol%
1729 \write\linenum@outR{\string\@lemma}%
1730 \else%
1731 \write\linenum@out{\string\@lemma}%
1732 \fi%
1733 \fi%
1734 %

```

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.

```

1735 \endgroup%
1736 \ifdef{\hypertarget}%
1737 {%
1738 \Hy@raisedlink@left{\hypertarget{\csuse{thisedtext@the\
@edtext@level}:start}{}}%
1739 \showlemma{#1}%
1740 \Hy@raisedlink{\hypertarget{\csuse{thisedtext@the\@edtext@level}:
end}{}}%
1741 }%
1742 {%
1743 \showlemma{#1}%
1744 }%
1745 %

```

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.

```

1746 \ifx\end@lemmas\empty \else%
1747 \glp\end@lemmas\to\x@lemma%
1748 \x@lemma%
1749 \global\let\x@lemma=\relax%
1750 \fi%
1751 \if@RTL%
1752 \flag@end@RTL%
1753 \else%
1754 \flag@end%
1755 \fi%
1756 %

```

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`.

```

1757 \global\@noneed@Footnotefalse%
1758 \global\advance\@edtext@level by -1%
1759 \global\@lemmacommand@false%
1760 %

```

We also reset `\@beforeinsertofthisedtext`

```

1761 \global\let\@beforeinsertofthisedtext\relax%
1762 %

```

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

```

1763 \else%
1764 \showlemma{#1} (\textbf{\textsc{Edtext outside numbered paragraph}})\
    led@err@edtextoutsidestart%
1765 \fi%
1766 }%
1767
1768
1769 %

```

`\@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.

```

1770 \newcommand{\@check@edtext@args}[2]{%
1771 \begingroup%
1772 \noexpandarg%
1773 \IfSubStr{#1}{\pstart}{\led@err@PstartInEdtext{first}}{}%
1774 \IfSubStr{#1}{\pend}{\led@err@PendInEdtext{first}}{}%
1775 \IfSubStr{#2}{\pstart}{\led@err@PstartInEdtext{second}}{}%
1776 \IfSubStr{#2}{\pend}{\led@err@PendInEdtext{second}}{}%
1777 \endgroup%
1778 }%
1779 %
1780 %

```

`\@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`.

```

1781 \let\@beforeinsertofthisedtext\relax
1782 %

```

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

```

1783 \newif\ifnumberline
1784 \numberlinetrue
1785 %

```

`\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.

```

1786 \newcommand*{\set@line}{%
1787 \ifledRcol

```

```

1788 \ifx\line@listR\empty
1789 \global\noteschanged@true
1790 \xdef\l@d@nums{000|000|000|000|000|000|\edfont@info}%
1791 \else
1792 \gl@p\line@listR\to\@tempb
1793 \xdef\l@d@nums{\@tempb|\edfont@info}%
1794 \global\let\@tempb=\undefined
1795 \fi
1796 \ifx\annot@listR\empty%
1797 \xdef\l@current@annotR{||}%
1798 \else%
1799 \gl@p\annot@listR\to\@tempb%
1800 \xdef\l@current@annotR{\@tempb}%
1801 \global\let\@tempb=\undefined%
1802 \fi%
1803 \else
1804 \ifx\line@list\empty
1805 \global\noteschanged@true
1806 \xdef\l@d@nums{000|000|000|000|000|000|\edfont@info}%
1807 \else
1808 \gl@p\line@list\to\@tempb
1809 \xdef\l@d@nums{\@tempb|\edfont@info}%
1810 \global\let\@tempb=\undefined
1811 \fi
1812 \ifx\annot@list\empty%
1813 \xdef\l@current@annot{||}%
1814 \else%
1815 \gl@p\annot@list\to\@tempb%
1816 \xdef\l@current@annot{\@tempb}%
1817 \global\let\@tempb=\undefined%
1818 \fi%
1819 \fi}
1820
1821 %

```

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

```

1822 \newcommand*{\edfont@info}{\f@encoding/\f@family/\f@series/\f@shape}
1823
1824 %

```

## 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. 140).

```

1825 \newcommand*{\lemma}[1]{%
1826 \global\@lemmacommand@true%

```



```

1827 \global\renewcommand{\@tag}{%
1828 \no@expands #1%
1829 }%
1830 \ignorespaces%
1831 }%
1832 %

```

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

```

1833 \newcommand{\@lemma}{%
1834 \booltrue{lemmacommand@the\@edtext@level}%
1835 }%
1836 %

```

**\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.

```

1837 \newif\if@lemmacommand@%
1838 %

```

### 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. 109): 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.

```

1839 \newcommand*{\linenum}[1]{%
1840 \xdef\@tempa{#1||||||\noexpand\\l@d@nums}%
1841 \global\let\l@d@nums=\empty
1842 \expandafter\line@set\@tempa|\\ignorespaces}
1843 %

```

**\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.

```

1844 \def\line@set#1|#2|#3|#4|{%
1845 \gdef\@tempb{#1}%

```

```

1846 \ifx\@tempb\empty
1847 \l@d@add{#3}%
1848 \else
1849 \l@d@add{#1}%
1850 \fi
1851 \gdef\@tempb{#4}%
1852 \ifx\@tempb\empty\else
1853 \l@d@add{|\line@set#2\|#4\}%
1854 \fi}
1855 %

```

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

```

1856 \newcommand{\l@d@add}[1]{\xdef\l@d@nums{\l@d@nums#1}}
1857
1858 %

```

`\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`

```

1859 \newcommand*{\lineannot}[1]{%
1860 \lineannot@set#1|}%
1861 }%
1862 \def\lineannot@set#1|#2|{%
1863 \expandafter\parse@annot#1|#2|}%
1864 \IfStrEq{#1}{}%
1865 {\let\@tempa\annot@start}%
1866 {\def\@tempa{#1}}}%
1867 \IfStrEq{#2}{}%
1868 {\let\@tempb\annot@start}%
1869 {\def\@tempb{#2}}}%
1870 \xdef\l@current@annot{\@tempa|\@tempb}%
1871 }%
1872 %

```

## 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 `\sameword` 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 `\sameword` in first argument of `\edtext`.

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

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

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

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

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

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

- At the second run, when a critical notes is called, we associate it to the next item of the list associated to its `\edtext` level. So, in the previous example:
  - Critical notes (A) and (B) are associated with  $\{1, 2, 2, 3\}$ .
  - Critical note (C) is associated with  $\{1, 2, 2, 3, 4, 3\}$ .
  - Critical note (D) is associated with  $\{5\}$ .
  - Critical note (E) is associated with  $\{5, 4\}$ .
- At the second run, when a critical note is printed:
  - The `\sameword` command is let `\sameword@inedtext`.
  - At each call of this `\sameword@inedtext`, we step to the next element of the list associated to the note. Let it be  $r$ .
  - For the word marked by `\sameword`, we calculate how many time it is called in its line. To do it:
    - \* We get the absolute line number of the current `\sameword`. This absolute line number was stored with a list of relative ranks for the current `\edtext`. That means, in the previous example, that if the absolute line number of `\edtext` was 1, that critical notes (A) and (B) were not associated with  $\{1, 2, 2, 3\}$  but with  $\{(1, 1), (2, 1), (2, 1), (3, 1)\}$ . Such a method of knowing the absolute line number associated to a `\sameword` is required because a `\edtext` can overlap many lines, but `\sameword` can't get it.
    - \* When reading the auxiliary file, we get the value associated to the pair composed by the current marked word and the current absolute line number. To this value, we subtract the value associated to the pair composed by the current marked word and the previous absolute line number. Let the result be  $n$ .
  - If  $n > 1$ , that means the current word appears more than once in its line. In this case, we call `\showwordrank` with the word as the first argument and  $r$  as the second argument. If the word is called only once, we just print it.

After theory, implementation.

`\get@sw@txt` As the argument of `\sameword` can contain an active character if we use `inputenc` with `utf8` option instead of native UTF-8 engine, we store its detokenized content in a macro in order to allow the dynamic name of macro with `\csname`.<sup>30</sup>

<sup>30</sup>See <http://tex.stackexchange.com/q/244538/7712>.

Because there is a bug with `\detokenize` and  $\text{\XeTeX}$  when using non BMP characters<sup>31</sup>, we detokenize only for non- $\text{\XeTeX}$  engines. In any case, in  $\text{\XeTeX}$  `\csname` construction can contain UTF-8 characters without a problem, as UTF-8 characters are not managed with category codes, but instead read directly as UTF-8 characters.

```
1873 \newcommand{\get@sw@txt}[1]{%
1874   \begingroup%
1875   \swnoexpands%
1876   %.
```

Using case sensibility option.

```
1877   \ifsw@caseinsensitive%
1878   \def\@tmpa##1{\lowercase{##1}}%
1879   \else%
1880   \def\@tmpa##1{##1}%
1881   \fi%
1882   %
```

And now, define `\sw@txt`.

```
1883   \ifxetex%
1884   \@tmpa{\xdef\sw@txt{##1}}%
1885   \else%
1886   \@tmpa{\expandafter\xdef\expandafter\sw@txt\expandafter{\detokenize
1887     {##1}}}%
1887   \fi%
1888   \endgroup%
1889 }%
1890 %
```

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\newcommand{\swnoexpands}{%
1892   \let\sameword\l@secondmandarg%Allow to have nested \sameword
1893   \let\emph\@firstofone%
1894   \let\textit\@firstofone%
1895   \let\textbf\@firstofone%
1896   \let\textsc\@firstofone%
1897   \let\framebox\@firstofone%
1898   \let\edtext\dummy@edtext%
1899   \RenewExpandableDocumentCommand{\edindex}{om}{}%
1900   \ifdefined\index%
1901     \RenewExpandableDocumentCommand{\index}{om}{}%
1902   \fi%
1903   \let\selectlanguage\@gobble%
1904   \let\foreignlanguage\@secondoftwo%
```

<sup>31</sup><http://sourceforge.net/p/xetex/bugs/108/>

```

1905 \ifdefined\xpg@loaded%
1906 \renewcommand\do[1]{%
1907 \expandafter\RenewExpandableDocumentCommand\expandafter{\csname
text##1\endcsname}{om}{####2}%
1908 }%
1909 \expandafter\docsvlist\expandafter{\xpg@loaded}%
1910 \fi%
1911 }%
1912 %

```

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

```

1913 \newcommandx{\sameword}[2][1,usedefault]{%
1914 \leavevmode%
1915 \get@sw@txt{#2}%
1916 %

```

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

```

1917 \unless\ifledRcol%
1918 \csnumgdef{sw@\sw@txt}{\csuse{sw@\sw@txt}+\@ne}%
1919 %

```

Then, write its value to the numbered file.

```

1920 \protected@write\linenum@out{\string\@sw{\sw@txt}{\csuse{sw@\sw@txt}
}}{#1}}%
1921 %

```

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

```

1922 \else%
1923 \csnumgdef{sw@\sw@txt}{\csuse{sw@\sw@txt}+\@ne}%
1924 \protected@write\linenum@outR{\string\@sw{\sw@txt}{\csuse{sw@\sw@txt}
}}{#1}}%
1925 \fi%
1926 %

```

And print the word.

```

1927 #2%
1928 }%
1929 %

```

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

```

\if@addsw30 \newif\if@addsw%
1931 %

```

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

```

1932 \newcommand{\@sw}[3]{%
1933   \get@sw@txt{#1}%
1934   \unless\ifledRcol%
1935   %

```

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

```

1936   \csxdef{sw@\sw@txt @\the\absline@num @\the\section@num}{#2}%
1937   %

```

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

```

1938   \numdef{\prev@line}{\the\absline@num-1}%
1939   \ifcsundef{sw@\sw@txt @\prev@line @\the\section@num}{%
1940     \csnumgdef{sw@\sw@txt @\prev@line @\the\section@num}{#2-1}%
1941   }{}%
1942   %

```

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

```

1943   \numdef{\the@sw}{#2-\csuse{sw@\sw@txt @\prev@line @\the\section@num}}%
1944   %

```

And do the same thing for the right side.

```

1945   \else%
1946     \csxdef{sw@\sw@txt @\the\absline@numR @\the\section@numR @R}{#2}%
1947     \numdef{\prev@line}{\the\absline@numR-1}%
1948     \ifcsundef{sw@\sw@txt @\prev@line @\the\section@numR @R}{%
1949       \csnumgdef{sw@\sw@txt @\prev@line @\the\section@numR @R}{#2-1}%
1950     }{}%
1951     \numdef{\the@sw}{#2-\csuse{sw@\sw@txt @\prev@line @\the\section@numR @R}}%
1952   \fi%
1953   %

```

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

```

1954   \@tempcnta=\@edtext@level
1955   \@whilenum{\@tempcnta>0}\do{%
1956     \ifcsdef{sw@list@edtext@tmp@\the\@tempcnta}%
1957     {%
1958       \@addswfalse%
1959       \notbool{lemmacommand@\the\@tempcnta}%
1960       {\@addswtrue}%
1961       {\IfStrEq{#3}{inlemma}%
1962        {\@addswtrue}%
1963       }%
1964       \def\do##1{%
1965         \ifnumequal{##1}{\the\@tempcnta}%
1966         {\@addswtrue\listbreak}%
1967         }%
1968     }%

```

```

1969         \docsvlist{#3}%
1970     }%
1971 }%
1972 \if@addsw%
1973     \letcs{\@tmp}{sw@list@edtext@tmp@the\@tempcnta}%
1974     \ifledRcol%
1975         \xright@appenditem{\the@sw}{\the\absline@numR}}\to\@tmp%
1976     \else%
1977         \xright@appenditem{\the@sw}{\the\absline@num}}\to\@tmp%
1978     \fi%
1979     \cslet{sw@list@edtext@tmp@the\@tempcnta}{\@tmp}%
1980 \fi%
1981 }%
1982 {}%
1983 \advance\@tempcnta by -1%
1984 }%
1985 }%
1986 %

```

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

```

1987 \newcommandx{\sameword@inedtext}[2][1,usedefault]{%
1988     \get@sw@txt{#2}%
1989     \unless\ifledRcol%
1990 %

```

Just a precaution.

```

1991     \ifx\sw@list@inedtext\empty%
1992         \def\the@sw{999}%
1993         \def\this@absline{-99}%
1994     \else%
1995 %

```

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

```

1996     \gl@p\sw@list@inedtext\to\@tmp%
1997     \edef\the@sw{\expandafter\@firstoftwo\@tmp}%
1998     \edef\this@absline{\expandafter\@secondoftwo\@tmp}%
1999 \fi%
2000 %

```

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

```

2001     \ifcsdef{sw@\sw@txt @\this@absline @\the\section@num}{%
2002         \numdef{\prev@line}{\this@absline-1}%
2003         \numdef{\sw@atthisline}{\csuse{sw@\sw@txt @\this@absline @\the\
section@num}-\csuse{sw@\sw@txt @\prev@line @\the\section@num}}%
2004     }%
2005     {\numdef{\sw@atthisline}{0}}%
2006 %

```



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

```

2007 \ifnumgreater{\sw@atthisline}{1}%
2008     {\showwordrank{#2}{\the@sw}}%
2009     {#2}%
2010 %

```

And the same for right side.

```

2011 \else%
2012     \ifx\sw@list@inedtext\empty%
2013         \def\the@sw{999}%
2014         \def\this@absline{-99}%
2015     \else%
2016         \gl@p\sw@list@inedtext\to\@tmp%
2017         \edef\the@sw{\expandafter\@firstoftwo\@tmp}%
2018         \edef\this@absline{\expandafter\@secondoftwo\@tmp}%
2019     \fi%
2020     \ifcsdef{sw\sw@txt @\this@absline @\the\section@numR @R}{%
2021         \numdef{\prev@line}{\this@absline-1}%
2022         \numdef{\sw@atthisline}{\csuse{sw\sw@txt @\this@absline @\the\
section@numR @R}-\csuse{sw\sw@txt @\prev@line @\the\section@numR @R}}%
2023     }%
2024     {\numdef{\sw@atthisline}{0}}%
2025     \ifnumgreater{\sw@atthisline}{1}%
2026         {\showwordrank{#2}{\the@sw}}%
2027         {#2}%
2028     \fi%
2029 }%
2030 %

```

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

```

2031 \newcommand{\showwordrank}[2]{%
2032     #1\textsuperscript{#2}%
2033 }%
2034 %

```

## 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.

```

2035 \newbox\raw@text
2036 \newif\ifnumberedpar@
2037 \newcount\num@lines
2038 \newbox\one@line
2039 \newcount\par@line
2040 %

```

`\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` typset 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` typset 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`.

```

2041 \newcount\pstarts@typeset@L%
2042 \newcount\pstarts@read@L%
2043 %

```

`\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.

`\numberpstarttrue` 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.

`\numberpstartfalse`

`\labelpstarttrue`

`\labelpstartfalse`

```

\thepstart
2044 \ifat@every@pstart@star@
2045 \newcommand{\AtStartEveryPstart}[1]{%
2046   \ifstrepty{#1}%
2047   {\gdef\@at@start@every@pstart{}}%
2048   {\gdef\@at@start@every@pstart{#1}}%
2049 }%
2050 \def\@at@start@every@pstart{}%
2051
2052 \newif\ifat@every@pstart@star@%

```

```

2053 \newcommand{\AtEveryPstart}[1]{%
2054   \ifstrempy{#1}%
2055     {\gdef\at@every@pstart{}}%
2056     {\gdef\at@every@pstart{\noindent#1}}%
2057   \global\at@every@pstart@star@false%
2058 }%
2059 \WithSuffix\newcommand\AtEveryPstart*[1]{%
2060   \ifstrempy{#1}%
2061     {\gdef\at@every@pstart{}}%
2062     {\gdef\at@every@pstart{#1}}%
2063   \global\at@every@pstart@star@true%
2064 }%
2065 \def\at@every@pstart{}%
2066
2067 \newcounter{pstart}
2068 \renewcommand{\thepstart}{\bfseries\@arabic\c@pstart}. }
2069 \newif\ifnumberpstart
2070 \numberpstartfalse
2071 \newif\iflabelpstart
2072 \labelpstartfalse
2073 \newcommandx*\pstart}[2][1,2,usedefault]{%
2074   \normal@pars%
2075   \ifboolexpr{%
2076     test {\ifstrempy{#1}}%
2077     and test {\ifstrempy{#2}}%
2078   }%
2079     {\at@every@pstart}%
2080     {%
2081       \ifstrempy{#1}{\noindent#1}%
2082       \ifstrempy{#2}{#2}%
2083     }%
2084   \ifautopar%
2085     \autopar%
2086   \fi%
2087   \ifluatex%
2088     \edef\l@luatextextdir@L{\the\textdir}%
2089   \fi%
2090   \@nbreaktrue%
2091   \ifnumbering \else%
2092     \led@err@PstartNotNumbered%
2093     \beginnumbering%
2094   \fi%
2095   \ifnumberedpar@%
2096     \led@err@PstartInPstart%
2097   \pend%
2098   \fi%
2099   \list@clear{\inserts@list}%
2100   \global\let\next@insert=\empty%
2101   \begingroup\normal@pars%
2102   \global\advance \l@dnumstartsL\@ne

```

```

2103 \global\advance \pstarts@typeset@L\@ne%
2104 \global\advance \pstarts@read@L\@ne%
2105 \global\setbox\raw@text=\vbox\bgroup%
2106 \if@nobreak%
2107 \if@afterindent\else%
2108 \noindent%
2109 \global\@afterindenttrue%
2110 \fi%
2111 \fi%
2112 \ifboolexpr{%
2113 bool{autopar}%
2114 and bool{by@autopar}%
2115 }%
2116 {}%
2117 {%
2118 \ifnumberpstart%
2119 \ifinstanza\else%
2120 \ifsidepstartnum\else%
2121 \thepstart%
2122 \fi%
2123 \fi%
2124 \fi%
2125 }%
2126 \numberedpar@true%
2127 \iflabelpstart%
2128 \protected@edef\@currentlabel{\p@pstart\thepstart}%
2129 \fi%
2130 \l@ddzeropenalties%
2131 \@at@start@every@pstart%
2132 \global\by@autoparfalse%
2133 \ignorespaces%because not automatically ignored if an optional argument
is used (classical TeX behavior for space after commands)
2134 }
2135 %

```

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

```

2136 \newcommand*{\pend}[2][1,2,usedefault]{\ifnumbering \else%
2137 \led@err@PendNotNumbered%
2138 \fi%
2139 \global\l@dskipversenumberfalse%
2140 \ifnumberedpar@ \else%
2141 \led@err@PendNoPstart%
2142 \fi%
2143 %

```

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.

```

2144 \l@dzzeropenalties%
2145 \@at@end@every@pend%
2146 \endgraf\global\num@lines=\prevgraf\egroup%
2147 \global\par@line=0%
2148 %

```

We check if lineation is by pstart: in this case, we reset the line number, but only in the second line of the pstart. We can't reset line number at the beginning of \pstart, as \setline is parsed at the end of previous \pend, and so, we must do it at the end of first line of pstart.

```

2149 \csnumdef{pstartline}{0}%
2150 \loop\ifvbox\raw@text%
2151   \csnumdef{pstartline}{\pstartline+\@ne}%
2152   \do@line%
2153   \ifbypstart@%
2154     \ifnumequal{\pstartline}{1}{%
2155       \bgroup%
2156       \let\leavevmode\relax%
2157       \setline{1}%
2158       \egroup%
2159       \resetprevline@}{}%
2160     \fi%
2161   \repeat%
2162 %

```

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

```

2163 \flush@notes%
2164 \endgroup%
2165 \ignorespaces%
2166 %

```

Increase pstart counter.

```

2167 \ifnumberpstart%
2168   \global\pstartnumtrue%
2169 \fi%
2170 \addtocounter{pstart}{1}%
2171 \ifcontinuousnumberingwithcolumns%
2172   \addtocounter{pstartL}{1}%
2173   \addtocounter{pstartR}{1}%
2174 \fi%
2175 %

```

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

```

2176 \normal@pars%
2177 \ifboolexpr{%
2178   test {\ifstrempy{#1}}%

```

```

2179 and test {\ifstrempy{#2}}%
2180 }%
2181 {\at@every@pend}%
2182 {%
2183 \ifstrempy{#1}{\noindent#1}%
2184 \ifstrempy{#2}{\#2}%
2185 }%
2186 %

```

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.

```

2187 \@nobreakfalse%
2188 \ifautopar%
2189 \autopar%
2190 \fi%
2191 }
2192 %

```

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.

```

\AtEveryPend93
\at@every@pend94
\ifat@every@pend@star@95 \newif\ifat@every@pend@star@%
2196 \newcommand{\AtEveryPend}[1]{%
2197 \ifstrempy{#1}%
2198 {\gdef\at@every@pend{}}%
2199 {\gdef\at@every@pend{\noindent#1}}%
2200 \global\at@every@pend@star@false%
2201 }%
2202 \WithSuffix\newcommand\AtEveryPend*[1]{%
2203 \ifstrempy{#1}%
2204 {\gdef\at@every@pend{}}%
2205 {\gdef\at@every@pend{#1}}%
2206 \global\at@every@pend@star@true%
2207 }%
2208 \xdef\at@every@pend{}%
2209
2210 %

```

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

```

2211 \newcommand{\AtEndEveryPend}[1]{%
2212 \ifstrempy{#1}%
2213 {\xdef\@at@end@every@pend{}}%
2214 {\gdef\@at@end@every@pend{#1}}%
2215 }%
2216 \def\@at@end@every@pend{}%
2217 %

```

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

```

2218 \newcommand*\l@dzzeropenalties{%
2219   \brokenpenalty \z@ \clubpenalty \z@
2220   \displaywidowpenalty \z@ \interlinepenalty \z@ \predisplaypenalty \z@
2221   \postdisplaypenalty \z@ \widowpenalty \z@}
2222
2223 %

```

`\autopar` In most cases it is only an annoyance to have to label the paragraphs to be numbered with `\pstart` and `\pend`. `\autopar` will do that automatically, allowing you to start a paragraph with its first word and no other preliminaries, and to end it with a blank line or a `\par` command. The command should be issued within a group, after `\beginnumbering` has been used to start the numbering; all paragraphs within the group will be affected.

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

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

The functioning of this macro is more tricky than the usual `\everypar`: we do not want anything to go onto the vertical list at all, so we have to end the paragraph, erase any evidence that it ever existed, and start it again using `\pstart`. We remove the paragraph-indentation box using `\lastbox` and save the width, and then skip backwards over the `\parskip` that has been added for this paragraph. Then we start again with `\pstart`, restoring the indentation that we saved, and locally change `\par` so that it will do our `\pend` for us.

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.

```

2224 \newif\ifautopar
2225 \newif\ifby@autopar%
2226 \newcommand*\autopar{%
2227   \ifledRcol
2228     \ifnumberingR \else
2229     \led@err@AutoparNotNumbered
2230     \beginnumberingR
2231     \fi
2232   \else
2233     \ifnumbering \else
2234     \led@err@AutoparNotNumbered
2235     \beginnumbering

```

```

2236 \fi
2237 \fi
2238 \autopartrue
2239 \everypar{\setbox0=\lastbox
2240 \endgraf \vskip-\parskip
2241 \global\by@autopartrue%
2242 \pstart \noindent \kern\wd0%
2243 \ifnumberpstart%
2244 \ifinstanza\else%
2245 \thepstart%
2246 \fi%
2247 \fi%
2248 \let\par=\pend}%
2249 \ignorespaces}
2250 %

```

**\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.

```

2251 \newcommand*{\normal@pars}{\everypar{} \let\par\endgraf}
2252
2253 %

```

**\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.

```

2254 \newif\ifautopar@pause
2255 %

```

## 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.

```

2256 \newcommand*{\l@dunhbox@line}[1]{\unhbox #1}
2257 \newcommand*{\do@line}{%
2258 {\vbadness=10000
2259 \splittopskip=\z@
2260 \do@linehook
2261 \l@demptyd@ta
2262 \global\setbox\one@line=\vsplit\raw@text to\baselineskip}%
2263 \unvbox\one@line \global\setbox\one@line=\lastbox
2264 \getline@num
2265 \IfStrEq{\led@pb@setting}{before}{\led@check@pb\led@check@nopb}{}}

```



```

2266 \ifnum\@lock>\@ne
2267 \inserthangingsymboltrue
2268 \else
2269 \inserthangingsymbolfalse
2270 \fi
2271 \check@pb@in@verse
2272 \ifl@dhidenumber%
2273 \global\l@dhidenumberfalse%
2274 \f@x@l@cks%
2275 \else%
2276 \affixline@num%
2277 \fi%
2278 %

```

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

```

2279 \xifinlist{\the\pstarts@typeset@L}{\eled@sections@@}%
2280 {\print@eledsection}%
2281 {\print@line}%
2282 \IfStrEq{\led@pb@setting}{after}{\led@check@pb\led@check@nopb}{\}
2283 }%
2284 %

```

### VII.2.2 Process for “normal” line

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

```

2285 \def\print@line{
2286 %

```

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

```

2287 \affixpstart@num%
2288 %

```

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

```

2289 \hb@xt@ \linewidth{%
2290 %

```

User hooks.

```

2291 \do@insidelinehook%
2292 \csuse{insidethis@the\absline@num @the\section@num}%
2293 \global\csundef{insidethis@the\absline@num @the\section@num}%
2294 %

```

Left line number

```

2295 \l@dld@ta%
2296 %

```

Prepare text to be inserted before notes.

```

2297 \if@firstlineofpage%
2298 \set@Xtxtbeforenotes%
2299 \set@txtbeforenotesX%
2300 \global\@firstlineofpagefalse%
2301 \fi%
2302 %

```

Insert footnotes made of manuscripts data and critical footnotes.

```

2303 \ifdefstring{\ms@data@position}{msdata-regular}{%
2304 \insert@msdata%
2305 \add@inserts%
2306 \add@Xgroupbyline%
2307 }{%
2308 \add@inserts%
2309 \add@Xgroupbyline%
2310 \insert@msdata%
2311 }%
2312 %

```

Insert marginal notes.

```

2313 \affixside@note%
2314 %

```

Print left notes.

```

2315 \l@dlsn@te
2316 %

```

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

```

2317 {\ledllfill\hb@xt@ \wd\one@line{%
2318 \new@line%
2319 %

```

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

```

2320 \continuousnumberingwithcolumns@sync@linenumber@singletext%
2321 %

```

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

```

2322 \ifluatex%
2323 \texdir\l@luatextextdir@L%
2324 \fi%
2325 %

```

Insert, if needed, the hanging symbol.

```

2326 \inserthangingsymbol%
2327 %

```

And so, print the line.

```

2328 \l@dunhbox@line{\one@line}}%
2329 %

```

Right line number

```
2330 \ledrlfill\l@drd@ta%
2331 %
```

Print right notes.

```
2332 \l@drsn@te%
2333 }}%
2334 %
```

And reinsert penalties (for page breaking)...

```
2335 \add@penalties%
2336 }
2337 %
```

### 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.

```
2338 \def\print@eledsection{%
2339 \if@firstlineofpage%
2340 \set@Xtxtbeforenotes%
2341 \set@txtbeforenotesX%
2342 \global\@firstlineofpagefalse%
2343 \fi%
2344 \ifdefstring{\ms@data@position}{msdata-regular}{%
2345 \insert@msdata%
2346 \add@inserts%
2347 \add@Xgroupbyline%
2348 }{%
2349 \add@inserts%
2350 \add@Xgroupbyline%
2351 \insert@msdata%
2352 }%
2353 \affixside@note%
2354 \numdef{\temp@}{\pstarts@typeset@L-1}%
2355 \xifinlist{\temp@}{\eled@sections@@}{\@nbreaktrue}{\@nbreakfalse}%
2356 \@eled@sectioningtrue%
2357 \csuse{eled@sectioning@the\pstarts@typeset@L}%
2358 \@eled@sectioningfalse%
2359 \global\csundef{eled@sectioning@the\pstarts@typeset@L}%
2360 \if@RTL%
2361 \hspace{-3\paperwidth}%
2362 {\hbox{\l@dunhbox@line{\one@line}} \new@line}%
2363 \else%
2364 \hspace{3\paperwidth}%
2365 {\new@line \hbox{\l@dunhbox@line{\one@line}}}%

```

```

2366 \fi%
2367 \vskip-\baselineskip%
2368 \continuousnumberingwithcolumns@sync@linenumber@singletext%
2369 }
2370 %

```

#### VII.2.4 Hooks

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

```

2371 \newcommand*\do@linehook{}
2372 \newcommand*\do@insidelinehook{}
2373 %

```

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

```

2374 \newcommand*\dolinehook[1]{\gdef\do@linehook{#1}}%
2375 \newcommand*\doinsidelinehook[1]{\gdef\do@insidelinehook{#1}}%
2376
2377 %

```

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

```

2378 \newcommand{\doinsidethislinehook}[1]{%
2379 \leavevmode%In case it begins with a \pstart, ensure the \@insidethisline
is written after \@nl
2380 \ifledRcol%
2381 \write\linenum@outR{\string\@insidethisline[\unexpanded{#1}]}%
2382 \else%
2383 \write\linenum@out{\string\@insidethisline[\unexpanded{#1}]}%
2384 \fi%
2385 }%
2386 %

```

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

```

2387 \newcommand{\@insidethisline}[1]{%
2388 \ifledRcol%
2389 \csgappto{insidethisR@the\absline@numR @the\section@numR}{#1}%
2390 \else%
2391 \csgappto{insidethis@the\absline@num @the\section@num}{#1}%
2392 \fi%
2393 }%
2394 %

```

## 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`

```

2395 \newcommand*{\l@emptyd@ta}{%
2396   \gdef\l@dld@ta{}%
2397   \gdef\l@drd@ta{}%
2398   \gdef\l@dcsnotetext@l{}%
2399   \gdef\l@dcsnotetext@r{}%
2400   \gdef\l@dcsnotetext{}%
2401
2402 %

```

`\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.

```

2403 \newcommand{\l@dlsn@te}{%
2404   \ifboolexpr{%
2405     bool {l@dprintingcolumns}%
2406     and bool {ledRcol}%
2407   }{% If we are on a right column
2408     \@tempdima=\@morespace@leftnote@rightcolumn%
2409   }{%
2410     \@tempdima=\z@%
2411   }%
2412   \hb@xt@ \z@{\hss\box\l@dlp@rbox\kern\ledlsnotesep\hskip\@tempdima}%
2413 }%
2414 \newcommand{\l@drsn@te}{%
2415   \ifboolexpr{%
2416     bool {l@dprintingcolumns}%
2417     and not bool {ledRcol}%
2418   }{% If we are on a left column
2419     \@tempdima=\@morespace@rightnote@leftcolumn%
2420   }{%
2421     \@tempdima=\z@%
2422   }%
2423   \hb@xt@ \z@{\hskip\@tempdima\kern\ledrsnotesep\box\l@drp@rbox\hss}%
2424 }%
2425
2426 %

```

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

```

2427 \newcommand*{\ledllfill}{\hfil}
2428 \newcommand*{\ledrlfill}{\hfil}
2429
2430 %

```

## 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.

```

2431 \newcommand*{\getline@num}{%
2432   \global\advance\absline@num \@ne%
2433   \do@actions
2434   \do@ballast
2435   \ifnumberline
2436     \ifsublines@
2437       \ifnum\sub@lock<\tw@
2438         \global\advance\subline@num \@ne
2439       \fi
2440     \else
2441       \ifnum\@lock<\tw@
2442         \global\advance\line@num \@ne
2443         \global\subline@num \z@
2444       \fi
2445     \fi
2446   \fi
2447 }
2448 %

```

`\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. 180).

`\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`

```

2449 \newcount\ballast@count
2450 \newcounter{ballast}
2451 \setcounter{ballast}{0}
2452 %

```

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.

```

2453 \newcommand*{\do@ballast}{\global\ballast@count \z@
2454   \begingroup
2455     \advance\absline@num \@ne
2456     \ifnum\next@actionline=\absline@num
2457       \ifnum\next@action>-1001\relax
2458         \global\advance\ballast@count by -\c@ballast
2459       \fi
2460     \fi

```

```

2461 \endgroup}
2462 %

```

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

**\do@actions@next**

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

```

2463 \newcommand*{\do@actions}{%
2464   \global\let\do@actions@next=\relax
2465   \ifnum\absline@num<\next@actionline\else
2466   %

```

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.

```

2467   \ifnum\next@action>-1001
2468     \global\page@num=\next@action
2469     \ifresumenumbering@start%
2470       \setbox0=\hbox{}}%Required to get the correct page number, when the
\resumenumbering is just after a \newpage
2471       \ifnum\pausenumbering@page@num<\page@num%
2472         \global\resumenumbering@startfalse%
2473       \fi%
2474     \fi%
2475     \ifboolexpr{%
2476       bool{resumenumbering@start}%
2477       and test {\ifdimgreater{\pagedepth}{\z@}}}%
2478     }%
2479     {}%
2480     {\global\@firstlineofpagetrue}%
2481     \ifcsdef{reset@line\the\absline@num @\the\section@num}%
2482     {%
2483       \global\line@num=\z@%
2484       \global\subline@num=\z@%
2485       \resetprevline@%
2486     }%
2487     {}%
2488     \global\resumenumbering@startfalse%
2489     \add@msdata@firstlineofpage%
2490   %

```

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`.)

```

2491   \else
2492     \ifnum\next@action<-4999
2493       \@l@tempcnta=-\next@action

```

```

2494 \advance\@l@dttempcnta by -5001
2495 \ifsublines@
2496 \global\subline@num=\@l@dttempcnta
2497 \else
2498 \global\line@num=\@l@dttempcnta
2499 \fi
2500 %

```

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

```

2501 \else
2502 \@l@dttempcnta=-\next@action
2503 \advance\@l@dttempcnta by -1000
2504 \do@actions@fixedcode
2505 \fi
2506 \fi
2507 %

```

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.

```

2508 \ifx\actionlines@list\empty
2509 \gdef\next@actionline{1000000}%
2510 \else
2511 \gl@p\actionlines@list\to\next@actionline
2512 \gl@p\actions@list\to\next@action
2513 \global\let\do@actions@next=\do@actions
2514 \fi
2515 \fi
2516 %

```

Make the recursive call, if necessary.

```

2517 \do@actions@next}
2518
2519 %

```

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

```

2520 \newcommand*{\do@actions@fixedcode}{%
2521 \ifcase\@l@dttempcnta
2522 \or% % 1001 = starting sublineation
2523 \global\sublines@true
2524 \or% % 1002 = ending sublineation
2525 \global\sublines@false
2526 \or% % 1003 = starting locking number
2527 \global\@lock=\@one
2528 \or% % 1004 = ending locking number
2529 \ifnum\@lock=\two@
2530 \global\@lock=\thr@@

```



```

2531 \else
2532 \global\@lock=\z@
2533 \fi
2534 \or% % 1005 = starting locking subnumber
2535 \global\sub@lock=\@ne
2536 \or% % 1006 = ending locking subnumber
2537 \ifnum\sub@lock=\tw@
2538 \global\sub@lock=\thr@@
2539 \else
2540 \global\sub@lock=\z@
2541 \fi
2542 \or% % 1007 = skipping numbering
2543 \l@dskipnumbertrue
2544 \or% % 1008 = skipping numbering in stanza
2545 \l@dskipversenumbertrue%
2546 \or% % 1009 = hiding number
2547 \l@dhiddenumbertrue
2548 \or% % 1010 = inserting msdata
2549 \add@msdata%
2550 \else
2551 \led@warn@BadAction
2552 \fi}
2553
2554
2555 %

```

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

`\sync@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 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.

```

2556 \newcommand{\continuousnumberingwithcolumns@sync@linenumber@singletext}{%
2557 \ifcontinuousnumberingwithcolumns%
2558 \unless\ifafterendnumberingR%
2559 \new@lineR%
2560 \xappto\next@line@list@stuffR{%
2561 \unexpanded{\global\line@numR=}\the\line@num%
2562 }%
2563 \fi%

```

```

2564 \fi%
2565 }%
2566 %

```

## 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.

```

2567 \newcommand{\linenumannotation}[1]{%
2568   \leavevmode%In case it begins with a \pstart, ensure the \@annot is
written after \@nl
2569   \ifledRcol%
2570     \write\linenum@outR{\string\@annot[#1]}%
2571   \else%
2572     \write\linenum@out{\string\@annot[#1]}%
2573   \fi%
2574 }%
2575 %

```

`\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.

```

2576 \def\Xlinenumannotationposition@side{after}%
2577 %

```

`\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`.

```

2578 \def\Xwraplinenumannotation@ref{\textsuperscript}%
2579 \def\Xwraplinenumannotation@side{\textsuperscript}%
2580 \newtoggle{Xnoidenticallinenumannotation@ref}%
2581 \newtoggle{Xnoidenticallinenumannotation@side}%
2582 %

```

`\@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.

```

2583 \newcommand{\@annot}[1]{%
2584   \store@annot@to@absline{#1}%
2585   \def\current@annot{#1}%
2586 }%
2587 \newcommand{\store@annot@to@absline}[1]{%
2588   \ifledRcol%
2589     \ifcsdef{annotR@the\absline@numR @the\section@numR}{%
2590       \csgappto{annotR@the\absline@numR @the\section@numR}{\
2591         @linenumannotationsep#1}%
2592       }{%
2593         \csgdef{annotR@the\absline@numR @the\section@numR}{#1}%
2594       }%
2595     \else%
2596       \ifcsdef{annot@the\absline@num @the\section@num}{%
2597         \csgappto{annot@the\absline@num @the\section@num}{\
2598           @linenumannotationsep#1}%
2599         }{%
2600           \csgdef{annot@the\absline@num @the\section@num}{#1}%
2601         }%
2602       \fi%
2603     }%
2604   %

```

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

```

2603 \let\current@annot=\empty%
2604 %

```

`\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.

```

2605 \newcommand{\reset@current@annot}{%
2606   \unless\ifnoresetlinenumannotation@

```

```

2607 \let\current@annot\empty%
2608 \fi
2609 }%
2610 %

```

**\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.

```

2611 \def\parse@annot#1|#2|{%
2612   \gdef\annot@start{#1}%
2613   \gdef\annot@end{#2}%
2614 }%
2615 %

```

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

```

\@linenumannotationsep
2616 \newcommand{\setlinenumannotationsep}[1]{\gdef\@linenumannotationsep{#1}}%
2617 \def\@linenumannotationsep{, }%
2618 %

```

## 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 `\line@num ≤ \firstlinenum`, we compare the two directly instead of making these calculations.

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

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

```

2619 \newcommand*{\affixline@num}{%
2620 %

```

No number is attached if `\ifl@dskipnumber` is TRUE (and then it is set to its normal FALSE value). No number is attached if `\ifnumberline` is FALSE (the normal value is TRUE).

```

2621 \ifledgroupnotesL@else
2622 \ifnumberline
2623 \ifl@dskipnumber
2624 \global\l@dskipnumberfalse
2625 \else
2626 \ifsublines@
2627 \l@dttempcntb=\subline@num
2628 \ifnum\subline@num>\c@firstsublinenum
2629 \l@dttempcnta=\subline@num
2630 \advance\l@dttempcnta by-\c@firstsublinenum
2631 \divide\l@dttempcnta by\c@sublinenumincrement
2632 \multiply\l@dttempcnta by\c@sublinenumincrement
2633 \advance\l@dttempcnta by\c@firstsublinenum
2634 \else
2635 \l@dttempcnta=\c@firstsublinenum
2636 \fi
2637 %

```

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.

```

2638 \ch@cksub@l@ck
2639 %

```

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

```

2640 \else
2641 \l@dttempcntb=\line@num
2642 %

```

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

```

2643 \ifx\linenumberlist\empty
2644 \ifnum\line@num>\c@firstlinenum
2645 \l@dttempcnta=\line@num
2646 \advance\l@dttempcnta by-\c@firstlinenum
2647 \divide\l@dttempcnta by\c@linenumincrement
2648 \multiply\l@dttempcnta by\c@linenumincrement
2649 \advance\l@dttempcnta by\c@firstlinenum
2650 \else
2651 \l@dttempcnta=\c@firstlinenum
2652 \fi
2653 \else
2654 %

```

The `\linenumberlist` was not `\empty`, so here is Wayne's numbering mechanism. This takes place in  $\TeX$ 's mouth.

```

2655 \l@dttempcnta=\line@num
2656 \edef\rem@inder{\,\linenumberlist,\number\line@num,}%
2657 \edef\sc@n@list{\def\noexpand\sc@n@list

```

```

2658      #####1,\number\@l@dttempcnta,#####2|{\def\noexpand\rem@inder
{\####2}}}%
2659      \sc@n@list\expandafter\sc@n@list\rem@inder|{%
2660      \ifx\rem@inder\empty%
2661      \advance\@l@dttempcnta\@ne
2662      \fi
2663      \fi
2664 %

```

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

```

2665      \ch@ck@l@ck
2666      \fi
2667 %

```

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

```

2668      \ifnum\@l@dttempcnta=\@l@dttempcntb
2669      \ifl@dskipversenumber\else
2670 %

```

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

For  $\text{\LaTeX}$  we have to consider two column documents as well. In this case Peter Wilson thought we need to put the numbers at the outside of the column — the left of the first column and the right of the second. Do the `twocolumn` stuff before going on with the original code.

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

```

\l@drd@ta
2671      \if@twocolumn
2672      \if@firstcolumn
2673      \gdef\l@dld@ta{\llap{\leftlinenum}}}%
2674      \else
2675      \gdef\l@drd@ta{\rlap{\rightlinenum}}}%
2676      \fi
2677      \else
2678      \ifboolexpr{bool {l@dprintingcolumns} and test {\
ifnumgreater{\line@margin@columns}{\m@ne}}}%
2679      {\@l@dttempcntb=\line@margin@columns}%
2680      {\@l@dttempcntb=\line@margin}%
2681      \ifnum\@l@dttempcntb>\@ne
2682      \advance\@l@dttempcntb \page@num
2683      \fi
2684      \ifboolexpr{%
2685      bool {l@dprintingcolumns}%
2686      and (%

```

```

2687         (test {\ifdefstring{\linenum@OnlyPages@ForColumns}{left
}}%
2688         and test {\ifnumodd{\page@num}}%
2689         )%
2690         or%
2691         (test {\ifdefstring{\linenum@OnlyPages@ForColumns}{right
}}%
2692         and not test {\ifnumodd{\page@num}}%
2693         )%
2694     )%
2695 }%
2696 {%
2697 %
2698     \ifodd\@l@tempcntb%
2699         \gdef\l@drd@ta{\rlap{{\rightlinenum}}}%
2700     \else%
2701         \gdef\l@dld@ta{\llap{{\leftlinenum}}}%
2702     \fi%
2703 }%
2704 \fi
2705 \fi
2706 \fi
2707 %

```

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

```

2708     \f@x@l@cks
2709     \fi
2710     \fi
2711     \fi
2712 }
2713
2714 %

```

`\ch@cksub@l@ck` These macros handle line number locking for `\affixline@num`. `\ch@cksub@l@ck` checks subline locking. If it fails, then we disable the line-number display by setting the counters to arbitrary but unequal values.

```

2715 \newcommand*{\ch@cksub@l@ck}{%
2716     \ifcase\sub@lock
2717     \or
2718         \ifnum\sublock@disp=\@ne
2719             \@l@tempcntb=\z@ \@l@tempcnta=\@ne
2720         \fi
2721     \or
2722         \ifnum\sublock@disp=\tw@ \else
2723             \@l@tempcntb=\z@ \@l@tempcnta=\@ne
2724         \fi
2725     \or

```

```

2726         \ifnum\sublock@disp=\z@
2727             \@l@tempcntb=\z@ \@l@tempcnta=\@ne
2728         \fi
2729     \fi}
2730 %

```

Similarly for line numbers.

```

2731 \newcommand*\ch@ck@l@ck}{%
2732     \ifcase\@lock
2733     \or
2734         \ifnum\lock@disp=\@ne
2735             \@l@tempcntb=\z@ \@l@tempcnta=\@ne
2736         \fi
2737     \or
2738         \ifnum\lock@disp=\tw@ \else
2739             \@l@tempcntb=\z@ \@l@tempcnta=\@ne
2740         \fi
2741     \or
2742         \ifnum\lock@disp=\z@
2743             \@l@tempcntb=\z@ \@l@tempcnta=\@ne
2744         \fi
2745     \fi}
2746 %

```

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

```

2747 \newcommand*\f@x@l@cks}{%
2748     \ifcase\@lock
2749     \or
2750         \global\@lock=\tw@
2751     \or \or
2752         \global\@lock=\z@
2753     \fi
2754     \ifcase\sub@lock
2755     \or
2756         \global\sub@lock=\tw@
2757     \or \or
2758         \global\sub@lock=\z@
2759     \fi}
2760
2761 %

```

## 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:



`\affixpstart@num`  
`\pstartnum`

- 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.

```

2762 \leftpstartnum
2763 \rightpstartnum \newif\ifsidepstartnum
2764 \ifsidepstartnum \newcommand*{\affixpstart@num}{%
2765     \ifsidepstartnum
2766         \if@twocolumn
2767             \if@firstcolumn
2768                 \gdef\l@dld@ta{\llap{\leftpstartnum}}}%
2769             \else
2770                 \gdef\l@drd@ta{\rlap{\rightpstartnum}}}%
2771             \fi
2772         \else
2773             \@l@dttempcntb=\line@margin%
2774             \ifnum\@l@dttempcntb>\@ne
2775                 \advance\@l@dttempcntb \page@num
2776             \fi
2777             \ifodd\@l@dttempcntb
2778                 \gdef\l@drd@ta{\rlap{\rightpstartnum}}}%
2779             \else
2780                 \gdef\l@dld@ta{\llap{\leftpstartnum}}}%
2781             \fi
2782         \fi
2783     \fi
2784 }
2785 %
2786
2787
2788 \newif\ifpstartnum
2789 \pstartnumtrue
2790 \newcommand*{\leftpstartnum}{
2791     \ifpstartnum\thepstart
2792     \kern\linenumsep\fi
2793     \global\pstartnumfalse
2794 }
2795 \newcommand*{\rightpstartnum}{
2796     \ifpstartnum
2797     \kern\linenumsep
2798     \thepstart
2799     \fi
2800     \global\pstartnumfalse
2801 }
```

```
2802 %
```

## 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.

```
2803 \list@create{\inserts@list}
2804 %
```

`\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`.

```
2805 \newcommand*{\add@inserts}{%
2806   \global\let\add@inserts@next=\relax
2807 %
```

If `\inserts@list` is empty, there are not any more notes or insertions for this paragraph, and we need not waste our time.

```
2808 \ifx\inserts@list\empty \else
2809 %
```

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.

```
2810 \ifx\next@insert\empty
2811   \ifx\insertlines@list\empty
2812     \global\noteschanged@true
2813     \gdef\next@insert{100000}%
2814   \else
2815     \gl@p\insertlines@list\to\next@insert
2816   \fi
```

```
2817 \fi
2818 %
```

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.

```
2819 \ifnum\next@insert=\absline@num
2820 \gl@p\inserts@list\to\@insert
2821 \@insert
2822 \global\let\@insert=\undefined
2823 \global\let\next@insert=\empty
2824 \global\let\add@inserts@next=\add@inserts
2825 \fi
2826 \fi
2827 %
```

Make the recursive call, if necessary.

```
2828 \add@inserts@next}
2829
2830 %
```

**\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`.

```
2831 \newcommand{\add@Xgroupbyline}{%
2832 \unless\ifnocritical%
2833 \def\do##1{%Looping on the series
2834 \let\olddo\do%Save the old \do macro, that is this macro itself!
2835 \def\do####1{%Looping on the ##1@forinserting command
2836 \ifcsdef{##1@forinserting####1}{%
2837 \X@beforeinsertion{##1}%
2838 \if@ledgroup%
2839 \global\setbox\@nameuse{mp##1footins}=\vbox%
2840 \else%
2841 \insert\csname ##1footins\endcsname%
2842 \fi%
2843 {%
2844 \nottoggle{Xparindent@##1}{\parindent=\z@}{}%
2845 \ifcsdef{Xhsize\csuse{series@display##1}@##1}%
2846 {\hsize \csuse{Xhsize\csuse{series@display##1}@##1}}%
2847 {}%
2848 \if@ledgroup%
2849 \unvbox\@nameuse{mp##1footins}%
2850 \fi%
2851 \X@atbegininsertion{##1}%
2852 \ifcsstring{series@display##1}%
2853 {%
2854 \Xledsetnormalparstuff{##1}%
```

```

2855 \rule\z@\splittopskip%
2856 }%
2857 {}%
2858 \csuse{##1@forinserting@###1}%
2859 \strut\par%
2860 }%
2861 \global\csundef{##1@forinserting@###1}%
2862 }%
2863 {}%
2864 }%
2865 \ifcsdef{##1@forinserting}{%
2866 \dolistcsloop{##1@forinserting}%
2867 }{}%
2868 \global\csundef{##1@forinserting}%
2869 \let\do\olddo%Restore old do
2870 }%
2871 \dolistloop{\@series}%
2872 \fi%
2873 }%
2874
2875
2876 %

```

## 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@tempcnta is used to calculate and accumulate the penalty; it is initially set to the value of \ballast@count, which has been worked out in \do@ballast above (VIII p. 166). Finally, the penalty is checked to see that it does not go below  $-10000$ .

```

2877 \newcommand*{\add@penalties}{\@l@tempcnta=\ballast@count
2878 \ifnum\num@lines>\@ne
2879 \global\advance\par@line \@ne
2880 \ifnum\par@line=\@ne
2881 \advance\@l@tempcnta \clubpenalty
2882 \fi
2883 \@l@tempcntb=\par@line \advance\@l@tempcntb \@ne
2884 \ifnum\@l@tempcntb=\num@lines
2885 \advance\@l@tempcnta \widowpenalty
2886 \fi
2887 \ifnum\par@line<\num@lines
2888 \advance\@l@tempcnta \interlinepenalty
2889 \fi

```

```

2890 \fi
2891 \ifnum\@l@dtmpcnta=\z@
2892 \relax
2893 \else
2894 \ifnum\@l@dtmpcnta>-10000
2895 \penalty\@l@dtmpcnta
2896 \else
2897 \penalty -10000
2898 \fi
2899 \fi}
2900
2901 %

```

### 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  $\TeX$ , 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

```

2902 \newcommand*{\flush@notes}{%
2903 \iftoggle{notfirstrun@}\jobname.\extensionchars\the\section@num}{%
2904 \@xloop%
2905 \ifx\inserts@list\empty \else%
2906 \gl@p\inserts@list\to\@insert%
2907 \@insert%
2908 \global\let\@insert=\undefined%
2909 \repeat%
2910 }{}}%
2911 }%
2912
2913
2914 %

```

**\@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.

```

2915 \def\@xloop#1\repeat{%
2916   \def\body{#1\expandafter\body\fi}%
2917   \body}
2918
2919 %

```

## 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.

```

2920 \newcommand{\set@Xtxtbeforenotes}{%
2921   \unless\ifnocritical@%
2922   \def\do##1{%
2923     \nottoggle{Xtxtbeforenotesonlyonce@##1}{%
2924       \global\togglefalse{Xtxtbeforenotes@##1@typeset}%
2925     }{}%
2926   }%
2927   \dolistloop{\@series}%
2928   \fi%
2929 }%
2930 %

```

**\set@txtbeforenotesX** The `\set@txtbeforenotesX` does the same for the `\textbeforenotesX`.

```

2931 \newcommand{\set@txtbeforenotesX}{%
2932   \unless\ifnofamiliar@%
2933   \def\do##1{%
2934     \nottoggle{txtbeforenotesonlyonceX@##1}{%
2935       \global\togglefalse{txtbeforenotesX@##1@typeset}%
2936     }{}%
2937   }%
2938   \dolistloop{\@series}%
2939   \fi%
2940 }%
2941 %

```

**\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.

```

2942 \newcommand{\insert@Xtxtbeforenotes}[1]{%
2943   \nottoggle{Xtxtbeforenotes@#1@typeset}{%
2944     \global\toggletrue{Xtxtbeforenotes@#1@typeset}%
2945     \ifcvoid{Xtxtbeforenotes@#1}{%
2946       \ifcsstring{series@display#1}{paragraph}%

```

```

2947     {\noindent\csuse{Xtxtbeforenotes@#1}}}%
2948     {\expandafter\insert\curname#1footins\endcurname%
2949       \bgroup%
2950       \noindent%
2951       \ifcsdef{\csuse{series@display#1}@begin@insert}{%
2952         \csuse{\csuse{series@display#1}@begin@insert}{#1}%
2953       }{}%
2954       \strut\csuse{Xnotefontsize@#1}%
2955       \csuse{Xtxtbeforenotes@#1}%
2956     \egroup%
2957   }%
2958 }%
2959 {}%
2960 {}%
2961 }%
2962
2963
2964 \newcommand{\insert@txtbeforenotesX}[1]{%
2965   \nottoggle{txtbeforenotesX@#1@typeset}{%
2966     \global\toggletrue{txtbeforenotesX@#1@typeset}%
2967     \ifcsvoid{txtbeforenotesX@#1}{%
2968       \ifcsstring{series@displayX#1}{paragraph}%
2969       {\noindent\csuse{txtbeforenotesX@#1}}%
2970       {\expandafter\insert\curname#1\endcurname%
2971         \bgroup%
2972         \noindent%
2973         \ifcsdef{\csuse{series@displayX#1}@begin@insert}{%
2974           \csuse{\csuse{series@displayX#1}@begin@insert}{#1}%
2975         }{}%
2976         \strut\csuse{notefontsizeX@#1}\csuse{txtbeforenotesX@#1}%
2977       \egroup%
2978     }%
2979   }%
2980 }%
2981 {}%
2982 }%
2983
2984
2985 %

```

### XIII Critical footnotes

The footnote macros are adapted from those in PLAIN  $\text{\TeX}$ , but they differ in these respects: the outer-level commands must add other commands to a list macro rather than doing insertions immediately; there are many separate levels of the footnotes, not just one; and there are options to reformat footnotes into paragraphs or into multiple columns.

### 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@lemmafont` `\select@lemmafont` is provided to set the right font for the lemma in a note. This macro extracts the font specifier from the line and page number cluster, and issues the associated font-changing command, so that the lemma is printed in its original font.

```

2986 \def\select@lemmafont#1|#2|#3|#4|#5|#6|#7|{\select@lemmafont#7|}
2987 \def\select@@lemmafont#1/#2/#3/#4|{%
2988   {\fontencoding{#1}\fontfamily{#2}\fontseries{#3}\fontshape{#4}%
2989   \selectfont}
2990
2991 %

```

### 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.

```

2992 \newcommand*\footnoteoptions@[3]{%
2993   \def\do##1{%
2994     \ifstrequal{#1}{L}{% On the left side
2995       \xright@appenditem{\noexpand\setkeys[mac]{#3footnoteoption}{\
2996 unexpanded{##1}}}{\to\inserts@list}%
2997       \global\advance\insert@count \@ne% Increment the left insert
2998 counter.
2999     }%
3000     \xright@appenditem{\noexpand\setkeys[mac]{#3footnoteoption}{\
3001 unexpanded{##1}}}{\to\inserts@listR}%
3002     \global\advance\insert@countR \@ne% Increment the right insert
3003 counter insert.
3004   }%
3005   \notblank{#2}{\docsvlist{#2}}{}}% Parsing all options
3006 }
3007 %

```

### XIII.3 Notes language

`\footnotelang@lua` `\footnotelang@lua` is called to remember the information about the direction of a lemma when Lua<sup>2</sup>TeX is used.

```

3006 \newcommandx*\footnotelang@lua[1][1=L,usedefault]{%
3007   \ifstrequal{#1}{L}{%

```



```

3008 \xright@appenditem{\csxdef{footnote@luatextextdir}{\the\textdir}}\to\
inserts@list%Know the dir of lemma
3009 \global\advance\insert@count \@ne%
3010 \xright@appenditem{\csxdef{footnote@luatexpardir}{\the\pardir}}\to\
inserts@list%Know the dir of lemma
3011 \global\advance\insert@count \@ne%
3012 }%
3013 {%
3014 \xright@appenditem{\csxdef{footnote@luatextextdir}{\the\textdir}}\to\
inserts@listR%Know the dir of lemma
3015 \global\advance\insert@countR \@ne%
3016 \xright@appenditem{\csxdef{footnote@luatexpardir}{\the\pardir}}\to\
inserts@listR%Know the dir of lemma
3017 \global\advance\insert@countR \@ne%
3018 }%
3019 }
3020 %

```

`\footnotelang@poly` `\footnotelang@poly` is called to remember the information about the language of a lemma when polyglossia is used.

```

3021 \newcommand*{\footnotelang@poly}[1][1=L,usedefault]{%
3022 \ifstrequal{#1}{L}{%
3023 \if@RTL%
3024 \xright@appenditem{\csxdef{footnote@dir}{@RTLtrue}}\to\
inserts@list%Know the language used in the lemma
3025 \global\advance\insert@count \@ne%
3026 \else
3027 \xright@appenditem{\csxdef{footnote@dir}{@RTLfalse}}\to\
inserts@list%Know the language of lemma
3028 \global\advance\insert@count \@ne%
3029 \fi%
3030 \xright@appenditem{\csxdef{footnote@lang}{\expandonce\language}}\
to\inserts@list%Know the language of lemma
3031 \global\advance\insert@count \@ne%
3032 }%
3033 {%
3034 \if@RTL
3035 \xright@appenditem{\csxdef{footnote@dir}{@RTLtrue}}\to\
inserts@listR%Know the language of lemma
3036 \global\advance\insert@countR \@ne%
3037 \else
3038 \xright@appenditem{\csxdef{footnote@dir}{@RTLfalse}}\to\
inserts@listR%Know the language of lemma
3039 \global\advance\insert@countR \@ne%
3040 \fi
3041 \xright@appenditem{\csxdef{footnote@lang}{\expandonce\language}}\
to\inserts@listR%Know the language of lemma
3042 \global\advance\insert@countR \@ne%
3043 }%

```

```

3044 }
3045 %

```

### 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 `\Afootnote`, `\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`.

```

3046 \newcommand*{\footsplitskips}{%
3047   \interlinepenalty=\interfootnotelinepenalty
3048   \unless\ifl@dprintingpages%
3049     \floatingpenalty=\@MM%
3050   \fi%
3051   \splittopskip=\ht\strutbox \splitmaxdepth=\dp\strutbox
3052   \leftskip=\z@skip \rightskip=\z@skip}
3053
3054 %

```

`\normalfootnoterule` `\normalfootnoterule` is a standard footnote-rule macro, for use by a `footstart` macro: just the same as the `PLAIN` `TEX` footnote rule.

```
3055 \let\normalfootnoterule=\footnoterule
3056 %
```

## 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.

```
3057 \newcommand{\Xarrangement}[2][1,usedefault]{%
3058   \def\do##1{%
3059     \csname Xarrangement@#2\endcsname{##1}%
3060   }%
3061   \ifstrempy{#1}%
3062     {%
3063       \dolistloop{\@series}%
3064     }%
3065     {
3066       \docsvlist{#1}%
3067     }%
3068   }%
3069   %
```

### 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\vsiz
\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.

```

3070 \newcommand*\Xarrangement@normal}[1]{%
3071   \csgdef{series@display#1}{normal}
3072   \expandafter\let\csname #1footstart\endcsname=\normalfootstart
3073   \expandafter\let\csname v#1footnote\endcsname=\normalvfootnote
3074   \expandafter\let\csname #1footfmt\endcsname=\normalfootfmt
3075   \expandafter\let\csname #1footgroup\endcsname=\normalfootgroup
3076   \expandafter\let\csname #1footnoterule\endcsname=%
3077                                   \normalfootnoterule
3078   \count\csname #1footins\endcsname=1000
3079   \dimen\csname #1footins\endcsname=\csuse{Xmaxhnotes@#1}
3080   \skip\csname #1footins\endcsname=\csuse{Xbeforenotes@#1}%
3081   \advance\skip\csname #1footins\endcsname by\csuse{Xafterrule@#1}%
3082   %

```

The `reledpar` provides tools in order to confine notes to one side. The mechanism is explained in the `reledpar`'s handbook. For now, just retain we need to store default value of the counter associated to the notes  $\TeX$ 's inserts.

```

3083   \csxdef{default@#1footins}{1000}%Use this to confine the notes to one
side only
3084   %

```

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).

```

3085   \ifnoledgroup@else%
3086     \expandafter\let\csname mpv#1footnote\endcsname=\mpnormalvfootnote
3087     \expandafter\let\csname mp#1footgroup\endcsname=\mpnormalfootgroup
3088     \count\csname mp#1footins\endcsname=1000
3089     \dimen\csname mp#1footins\endcsname=\csuse{Xmaxhnotes@#1}
3090     \skip\csname mp#1footins\endcsname=\csuse{Xbeforenotes@#1}%
3091     \advance\skip\csname mp#1footins\endcsname by\csuse{Xafterrule@#1}%
3092   \fi
3093 }
3094
3095 %

```

`\normalvfootnote` We now begin a series of commands that do 'normal' footnote formatting: a format much like that implemented in PLAIN  $\TeX$ , in which each footnote is a separate paragraph.

`\normalvfootnote` takes the series letter as #1 and the entire text of the footnote is #2. It does the `\insert` for this note, calling on the `\footfmt` macro for this note series to format the text of the note.

```

3096 \notbool{parapparatus@}\newcommand*\Xgroupbyline{\normalvfootnote}[2]{%
3097   \iftoggle{Xgroupbyline@#1}{%In the case we use \Xgroupbyline, the
insertion is done later, in \add@Xgroupbyline.
3098   \prepare@Xgroupbyline{#1}{#2}{\normalvfootnote@inserted}%
3099 }{%In the case we don't use \Xgroupbyline, the insertion is made directly
3100   \X@beforeinsertion{#1}%
3101   \insert\csname #1footins\endcsname{%
3102     \X@atbegininsertion{#1}%

```

```

3103 \normalvfootnote@inserted{#1}{#2}%
3104 }%
3105 }%
3106 }%
3107 %

```

**\normalvfootnote@inserted** The `\normalvfootnote@inserted` macro is expanded to the content to be added to a `\insert` for normal critical footnote.

```

3108 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\normalvfootnote@inserted}[2]{%
3109 \nottoggle{Xgroupbyline@#1}{\noindent}{\csuse{Xhooknote@#1}%
3110 \csuse{Xnotefontsize@#1}%
3111 \iftoggle{Xgroupbyline@#1}{\strut}{}%
3112 \footssplitskips
3113 \ifl@dpairing\ifl@dpadding\else%
3114 \setXnoteswidthliketwocolumns@{#1}%
3115 \fi\fi%
3116 \setXnotespositionliketwocolumns@{#1}%
3117 \spaceskip=\z@skip \xspaceskip=\z@skip%
3118 \csname #1footfmt\endcsname #2{#1}%
3119 }%
3120 %

```

```

\X@beforeinsertion21 \newcommand{\X@beforeinsertion}[1]{%
3122 \if@ledgroup\else%
3123 \insert@Xtxtbeforenotes{#1}%
3124 \fi%
3125 \csuse{Xbeforeinserting@#1}%
3126 }%
3127 %

```

```

\beforeinsertion@X28 \newcommand{\beforeinsertion@X}[1]{%
3129 \if@ledgroup\else%
3130 \insert@txtbeforenotesX{#1}%
3131 \fi%
3132 \csuse{beforeinsertingX@#1}%
3133 }%
3134 %

```

```

\X@atbegininsertion35 \newcommand{\X@atbegininsertion}[1]{%
3136 \hspace=\expandafter\dimexpr\csuse{Xwidth@#1}\relax%
3137 }%
3138 %

```

And somewhat different versions of `\normalvfootnote` and `\normalvfootnote@inserted` for minipages.

```

\mpnormalvfootnote 39 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\mpnormalvfootnote}[2]{%
3140 \iftoggle{Xgroupbyline@#1}{%
3141 \prepare@Xgroupbyline{#1}{#2}{\mpnormalvfootnote@inserted}%
3142 }%
3143 {%
3144 \global\setbox\@nameuse{mp#1footins}%
3145 \vbox{%
3146 \unvbox\@nameuse{mp#1footins}%
3147 \mpnormalvfootnote@inserted{#1}{#2}%
3148 }%
3149 }%
3150 }%
3151
3152 %

```

```

\mpnormalvfootnote@inserted 53 \newcommand{\mpnormalvfootnote@inserted}[2]{%
3154 \noindent\csuse{Xbhooknote@#1}%
3155 \csuse{Xnotefontsize@#1}%
3156 \hsize\columnwidth%
3157 \@parboxrestore%
3158 \color@begingroup%
3159 \csname #1footfmt\endcsname #2{#1}\color@endgroup%
3160 }%
3161 %

```

**\normalfootfmt** `\normalfootfmt` is a ‘normal’ macro to take the footnote line and page number information (see V.9 p. 109), 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.

```

3162 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\normalfootfmt}[4]{%
3163 \Xstorelineinfo{#1}{#4}%
3164 \nottoggle{Xgroupbyline@#4}{\Xledsetnormalparstuff{#4}}{%
3165 \Xsethangindent{#4}%
3166 \nottoggle{Xgroupbyline@#4}{\rule\z@\splittopskip}{}%
3167 {\printlinefootnote{#1}{#4}}%
3168 \print@lemma{#1}{#2}{#4}%
3169 \csuse{Xwrapcontent@#4}{#3}%
3170 \nottoggle{Xgroupbyline@#4}{\strut\par}{}%
3171 }%
3172 %

```

**\normalfootstart** `\normalfootstart` is a standard footnote-starting macro, called in the output routine whenever there are footnotes of this series to be printed: it skips a bit and then draws a rule.

Any `\footstart` macro must put onto the page something that takes up space exactly equal to the `\skip\Xfootins` value for the associated series of notes.  $\TeX$  makes page computations based on that `\skip` value, and the output pages will suffer from spacing problems if what you add takes up a different amount of space.

But if the skip `\Xprenotes@` is greater than 0 pt, it is used instead of `\skip\footins` for the first printed series in one page.

The `\leftskip` and `\rightskip` values are both zeroed here. Similarly, these skips are cancelled in the `\vfootnote` macros for the various types of notes. Strictly speaking, this is necessary only if you are using paragraphed footnotes, but we have put it here and in the other `\vfootnote` macros too so that the behavior of `reledmac` in this respect is general across all footnote types. What this means is that any `\leftskip` and `\rightskip` you specify applies to the main text, but not the footnotes. The footnotes continue to be of width `\hsize`.

```
3173 \newcommand*{\normalfootstart}[1]{%
3174 %
```

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. 244. Here is part of this algorithm, when the block of notes are ready to be printed.

```
3175 \ifdimequal{0pt}{\Xprenotes@}{}%
3176 {%
3177   \iftoggle{Xprenotes@}{%
3178     \togglefalse{Xprenotes@}%
3179     \skip\csname #1footins\endcsname=%
3180     \glueexpr\csuse{Xprenotes@}+\csuse{Xafterterrule@#1}\relax%
3181   }%
3182   {}%
3183 }%
3184 \vskip\skip\csname #1footins\endcsname%
3185 %
```

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. 242 for the general description of the problem.

```
3186 \leftskip0pt \rightskip0pt
3187 \ifl@dpairing\else%
3188   \hsize=\old@hsize%
3189 \fi%
3190 \setXnoteswidthliketwocolumns@{#1}%
3191 \setXnotespositionliketwocolumns@{#1}%
3192 %
```

And now, print the footnote's rule to finish the footnote's introduction.

```
3193 \print@Xfootnoterule{#1}%
3194 }%
3195 %
```

`\normalfootgroup` `\normalfootgroup` is a standard footnote-grouping macro: it sends the contents of the footnote-insert box to the output page without alteration.

```

3196 \newcommand*{\normalfootgroup}[1]{%
3197   \csuse{Xhookgroup@#1}%
3198   \unvbox\csname #1footins\endcsname%
3199   \hsize=\old@hsize%
3200 }%
3201
3202 %

```

`\mpnormalfootgroup` A somewhat different version for minipages. Note that, in this case, we do not make distinctions between the `\Xfootgroup` and `\Xfootstarts` macros.

```

3203 \unless\ifnoledgroup@
3204 \newcommand*{\mpnormalfootgroup}[1]{%
3205   \vskip\skip\@nameuse{mp#1footins}
3206   \ifl@dpairing\ifparledgroup%
3207     \leavevmode\marks\parledgroup@{begin}%
3208     \marks\parledgroup@series{#1}%
3209     \marks\parledgroup@type{Xfootnote}%
3210     \fi\fi\normalcolor%
3211     \ifparledgroup%
3212       \ifl@dpairing%
3213       \else%
3214         \setXnoteswidthliketwocolumns@{#1}%
3215         \setXnotespositionliketwocolumns@{#1}%
3216         \print@Xfootnoterule{#1}%
3217       \fi%
3218     \else%
3219       \setXnoteswidthliketwocolumns@{#1}%
3220       \setXnotespositionliketwocolumns@{#1}%
3221       \print@Xfootnoterule{#1}%
3222     \fi%
3223     \setlength{\parindent}{0pt}
3224     \csuse{Xhookgroup@#1}%
3225     \unvbox\csname mp#1footins\endcsname}}
3226 \fi
3227 %

```

### 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.

```

3228 \newcommand*{\Xarrangement@paragraph}[1]{%
3229   \csgdef{series@display#1}{paragraph}
3230   \expandafter\let\csname #1footstart\endcsname=\parafootstart
3231   \expandafter\let\csname v#1footnote\endcsname=\paravfootnote
3232   \expandafter\let\csname #1footfmt\endcsname=\parafootfmt
3233   \expandafter\let\csname #1footgroup\endcsname=\parafootgroup
3234   \count\csname #1footins\endcsname=1000
3235   \csxdef{default@#1footins}{1000}%Use this to confine the notes to one
side only
3236   \dimen\csname #1footins\endcsname=\csuse{Xmaxhnotes@#1}
3237   \skip\csname #1footins\endcsname=\csuse{Xbeforenotes@#1}%
3238   \advance\skip\csname #1footins\endcsname by\csuse{Xafterrule@#1}%
3239   \para@footsetup{#1}
3240   %

```

And the extra setup for minipages.

```

3241 \ifnoledgroup@else
3242   \expandafter\let\csname mpv#1footnote\endcsname=\mpparavfootnote
3243   \expandafter\let\csname mp#1footgroup\endcsname=\mpparafootgroup
3244   \count\csname mp#1footins\endcsname=1000
3245   \dimen\csname mp#1footins\endcsname=\csuse{Xmaxhnotes@#1}
3246   \skip\csname mp#1footins\endcsname=\csuse{Xbeforenotes@#1}%
3247   \advance\skip\csname mp#1footins\endcsname by\csuse{Xafterrule@#1}%
3248   \fi
3249 }
3250 %

```

**`\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.

```

3251 \providecommand{\footfudgefiddle}{64}
3252 %

```

**`\para@footsetup`** `\footparagraph` calls the `\para@footsetup` macro to calculate a special fudge factor, which is the ratio of the `\baselineskip` to the `\hsize`. We assume that the proper value of `\baselineskip` for the footnotes (normally 9 pt) has been set already. The argument of the macro is again the note series letter.

Peter Wilson thinks that `\columnwidth` should be used here for  $\LaTeX$  not `\hsize`. Peter Wilson have also included `\footfudgefiddle`.

```

3253 \newcommand*{\para@footsetup}[1]{\csuse{Xhookgroup@#1}\csuse{
Xnotefontsize@#1}

```

```

3254 \setXnoteswidthliketwocolumns@{#1}%
3255 \ifcsemt{Xwidth@#1}%
3256 {}%
3257 {\columnwidth=\expandafter\dimexpr\csuse{Xwidth@#1}\relax}%
3258 \dimen0=\baselineskip
3259 \multiply\dimen0 by 1024
3260 \divide \dimen0 by \columnwidth \multiply\dimen0 by \footfudgefiddle\
relax
3261 \csxdef{#1footfudgefactor}{%
3262 \expandafter\strip@pt\dimen0 }}
3263
3264 %

```

\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.

```

3265 \newcommand*{\parafootstart}[1]{%
3266 \rightskip=0pt \leftskip=0pt%
3267 \nottoggle{Xparindent@#1}{\parindent=\z@}{}%
3268 \ifdimequal{0pt}{\Xprenotes@}{}%
3269 {%
3270 \iftoggle{Xprenotes@}{%
3271 \togglefalse{Xprenotes@}%
3272 \skip\csname #1footins\endcsname=%
3273 \glueexpr\csuse{Xprenotes@}+\csuse{Xafterterrule@#1}\relax%
3274 }%
3275 }%
3276 }%
3277 \vskip\skip\csname #1footins\endcsname%
3278 \setXnoteswidthliketwocolumns@{#1}%
3279 \setXnotespositionliketwocolumns@{#1}%
3280 \print@Xfootnoterule{#1}%
3281 \let\old@bidi@RTL@everypar\bidi@RTL@everypar%
3282 \let\bidi@RTL@everypar\@empty%
3283 \noindent\leavevmode%
3284 \let\bidi@RTL@everypar\old@bidi@RTL@everypar%
3285 }%
3286 %

```

**\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>32</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>33</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>34</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>35</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 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. 191 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.

```
3287 \newcommand*{\paravfootnote}[2]{%
```

<sup>32</sup>Michael Downes, 'Line Breaking in `\unhboxed` Text', *TUGboat* 11 (1990), pp. 605–612.

<sup>33</sup>See *The TeXbook*, p. 455 (editions after January 1990).

<sup>34</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>35</sup>'Line Breaking', p. 610.

```

3288 \csuse{Xbeforeinserting@#1}%
3289 \insert\csname #1footins\endcsname
3290 \bgroup
3291 \csuse{Xnotefontsize@#1}
3292 \footssplitsskips
3293 \setbox0=\vbox{\hsize=\maxdimen%
3294 \let\bidir@RTL@everypar\@empty%
3295 \insert@Xtxtbeforenotes{#1}%
3296 \noindent\csuse{Xhooknote@#1}%
3297 \csname #1footfmt\endcsname #2{#1}}%
3298 \setbox0=\hbox{\Xunvxh{0}{#1}}%
3299 \dp0=0pt
3300 \ht0=\csname #1footfudgefactor\endcsname\wd0
3301 %

```

Here we produce the contents of the footnote from box 0, and add a penalty of 0 between boxes in this insert.

```

3302 \if@RTL\noindent \leavevmode\fi\box0%
3303 \penalty0
3304 \egroup}
3305 %
3306 %

```

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.

```

3307 \newcommand*{\mpparavfootnote}[2]{%
3308 \global\setbox\@nameuse{mp#1footins}\vbox{%
3309 \unvbox\@nameuse{mp#1footins}%
3310 \csuse{Xnotefontsize@#1}
3311 \footssplitsskips
3312 \setbox0=\vbox{\hsize=\maxdimen%
3313 \let\bidir@RTL@everypar\@empty%
3314 \insert@Xtxtbeforenotes{#1}%
3315 \noindent\color@begingroup%
3316 \csuse{Xhooknote@#1}%
3317 \csname #1footfmt\endcsname #2{#1}\color@endgroup}%
3318 \setbox0=\hbox{\Xunvxh{0}{#1}}%
3319 \dp0=\z@
3320 \ht0=\csname #1footfudgefactor\endcsname\wd0

```

```

3321 \box0
3322 \penalty0
3323 }}
3324
3325 %

```

`\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 T<sub>E</sub>X automatically attaches to the end of paragraphs. When T<sub>E</sub>X 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`.

```

3326 \newcommand*{\Xunvxh}[2]{%
3327   \setbox0=\vbox{\unvbox#1%
3328     \global\setbox1=\lastbox}%
3329   \unhbox1
3330   \unskip           % remove \rightskip,
3331   \unskip           % remove \parfillskip,
3332   \unpenalty        % remove \penalty of 10000,
3333   \hskip\csuse{Xafternote@#2}\relax}% add the glue to go between the notes
3334
3335 %

```

`\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).

```

3336 \newcommand*{\parafootfmt}[4]{%
3337   \Xstorelineinfo{#1}{#4}%
3338   \Xinsertparafootsep{#4}%
3339   \ledsetnormalparstuff@common%
3340   \printlinefootnote{#1}{#4}%
3341   \print@lemma{#1}{#2}{#4}%
3342   \csuse{Xwrapcontent@#4}{#3}%
3343   \penalty-10 }
3344 %

```

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.

```

3345 \newcommand*{\parafootgroup}[1]{%
3346   \hsize=\expandafter\dimexpr\csuse{Xwidth@#1}\relax%
3347   \unvbox\csname #1footins\endcsname
3348   \ifcsstring{Xragged@#1}{L}{\RaggedLeft}{}%
3349   \ifcsstring{Xragged@#1}{R}{\RaggedRight}{}%
3350   \makehboxofhboxes
3351   \setbox0=\hbox{\unhbox0 \removehboxes}%
3352   \csuse{Xbhookgroup@#1}%
3353   \csuse{Xnotefontsize@#1}%
3354   \unhbox0\par%
3355   \global\hsize=\oldhsize%
3356   }%
3357
3358 %

```

`\mpparafootgroup` The minipage version.

```

3359 \newcommand*{\mpparafootgroup}[1]{%
3360   \setXnoteswidthliketwocolumns@{#1}%
3361   \vskip\skip\@nameuse{mp#1footins}
3362   \ifl@dpairing\ifparledgroup%
3363     \leavevmode\marks\parledgroup@{begin}%
3364     \marks\parledgroup@series{#1}%
3365     \marks\parledgroup@type{Xfootnote}%
3366     \fi\fi\normalcolor
3367   \ifparledgroup%
3368     \ifl@dpairing%
3369     \else%
3370       \setXnoteswidthliketwocolumns@{#1}%
3371       \setXnotespositionliketwocolumns@{#1}%
3372       \print@Xfootnoterule{#1}%
3373     \fi%
3374   \else%
3375     \setXnoteswidthliketwocolumns@{#1}%
3376     \setXnotespositionliketwocolumns@{#1}%
3377     \print@Xfootnoterule{#1}%
3378   \fi%
3379   \unvbox\csname mp#1footins\endcsname
3380   \ifcsstring{Xragged@#1}{L}{\RaggedLeft}{}%
3381   \ifcsstring{Xragged@#1}{R}{\RaggedRight}{}%
3382   \makehboxofhboxes
3383   \setbox0=\hbox{\unhbox0 \removehboxes}%
3384   \csuse{Xbhookgroup@#1}%
3385   \csuse{Xnotefontsize@#1}%
3386   \nottoggle{Xparindent@#1}{\parindent=\z@}{}%
3387   \unhbox0\par}}
3388
3389 %

```

And finally, the two macros which are required to transform the long horizontal box stored in the insert' box to a printable text.

```
\makehboxofhboxes%0 \newcommand*\makehboxofhboxes{\setbox0=\hbox{}}%
\removehboxes%01 \loop
3392 \unpenalty
3393 \setbox2=\lastbox
3394 \ifhbox2
3395 \setbox0=\hbox{\box2\unhbox0}%
3396 \repeat}
3397
3398 \newcommand*\removehboxes{\setbox0=\lastbox
3399 \ifhbox0{\removehboxes}\unhbox0 \fi}
3400
3401 %
```

**Insertion of the footnotes separator** The command `\Xinsertparafootsep{<series>}` must be called at the beginning of `\parafootftm`.

```
\prevpage@num%02 \newcommand{\Xinsertparafootsep}[1]{%
\Xinsertparafootsep%03 \ifledRcol%
3404 \ifnumequal{\csuse{#1prevpage@numR}}{\page@numR}%
3405 {\ifcsdef{prevline#1}% Be sur \prevline#1 exists.
3406 {\ifcsequal{prevline#1}{lineinfo@}%
3407 {\ifcsequal{Xsymlinenum@#1}{\csuse{Xparafootsep@#1}}{}}%
3408 {\csuse{Xparafootsep@#1}}}%
3409 }%
3410 {\csuse{Xparafootsep@#1}}}%
3411 }%
3412 {}%
3413 \global\csname #1prevpage@numR\endcsname=\page@numR%
3414 \else%
3415 \ifnumequal{\csuse{#1prevpage@num}}{\page@num}%
3416 {\ifcsdef{prevline#1}% Be sur \prevline#1 exists.
3417 {\ifcsequal{prevline#1}{lineinfo@}%
3418 {\ifcsequal{Xsymlinenum@#1}{\csuse{Xparafootsep@#1}}{}}%
3419 {\csuse{Xparafootsep@#1}}}%
3420 }%
3421 {\csuse{Xparafootsep@#1}}}%
3422 }%
3423 {}%
3424 \global\csname #1prevpage@num\endcsname=\page@num%
3425 \fi%
3426 }
3427 %
```

### 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 a number (#2) of columns, each with a space (#3) between the top baseline and the top of the `\vbox`. The `\rigidbalanceX` 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  $\TeX$  `\line` macro has no relationship to the TeX `\line`. The  $\TeX$  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`.

```

3428 \newcount\@k \newdimen\@h
3429 \newcommand*\Xrigidbalance}[3]{%
3430   \hsize=\expandafter\dimexpr\csuse{Xwidth@\@currentseries}\relax%
3431   \rigidbalance{#1}{#2}{#3}%
3432 }%
3433
3434 \newcommand*\rigidbalanceX}[3]{%
3435   \hsize=\expandafter\dimexpr\csuse{widthX@\@currentseries}\relax%
3436   \rigidbalance{#1}{#2}{#3}%
3437 }%
3438
3439 \newcommand*\rigidbalance}[3]{%
3440   \setbox0=\box#1 \@k=#2 \@h=#3%
3441   \@@line{\splittopskip=\@h \vbadness=\@M \hfilneg
3442     \valign{##\vfil\cr\dosplits}}}%
3443
3444 \newcommand*\dosplits{\ifnum\@k>0 \noalign{\hfil}\splitoff
3445   \global\advance\@k-1\cr\dosplits\fi}
3446
3447 \newcommand*\splitoff{\dimen0=\ht0
3448   \divide\dimen0 by\@k \advance\dimen0 by\@h
3449   \setbox2 \vsplit0 to \dimen0
3450   \unvbox2 }
3451
3452 %

```

#### Three columns

```

\Xarrangement@threecol\newcommand*\Xarrangement@threecol}[1]{%
3454   \csgdef{series@display#1}{threecol}
3455   \expandafter\let\csname v#1footnote\endcsname=\threecolvfootnote
3456   \expandafter\let\csname #1footfmt\endcsname=\threecolfootfmt
3457   \expandafter\let\csname #1footgroup\endcsname=\threecolfootgroup

```



```

3458 \dimen\csname #1footins\endcsname=\csuse{Xmaxhnotes@#1}%
3459 \skip\csname #1footins\endcsname=\csuse{Xbeforenotes@#1}%
3460 \advance\skip\csname #1footins\endcsname by\csuse{Xafterrule@#1}%
3461 \threecolfootsetup{#1}
3462 %

```

The additional setup for minipages.

```

3463 \ifnoledgroup@else
3464   \expandafter\let\csname mpv#1footnote\endcsname=\mpnormalvfootnote
3465   \expandafter\let\csname mp#1footgroup\endcsname=\mpthreecolfootgroup
3466   \skip\csname mp#1footins\endcsname=\csuse{Xbeforenotes@#1}%
3467   \advance\skip\csname mp#1footins\endcsname by\csuse{Xafterrule@#1}%
3468   \mpthreecolfootsetup{#1}
3469 \fi
3470 }
3471 %
3472 %

```

The `\footstart` and `\footnoterule` macros for these notes assume the normal values (XIII.6.2 p. 190 above).

**`\threecolfootsetup`** The `\threecolfootsetup` macro calculates and sets some numbers for three-column footnotes.

We set the `\count` of the foot insert to 333. Each footnote can be thought of as contributing only one third of its height to the page, since the footnote insertion has been made as a long narrow column, which then gets trisected by the `\rigidbalance` routine (inside `\threecolfootgroup`). These new, shorter columns are saved in a box, and then that box is *put back* into the footnote insert, replacing the original collection of the footnotes. This new box is, therefore, only about a third of the height of the original one.

The `\dimen` value for this note series has to change in the inverse way: it needs to be three times the actual limit on the amount of space these notes are allowed to fill on the page, because when  $\TeX$  is accumulating material for the page and checking that limit, it does not apply the `\count` scaling.

```

3473 \newcommand*{\threecolfootsetup}[1]{%
3474   \count\csname #1footins\endcsname 333
3475   \csxdef{default@#1footins}{333}%Use this to confine the notes to one
side only
3476   \multiply\dimen\csname #1footins\endcsname \thr@@}
3477 %

```

**`\mpthreecolfootsetup`** The setup for minipages.

```

3478 \newcommand*{\mpthreecolfootsetup}[1]{%
3479   \count\csname mp#1footins\endcsname 333
3480   \multiply\dimen\csname mp#1footins\endcsname \thr@@}
3481 %
3482 %

```

`\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).

```

3483 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\threecolvfootnote}[2]{%
3484   \iftoggle{Xgroupbyline@#1}{%
3485     \prepare@Xgroupbyline{#1}{#2}{\threecolvfootnote@inserted}%
3486   }%
3487   {%
3488     \X@beforeinsertion{#1}%
3489     \insert\csname #1footins\endcsname{%
3490       \threecolvfootnote@inserted{#1}{#2}%
3491     }%
3492   }%
3493 }%
3494 %

```

```

\threecolvfootnote@inserted%5 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\
threecolvfootnote@inserted}[2]{%
3496   \hsize=\expandafter\dimexpr\csuse{Xwidth@#1}\relax%
3497   \noindent\csuse{Xbhooknote@#1}%
3498   \csuse{Xnotefontsize@#1}%
3499   \footplitskips%
3500   \csname #1footfmt\endcsname #2{#1}%
3501 }%
3502 %

```

`\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.

```

3503 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\threecolfootfmt}[4]{%
3504   \Xstorelineinfo{#1}{#4}%
3505   \threecol@begin@insert{#4}%
3506   \hspace{\parindent}%
3507   \printlinefootnote{#1}{#4}%
3508   \print@lemma{#1}{#2}{#4}%
3509   \csuse{Xwrapcontent@#4}{#3}%
3510   \nottoggle{Xgroupbyline@#4}%
3511   {\strut\par\allowbreak}%
3512   {}%

```

```
3513 }%
3514 %
```

**\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`.

```
3515 \newcommand{\threecol@begin@insert}[1]{%
3516   \normal@pars%
3517   \nottoggle{Xgroupbyline@#1}%
3518   {\hspace \csuse{Xsizethreecol@#1}}%
3519   {}%
3520   \nottoggle{Xparindent@#1}{\parindent=\z@}{}%
3521   \tolerance=5000%
3522   \Xsethangindent{#1}%
3523   \@tempdima=\parindent%
3524   \csuse{Xcolalign@#1}%
3525   \parindent=\@tempdima%
3526   \strut%
3527 }%
3528 %
```

**\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.

```
3529 \newcommand*\threecolfootgroup}[1]{%
3530   \begingroup%
3531   \csuse{Xbhookgroup@#1}%
3532   \csuse{Xnotefontsize@#1}%
3533   \par%
3534   \splittopskip=\ht\strutbox%
3535   \expandafter%
3536   \Xrigidbalance\csname #1footins\endcsname \thr@@ \splittopskip%
3537   \endgroup%
3538 }%
3539 %
```

**\mpthreecolfootgroup** The setup for minipages.

```
3540 \newcommand*\mpthreecolfootgroup}[1]{%
3541   \vskip\skip\@nameuse{mp#1footins}
3542   \ifl@dpairing\ifparledgroup%
```

```

3543 \leavevmode\marks\parledgroup@{begin}%
3544 \marks\parledgroup@series{#1}%
3545 \marks\parledgroup@type{Xfootnote}%
3546 \fi\fi\normalcolor
3547 \ifparledgroup%
3548 \ifl@dpairing%
3549 \else%
3550 \setXnoteswidthliketwocolumns@{#1}%
3551 \setXnotespositionliketwocolumns@{#1}%
3552 \print@Xfootnoterule{#1}%
3553 \fi%
3554 \else%
3555 \setXnoteswidthliketwocolumns@{#1}%
3556 \setXnotespositionliketwocolumns@{#1}%
3557 \print@Xfootnoterule{#1}%
3558 \fi%
3559 \csuse{Xbhookgroup@#1}\par%
3560 \splittopskip=\ht\strutbox
3561 \expandafter
3562 \Xrigidbalance\csname mp#1footins\endcsname \thr@@ \splittopskip}}
3563
3564 %

```

## Two columns

```

\Xarrangement@twocol 3565 \newcommand*{\Xarrangement@twocol}[1]{%
3566 \csgdef{series@display#1}{twocol}
3567 \expandafter\let\csname v#1footnote\endcsname=\twocolvfootnote
3568 \expandafter\let\csname #1footfmt\endcsname=\twocolfootfmt
3569 \expandafter\let\csname #1footgroup\endcsname=\twocolfootgroup
3570 \dimen\csname #1footins\endcsname=\csuse{Xmaxhnotes@#1}%
3571 \skip\csname #1footins\endcsname=\csuse{Xbeforenotes@#1}%
3572 \advance\skip\csname #1footins\endcsname by\csuse{Xafterrule@#1}%
3573 \twocolfootsetup{#1}
3574 %

```

The additional setup for minipages.

```

3575 \ifnoledgroup@else
3576 \expandafter\let\csname mpv#1footnote\endcsname=\mpnormalvfootnote
3577 \expandafter\let\csname mp#1footgroup\endcsname=\mptwocolfootgroup
3578 \skip\csname mp#1footins\endcsname=\csuse{Xbeforenotes@#1}%
3579 \advance\skip\csname mp#1footins\endcsname by\csuse{Xafterrule@#1}%
3580 \mptwocolfootsetup{#1}
3581 \fi
3582 }
3583
3584 %

```

`\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`.

```

\twocolvfootnote@inserted
\twocolfootfmt
\twocolfootgroup
3585 \newcommand*{\twocolfootsetup}[1]{%
3586   \count\csname #1footins\endcsname 500
3587   \csxdef{default@#1footins}{500}%
3588   \multiply\dimen\csname #1footins\endcsname \tw@}
3589   %

3590 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\twocolvfootnote}[2]{%
3591   \iftoggle{Xgroupbyline@#1}{%
3592     \prepare@Xgroupbyline{#1}{#2}{\twocolvfootnote@inserted}%
3593   }{%
3594     \X@beforeinsertion{#1}%
3595     \insert\csname #1footins\endcsname{%
3596       \twocolvfootnote@inserted{#1}{#2}%
3597     }%
3598   }%
3599 }%

3600 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\twocolvfootnote@inserted}[2]{%
3601   \hsize=\expandafter\dimexpr\csuse{Xwidth@#1}\relax%
3602   \noindent\csuse{Xhooknote@#1}%
3603   \csuse{Xnotefontsize@#1}%
3604   \footsplitskips%
3605   \csname #1footfmt\endcsname #2{#1}%
3606 }%
3607 %

3608 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\twocolfootfmt}[4]{% 4th
   arg is optional, for backward compatibility
3609   \Xstorelineinfo{#1}{#4}%
3610   \twocol@begin@insert{#4}%
3611   \hspace{\parindent}%
3612   \printlinefootnote{#1}{#4}%
3613   \print@lemma{#1}{#2}{#4}%
3614   \csuse{Xwrapcontent@#4}{#3}%
3615   \nottoggle{Xgroupbyline@#4}%
3616     {\strut\par\allowbreak}%
3617   {}%
3618 }%
3619 %

3620 \newcommand{\twocol@begin@insert}[1]{%
3621   \normal@pars%
3622   \hsize \csuse{Xhsizetwocol@#1}%
3623   \nottoggle{Xparindent@#1}{\parindent=\z@}{}%
3624   \tolerance=5000%
3625   \Xsethangindent{#1}%

```

```

3626 \@tempdima=\parindent%
3627 \csuse{Xcolalign@#1}%
3628 \parindent=\@tempdima%
3629 \strut%
3630 }%
3631
3632 \newcommand*{\twocolfootgroup}[1]{%
3633   \begingroup%
3634     \csuse{Xbhookgroup@#1}%
3635     \csuse{Xnotefontsize@#1}%
3636     \par%
3637     \splittopskip=\ht\strutbox%
3638     \expandafter%
3639     \Xrigidbalance\csname #1footins\endcsname \tw@ \splittopskip%
3640   \endgroup%
3641 }%
3642
3643 %

```

`\mptwocolfootsetup` The versions for minipages.

`\mptwocolfootgroup`

```

3644 \newcommand*{\mptwocolfootsetup}[1]{%
3645   \count\csname mp#1footins\endcsname 500
3646   \multiply\dimen\csname mp#1footins\endcsname \tw@}
3647 %
3648
3649 \newcommand*{\mptwocolfootgroup}[1]{%
3650   \vskip\skip\@nameuse{mp#1footins}
3651   \ifl@dpairing\ifparledgroup%
3652     \leavevmode\marks\parledgroup@{begin}%
3653     \marks\parledgroup@series{#1}%
3654     \marks\parledgroup@type{Xfootnote}%
3655   \fi\fi\normalcolor
3656   \ifparledgroup%
3657     \ifl@dpairing%
3658       \else%
3659         \setXnoteswidthliketwocolumns@{#1}%
3660         \setXnotespositionliketwocolumns@{#1}%
3661         \print@Xfootnoterule{#1}%
3662       \fi%
3663     \else%
3664       \setXnoteswidthliketwocolumns@{#1}%
3665       \setXnotespositionliketwocolumns@{#1}%
3666       \print@Xfootnoterule{#1}%
3667     \fi%
3668     \csuse{Xbhookgroup@#1}\par%
3669     \splittopskip=\ht\strutbox
3670     \expandafter
3671     \Xrigidbalance\csname mp#1footins\endcsname \tw@ \splittopskip}

```

3672 %

## XIII.7 Footnote hanging indent

`\Xsethangindent` `\Xsethangindent` sets the hangindent for a critical footnote, while `\sethangindentX` does it for a familiar footnote.

```

3673 \newcommand{\Xsethangindent}[1]{%
3674   \if@RTL%
3675     \hangindent=-\csuse{Xhangindent@#1}%
3676     \everypar{\hangindent=-\csuse{Xhangindent@#1}}%
3677   \else%
3678     \hangindent=\csuse{Xhangindent@#1}%
3679     \everypar{\hangindent=\csuse{Xhangindent@#1}}%
3680   \fi%
3681 }%
3682 \newcommand{\sethangindentX}[1]{%
3683   \if@RTL%
3684     \hangindent=-\csuse{hangindentX@#1}%
3685     \everypar{\hangindent=-\csuse{hangindentX@#1}}%
3686   \else%
3687     \hangindent=\csuse{hangindentX@#1}%
3688     \everypar{\hangindent=\csuse{hangindentX@#1}}%
3689   \fi%
3690 }%
3691 %

```

## XIII.8 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.8.1 Font tools

`\endashchar` `\fullstop` `\rbracket` The fonts that are used for printing notes might not have the character mapping we expect: for example, the Computer Modern font that contains old-style numerals does not contain an en-dash or square brackets, and its period and comma are in odd locations. To allow use of the standard footnote macros with such fonts, we use the following macros for certain characters.

The `\endashchar` macro is simply an en-dash from the normal font and is immune to changes in the surrounding font. The same goes for the full stop. These two are used in `\printlines`. The right bracket macro is the same again; it crops up in `\normalfootfmt` and the other footnote macros for controlling the format of the footnotes.

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 `\Xlinangeseparator`, `\Xendlinangeseparator`, `\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.

```

3692 \def\endashchar{\textnormal{--}}
3693
3694 \newcommand*{\fullstop}{\textnormal{.}}
3695 \def\Xsublinesep@side{\fullstop}
3696
3697 \newcommand*{\rbracket}{\textnormal{%
3698   \csuse{text}{\csuse{footnote@lang}}{%
3699     \ifluatex%
3700       \ifdefstring{\footnote@luatextextdir}{TRT}{\thinspace[]}{\thinspace
3701     }}%
3702     \else%
3703     \thinspace}%
3704     \fi}%
3705   }}%
3706 }
3707 %

```

### XIII.8.2 Pstart number in footnote

`\printpstart` The `\printpstart` macro prints the pstart number for a note.

```

3708 \newcommand{\printpstart}[0]{%
3709   \ifboolexpr{bool{l@dpairing} or bool{l@dprintingpages} or bool{
3710     l@dprintingcolumns}}{%
3711     \ifledRcol%
3712       \thepstartR%
3713     \else%
3714       \thepstartL%
3715     \fi%
3716   }{%
3717     \thepstart%
3718   }}%
3719 %

```

### XIII.8.3 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).

```

3720 %
3721 \newcommand{\print@lemma}[3]{%
3722   \bgroup%
3723   \nottoggle{Xlemmadisablefontselection@#3}%

```



```

3724     {\select@lemmafont#1|}%
3725     }%
3726     \bgroup%
3727     \csuse{Xlemmafont@#3}%Deprecated
3728     \csuse{Xwraplemma@#3}{#2}%
3729     \egroup%
3730     \egroup%
3731     \iftoggle{nosep@}{%
3732     \hskip\csuse{Xinplaceoflemmaseparator@#3}%
3733     \relax%
3734     }%
3735     {\ifcsemtyp{Xlemmaseparator@#3}%
3736     {%
3737     \hskip\csuse{Xinplaceoflemmaseparator@#3}%
3738     \relax%
3739     }%
3740     {%
3741     \nobreak%
3742     \hskip\csuse{Xbeforelemmaseparator@#3}%
3743     \csuse{Xlemmaseparator@#3}%
3744     \hskip\csuse{Xafterlemmaseparator@#3}%
3745     \relax%
3746     }%
3747     }%
3748     }%
3749     %

```

#### XIII.8.4 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.

```

3750 \newcommand{\Xstorelineinfo}[2]{%
3751     \l@dp@rsefootspec#1|}%
3752     \iftoggle{Xnumberonlyfirstintwolines@#2}{%
3753     \xdef\lineinfo@{\l@dparsedstartline - \l@dparsedstartsub - \
3754     \l@dparsedendline - \l@dparsedendsub}%
3755     }%
3756     {%
3757     \xdef\lineinfo@{\l@dparsedstartline - \l@dparsedstartsub}%
3758     }%
3759     }%

```

**\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`.

```

3760 \newcommand{\printlinefootnote}[2]{%
3761   \iftoggle{nonum@}{%Try if the line number must printed for this specific
not (by default, yes)
3762   \hspace{\csuse{Xinplaceofnumber@#2}}%
3763   }%
3764   {%
3765     {%
3766       \iftoggle{Xnonumber@#2}{%Try if the line number must printed (by
default, yes)
3767       {%
3768         \hspace{\csuse{Xinplaceofnumber@#2}}%
3769         }%
3770         {%
3771           {\iftoggle{Xnumberonlyfirstinline@#2}% If for this series the
line number must be printed only in the first time.
3772             {%
3773               \ifcsdef{prevline#2}%
3774               {%Be sure the \prevline exists.
3775               \ifcsequal{prevline#2}{\lineinfo@}%Try it
3776               {%
3777                 \ifcseempty{Xsymlinenum@#2}%
3778                 {%
3779                   \hspace{\csuse{Xinplaceofnumber@#2}}%
3780                   }%
3781                   {\printsymlinefootnotearea{#2}}%
3782                   }%
3783                   {%
3784                     \printlinefootnotearea{#1}{#2}%
3785                     }%
3786                   }%
3787                   {%
3788                     \printlinefootnotearea{#1}{#2}%
3789                     }%
3790                   }%
3791                   {%
3792                     \printlinefootnotearea{#1}{#2}%
3793                     }%
3794                     \csxdef{prevline#2}{\lineinfo@}%
3795                   }%
3796                 }%
3797               }%
3798             }%
3799           }
3800         }%

```

`\printsymlinefootnotearea` This macro prints the space before the line symbol, changes the font, when prints the line symbol and the space after it.

```

3801 \newcommand{\printsymlinefootnotearea}[1]{%
3802   \hspace{\csuse{Xbeforexymlineenum@#1}}%
3803   \csuse{Xnotenumfont@#1}%
3804   \ifdimequal{\csuse{Xboxxymlineenum@#1}}{\z@}%
3805     {\csuse{Xsymlineenum@#1}}%
3806     {\hbox to \csuse{Xboxxymlineenum@#1}%
3807       {\csuse{Xsymlineenum@#1}\hfill}}%
3808   }%
3809   \hspace{\csuse{Xaftersymlineenum@#1}}%
3810 }%
3811 %

```

**\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.)

```

3812 \newcommand{\printlinefootnotearea}[2]{%
3813   \printXbeforenumber{#2}%
3814   \csuse{Xnotenumfont@#2}%
3815   \boxfootnotenumbers{#1}{#2}%
3816   \printXafternumber{#2}%
3817 }%
3818 %

```

**\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.

```

3819 \newcommand{\boxfootnotenumbers}[2]{%
3820   \ifdimequal{\csuse{Xboxlinenum@#2}}{\Opt}{%
3821     \printlinefootnotenumbers{#1}{#2}%
3822   }%
3823   {%
3824     \hbox to \csuse{Xboxlinenum@#2}%
3825     {%
3826       \IfSubStr{RC}{\csuse{Xboxlinenumalign@#2}}{\hfill}{}%
3827       \printlinefootnotenumbers{#1}{#2}%
3828       \IfSubStr{LC}{\csuse{Xboxlinenumalign@#2}}{\hfill}{}%
3829     }%
3830   }%
3831 }%
3832 %

```

**\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.

```

3833 \newcommand{\printlinefootnotenumbers}[2]{%

```

```

3834 \xdef\@currentseries{#2}%
3835 \ifboolexpr{%
3836   (togl{Xpstart@#2} and bool{numberpstart})%
3837   or togl{Xpstarteverytime@#2}}%
3838   {\printpstart}{}%
3839 \iftoggle{Xstanza@#2}{%
3840   \ifnumberstanza%
3841     \printstanza%
3842     \csuse{Xstanzaseparator@#2}%
3843   \fi%
3844 }{%
3845 \iftoggle{Xonlypstart@#2}{}%
3846   \csuse{Xtxtbeforenumber@#2}%
3847   \printlines#1|\ifledRcol@{\@Rlineflag\fi}|}%
3848 }%
3849 %

```

**\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.)

```

3850 \newcommand{\printXbeforenumber}[1]{%
3851   \hspace{\csuse{Xbeforenumber@#1}}%
3852 }%
3853 %

```

**\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

```

3854 \newcommand{\printXafternumber}[1]{%
3855   \iftoggle{Xnonbreakableafternumber@#1}{\nobreak}{}%
3856   \hspace{\csuse{Xafternumber@#1}}%
3857 }%
3858 %

```

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. 109: 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@pnum59 \newif\ifl@d@pnum
\ifl@d@ssub60 \newif\ifl@d@ssub
\ifl@d@elin61 \newif\ifl@d@elin
\ifl@d@esl62 \newif\ifl@d@esl
\ifl@d@dash63 \newif\ifl@d@dash
\ifl@d@Xtwolines3864 \newif\ifl@d@Xtwolines%
\ifl@d@Xmorethantwolines3865 \newif\ifl@d@Xmorethantwolines%
\@annot@start@print3866 \let\@annot@start@print\relax%
\@annot@end@print3867 \let\@annot@end@print\relax%
3868 %

\l@dp@rsefootsspec \l@dp@rsefootsspec{<spec>}{<lemma>}{<text>} parses a footnote specification. <lemma>
\l@dp@rsefootsspec and <text> are the lemma and text respectively. <spec> is the line and page num-
\l@dp@rsefootsspec ber and lemma font specifier in \l@d@n@ums style format. The real work is done by
\l@dp@rsefootsspec \l@dp@rsefootsspec which defines macros holding the numeric values. In many cases,
\l@dp@rsefootsspec this last command is called directly. Just a reminder of the arguments:
\l@dp@rsefootsspec \printlines #1 | #2 | #3 | #4 | #5 | #6 | #7
\l@dp@rsefootsspec \printlines start-page | line | subline | end-page | line | subline | fontflag
\l@dp@rsefootsspec \l@dp@rsefootsspec
3869 \newcommand*{\l@dp@rsefootsspec}[3]{\l@dp@rsefootsspec#1|}
3870 \def\l@dp@rsefootsspec#1|#2|#3|#4|#5|#6|#7|{%
3871 \gdef\l@dp@rsefootsspec#1|#2|#3|#4|#5|#6|#7|{%
3872 \gdef\l@dp@rsefootsspec#1|#2|#3|#4|#5|#6|#7|{%
3873 \gdef\l@dp@rsefootsspec#1|#2|#3|#4|#5|#6|#7|{%
3874 \gdef\l@dp@rsefootsspec#1|#2|#3|#4|#5|#6|#7|{%
3875 \gdef\l@dp@rsefootsspec#1|#2|#3|#4|#5|#6|#7|{%
3876 \gdef\l@dp@rsefootsspec#1|#2|#3|#4|#5|#6|#7|{%
3877 }
3878 %

Initialise the several number value macros.

3879 \def\l@dp@rsefootsspec{0}%
3880 \def\l@dp@rsefootsspec{0}%
3881 \def\l@dp@rsefootsspec{0}%
3882 \def\l@dp@rsefootsspec{0}%
3883 \def\l@dp@rsefootsspec{0}%
3884 \def\l@dp@rsefootsspec{0}%
3885 %
3886 %

```

`\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`.

```

3887 \newcommand*{\setprintlines}[6]{%
3888   \let\@annot@start@print\relax%
3889   \let\@annot@end@print\relax%
3890   \l@dpnumfalse%
3891   \l@ddashfalse%
3892   \l@d@elinfalse%
3893   %

```

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

```

3894   \ifbypage@
3895     \ifnum#4=#1 \else
3896       \l@dpnumtrue
3897       \l@ddashtrue
3898     \fi
3899   \fi
3900   %

```

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.

```

3901   \ifboolexpr{%
3902     bool{\l@dpnum}%
3903     or not test{\ifnumequal{#2}{#5}}%
3904   }{%
3905     \l@d@elintrue%
3906     \l@ddashtrue%
3907     \unless\ifx\relax\@annot@end%
3908       \def\@annot@end@print{%
3909         \l@wrapcs@ifnotemptybox{Xwraplinenumannotation@\@currentseries}{\
annot@end}%
3910       }%
3911     \fi%
3912   }{%
3913     %

```

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.

```

3914   \ifl@d@elin%
3915     \def\@annot@start@print{%
3916       \l@wrapcs@ifnotemptybox{Xwraplinenumannotation@\@currentseries}{\
annot@start}%
3917     }%
3918   \else%

```

```

3919 \ifx\annot@start\annot@end%
3920 \unless\ifx\@annot@start\relax%
3921 \def\@annot@start@print{%
3922 \l@wrapcs@ifnotemptybox{Xwraplinenumannotation@\@currentseries}{\
annot@start}%
3923 }%
3924 \fi%
3925 \else%
3926 \ifx\@annot@end@print\relax%
3927 \def\@annot@start@print{%
3928 \l@wrapcs@ifnotemptybox{Xwraplinenumannotation@\@currentseries}{%
3929 \ifx\annot@start\empty\else%
3930 \annot@start%
3931 \ifdefined\linrangesep%
3932 \linrangesep%
3933 \else%
3934 \csuse{Xlinrangeseparator@\@currentseries}%
3935 \fi%
3936 \fi%
3937 \annot@end%
3938 }%
3939 }%
3940 \else%
3941 \let\@annot@start@print\@annot@end@print%
3942 \let\@annot@end@print\relax%
3943 \fi%
3944 \fi%
3945 \fi%
3946 %

```

We print the starting sub-line if it is nonzero.

```

3947 \l@d@ssubfalse
3948 \ifnum#3=0 \else
3949 \l@d@ssubtrue
3950 \fi
3951 %

```

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.

```

3952 \l@d@eslfalse
3953 \ifnum#6=0 \else
3954 \ifnum#6=#3
3955 \ifl@d@elin \l@d@esltrue \else \l@d@eslfalse \fi
3956 \else
3957 \l@d@esltrue
3958 \l@d@dashtrue
3959 \fi
3960 \fi%
3961 %

```

However, if the `\Xtwolines` is set for the current series, we do not print the last line number.

```

3962 \ifl@d@dash%
3963 \ifbool{expr{togl{fulllines@} or test{\ifcempty{Xtwolines@}\
@currentseries}}}%
3964 {}%
3965 {%
3966 \setistwofollowinglines{#1}{#2}{#4}{#5}%
3967 \ifbool{expr{%
3968 (%
3969 togl {Xtwolinesbutnotmore@\@currentseries}%
3970 and not%
3971 (%
3972 bool {istwofollowinglines@}%
3973 )%
3974 }%
3975 or%
3976 (%
3977 (not test{\ifnumequal{#1}{#4}})%
3978 and togl{Xtwolinesonlyinsamepage@\@currentseries}%
3979 )%
3980 }%
3981 {}%
3982 {%
3983 \l@d@d@dashfalse%
3984 \l@d@Xtwolinestrue%
3985 \l@d@elinfalse%
3986 \l@d@eslfalse%
3987 \ifcempty{Xmorethantwolines@\@currentseries}%
3988 {}%
3989 {\ifistwofollowinglines@\else%
3990 \l@d@Xmorethantwolinestrue%
3991 \fi%
3992 }%
3993 }%
3994 }%
3995 \fi%
3996 %

```

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

```

3997 \iftoggle{Xnoidenticallinenumannotation@\@currentseries}{%
3998 \ifx\annot@start\annot@end%
3999 \let\annot@end@print\relax%
4000 \ifx\linenumrep@gobble%Don't print the dash if we're not printing
the line number
4001 \l@d@d@dashfalse%
4002 \fi%
4003 \fi%

```



```

4004 }{}%
4005 %

```

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.

```

4006 \iftoggle{Xlinenumannotationonlyfirst@{\currentseries}{%
4007   \ifbool{expr{%
4008     (%
4009     togl{Xlinenumannotationonlyfirstintwo@{\currentseries}%
4010     and test{\ifcsequal{annot@start}{prevannot@start@{\currentseries}}}%
4011     and test{\ifcsequal{annot@end}{prevannot@end@{\currentseries}}}%
4012   )%
4013   or%
4014   (%
4015     not togl{Xlinenumannotationonlyfirstintwo@{\currentseries}%
4016     test{\ifcsequal{annot@start}{prevannot@start@{\currentseries}}}%
4017   )%
4018 }{}%
4019   \def\@annot@start@print{%
4020     \l@wrap{cs@ifnotemptybox{Xwraplinenumannotation@{\currentseries}}{\
csuse{Xsymlinenumannotation@{\currentseries}}}%
4021   }%
4022   \let\@annot@end@print\relax%
4023   \ifx\linenumrep\@gobble%Don't print the dash if we're not printing
the line number
4024     \l@d@dashfalse%
4025     \fi%
4026 }{}%
4027 \global\cslet{prevannot@start@{\currentseries}}{\annot@start}%
4028 \global\cslet{prevannot@end@{\currentseries}}{\annot@end}%
4029 }%
4030 {}%
4031 %

```

End of `\setprintlines`.

```

4032 }%
4033 %

```

`\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.

```

4034 \newif\ifistwofollowinglines@%
4035 \newcommand{\setistwofollowinglines}[4]{%
4036   \ifcsdef{lastlinenumberon@#1}%
4037     {\numdef{\tmp}{\csuse{lastlinenumberon@#1}}}%
4038     {\numdef{\tmp}{0}}%
4039   \istwofollowinglines@false%
4040   \ifnumequal{#4-#2}{1}%
4041     {\istwofollowinglines@true}%
4042   {\ifbypage@%
4043     \ifnumequal{#3-#1}{1}%
4044     {%
4045       \ifnumequal{#2}{\tmp}%
4046       {\ifnumequal{#4}{1}{\istwofollowinglines@true}{}}%
4047       {}%
4048     }%
4049     {}%
4050   \fi%
4051 }%
4052 }%
4053 %

```

`\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

```

4054 \def\printlines#1|#2|#3|#4|#5|#6|#7|#8|{%
4055   \begingroup%
4056   %

```

If we use Lua<sub>T</sub><sub>E</sub>X, ensure we use good text's direction.

```

4057   \ifluatex%
4058     \edef\@tmp{\the\textdir}%
4059     \ifdefstring{\@tmp}{TLT}{\textdir TLT}%Test in order to prevent
spurious space (bug #397)
4060   \fi%
4061 %

```

Decide which part of line number components we will print.

```

4062 \setprintlines{#1}{#2}{#3}{#4}{#5}{#6}%
4063 %

```

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.

```

4064 \ifdimequal{\csuse{Xboxstartlinenum@\@currentseries}}{0pt}%
4065 {\bgroup}%
4066 {\leavevmode\hbox to \csuse{Xboxstartlinenum@\@currentseries}\bgroup\
hfill}%
4067 \ifcsstring{Xlinenumannotationposition@\@currentseries}{before}%
4068 {\@cannot@start@print}%
4069 {}%
4070 \ifl@d@pnum%
4071 \wrap@edcrossref{\@this@crossref@start}{#1}%
4072 \csuse{Xpagelinesep@\@currentseries}%
4073 \fi%
4074 \wrap@edcrossref{\@this@crossref@start}{%
4075 \linenumrep{#2}%
4076 \iftoggle{Xlineflag@\@currentseries}{#8}{}}%
4077 }%
4078 \ifl@d@ssub%
4079 \csuse{Xsublinesep@\@currentseries}%
4080 \wrap@edcrossref{\@this@crossref@start}{\sublinenumrep{#3}}%
4081 \fi
4082 \ifcsstring{Xlinenumannotationposition@\@currentseries}{after}%
4083 {\@cannot@start@print}%
4084 {}%
4085 \egroup%
4086 %

```

Then print the dash + end line number, or the range symbol.

```

4087 \ifdimequal{\csuse{Xboxendlinenum@\@currentseries}}{0pt}%
4088 {\bgroup}%
4089 {\hbox to \csuse{Xboxendlinenum@\@currentseries}\bgroup}%
4090 \ifl@d@Xtwolines%
4091 \ifl@d@Xmorethantwolines%
4092 \csuse{Xmorethantwolines@\@currentseries}%
4093 \else%
4094 \csuse{Xtwolines@\@currentseries}%
4095 \fi%
4096 \else%
4097 \ifl@d@dash%
4098 \ifdefined\linerangesep%
4099 \linerangesep%
4100 \else%
4101 \csuse{Xlinerangeseparator@\@currentseries}%
4102 \fi%

```

```

4103 \fi%
4104 \ifcsstring{Xlinenumannotationposition@ \@currentseries}{before}%
4105 {\@annot@end@print}%
4106 }%
4107 \ifl@d@pnum%
4108 \wrap@edcrossref{\@this@crossref@end}{#4}%
4109 \csuse{Xpagelinesep@ \@currentseries}%
4110 \fi%
4111 \ifl@d@elin%
4112 \wrap@edcrossref{\@this@crossref@end}{%
4113 \linenumrep{#5}%
4114 \iftoggle{Xlineflag@ \@currentseries}{#8}{}%
4115 }%
4116 \fi%
4117 \ifl@d@esl%
4118 \ifl@d@elin%
4119 \csuse{Xsublinesep@ \@currentseries}%
4120 \fi%
4121 \wrap@edcrossref{\@this@crossref@end}{\sublinenumrep{#6}}%
4122 \fi%
4123 \ifcsstring{Xlinenumannotationposition@ \@currentseries}{after}%
4124 {\@annot@end@print}%
4125 }%
4126 \fi%
4127 \ifdimequal{\csuse{Xboxendlinenum@ \@currentseries}}{0pt}%
4128 {}%
4129 {\hfill}%Prevent underfull hbox
4130 \egroup%
4131 \endgroup%
4132 }%
4133 %

```

### XIII.8.5 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

```

4134 \newcommand{\prepare@Xgroupbyline}[3]{%
4135 \iftoggle{Xgroupbylineseparetwolines@#1}{%
4136 \l@dparsfootspec#2%
4137 \ifcsdef{#1@forinserting@ \l@dparseendpage- \l@dparseendline- \l@dparseendsub}%
4138 {%
4139 \csgappto{#1@forinserting@ \l@dparseendpage- \l@dparseendline- \l@dparseendsub}{%
4140 \hskip\csuse{Xafternote@#1}\relax%

```

```

4141     }%
4142   }%
4143   {}%
4144   \add@hooktoggle@specific@to@cs{#1@forinserting@l@dparsedendpage-\
l@dparsedendline-\l@dparsedendsub}%
4145   \add@hookarg@specific@to@cs{#1@forinserting@l@dparsedendpage-\
l@dparsedendline-\l@dparsedendsub}%
4146   \csxappto%
4147   {#1@forinserting@l@dparsedendpage-\l@dparsedendline-\l@dparsedendsub
}%
4148   {%
4149     \keep@this@crossref@forinserting%
4150     \unexpanded{%
4151       \ifcempty{Xsymlinenum@#1}%
4152         {\csuse{Xparafootsep@#1}}%
4153         {}%
4154         #3{#1}{#2}%
4155       }%
4156     }%
4157     \listcsxadd{#1@forinserting}{\l@dparsedendpage-\l@dparsedendline-\
l@dparsedendsub}%
4158   }{%
4159     \ifcsdef{#1@forinserting@all}{%
4160       \csgappto%
4161       {#1@forinserting@all}%
4162       {\hskip\csuse{Xafternote@#1}\relax}%
4163     }{%
4164       \add@hooktoggle@specific@to@cs{#1@forinserting@all}%
4165       \add@hookarg@specific@to@cs{#1@forinserting@all}%
4166       \csxappto%
4167       {#1@forinserting@all}%
4168     {%
4169       \keep@this@crossref@forinserting%
4170       \unexpanded{%
4171         \ifcempty{Xsymlinenum@#1}%
4172           {\csuse{Xparafootsep@#1}}%
4173           {}%
4174           #3{#1}{#2}%
4175         }%
4176       }%
4177     }%
4178     \listcsgadd{#1@forinserting}{all}%
4179   }%
4180   %

```

`\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.

```

4181 \newcommand{\keep@this@crossref@forinserting}{%

```

```

4182 \unexpanded{\gdef\@this@crossref@start}{\@this@crossref@start}%
4183 \unexpanded{\gdef\@this@crossref@end}{\@this@crossref@end}%
4184 }%
4185 %

```

## XIV Familiar footnotes

### XIV.1 Adjacent footnotes

The original edmac provided users with five series of critical footnotes (`\Afootnote` `\Bfootnote` `\Cfootnote` `\Dfootnote` `\Efootnote`), and  $\TeX$  provides a single numbered footnote. The `reledmac` package uses the edmac mechanism to provide six series of numbered footnotes.

First, though, the `footmisc` package has an option whereby two or more consecutive `\footnotes` have their marks separated by commas. This seemed to Peter Wilson such a useful ability that it was provided automatically by `eledmac`.

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`.

```

4186 \providecommand*\multiplefootnotemarker{\3sp}
4187 \providecommand*\multfootsep{\textsuperscript{\normalfont,}}
4188
4189 %

```

`\m@mmf@prepare` A pair of self-cancelling kerns. This may have been defined in the `memoir` class.

```

4190 \providecommand*\m@mmf@prepare{%
4191   \kern-\multiplefootnotemarker
4192   \kern\multiplefootnotemarker\relax}
4193 %

```

`\m@mmf@check` This may have been defined in the `memoir` class. If it recognises the last kern as `\multiplefootnotemarker` it typesets `\multfootsep`.

```

4194 \providecommand*\m@mmf@check{%
4195   \ifdim\lastkern=\multiplefootnotemarker\relax
4196     \edef\@x@sf{\the\spacefactor}%
4197     \unkern
4198     \multfootsep
4199     \spacefactor\@x@sf\relax
4200   \fi}
4201
4202 %

```

We have to modify `\@footnotetext` and `\@footnotemark`. However, if memoir is used the modifications have already been made.

```
4203 \@ifclassloaded{memoir}{\%
4204 %
```

`\@footnotetext` Add `\m@mmf@prepare` at the end of `\@footnotetext`.

```
4205 \apptocmd{\@footnotetext}{\m@mmf@prepare}{\%}{\%
4206 %
```

`\@footnotemark` Modify `\@footnotemark` to cater for adjacent `\footnotes`.

```
4207
4208 \patchcmd{\@footnotemark}
4209   {\nobreak}
4210   {\m@mmf@check
4211    \nobreak
4212   }
4213   {}{}
4214 \patchcmd{\@footnotemark}
4215   {\@makefnmark}
4216   {\@makefnmark
4217    \m@mmf@prepare
4218   }
4219   {}{}
4220 %
```

Finished the modifications for the non-memoir case.

```
4221 }
4222
4223 %
```

## 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.

```
4224 \pretocmd{\@footnotetext}{\%
4225   \ifnumberedpar@
4226     \edtext{\l@dbfnote{#1}}}%
4227   \else
4228   }{}{}
4229 \apptocmd{\@footnotetext}{\fi}{\%}{\%
4230 %
```

```

\l@dbfnote \l@dbfnote adds the footnote to the insert list, and \vl@dbfnote calls the original
\vl@dbfnote \@footnotetext. We also patch \footnote in order to get the correct footnote
\vl@dbfnote numbers when typesetting parallel texts. This is moved into a \get@fnmark command.
\footnote
\get@fnmark
\get@thisfootnote
4231
4232 \patchcmd%
4233   {\footnote}%
4234   {\stepcounter\@mpfn}%
4235   {%
4236     \ifboolexpr{bool{l@dpairing} or bool{l@dprintingpages} or bool{
l@dprintingcolumns}}{%
4237       \global\advance\footnote@reading by \@ne%
4238       \get@thisfootnote%
4239       \get@fnmark{\thisc@footnote}%
4240       \ifcsdef{footnotereading\the\footnote@reading=typeset}%
4241         {\setcounter{\@mpfn}{\csuse{footnotereading\the\footnote@reading=
typeset}}}%
4242         {\setcounter{\@mpfn}{\footnote@reading}}%
4243       }{%
4244         \stepcounter\@mpfn%
4245       }%
4246     }%
4247   }%
4248   {}
4249
4250 \newcommand{\get@thisfootnote}{%
4251   \ifboolexpr{bool{l@dpairing} or bool{l@dprintingpages} or bool{
l@dprintingcolumns}}{%
4252     \protected@xdef\thisc@footnote{\the\footnote@reading}%
4253   }{%
4254     \protected@xdef\thisc@footnote{\the\c@footnote}%
4255   }%
4256 }%
4257
4258 \newcommand{\l@dbfnote}[1]{%
4259   \get@thisfootnote%
4260   \gdef\@tag{#1\relax}%
4261   \ifledRcol%
4262     \xright@appenditem{%
4263       \ifdefined\Hy@footnote@currentHref%
4264         \noexpand\def\noexpand\Hy@footnote@currentHref{\
Hy@footnote@currentHref}%
4265       \fi%
4266       \noexpand\vl@dbfnote{\expandonce\@tag}{\thisc@footnote}%
4267     }%
4268     \to\inserts@listR
4269     \global\advance\insert@countR \@ne%
4270   \else%
4271     \xright@appenditem{%

```



```

4272         \ifdefined\Hy@footnote@currentHref%
4273         \noexpand\def\noexpand\Hy@footnote@currentHref{\
Hy@footnote@currentHref}%
4274         \fi%
4275         \noexpand\vl@dbfnote{{\expandonce\@tag}}{\thisc@footnote}%
4276     }%
4277         \to\inserts@list
4278     \global\advance\insert@count \@ne%
4279 \fi
4280 \ignorespaces%
4281 }%
4282
4283 \newcommand{\get@fnmark}[1]{%
4284     \ifboolexpr{bool{l@dpairing} or bool{l@dprintingpages} or bool{
l@dprintingcolumns}}%
4285     {%
4286         \stepcounter{footnote@typeset}%
4287         \setcounter{footnote}{\c@footnote@typeset}%
4288         \immediate\write\@mainaux{%
4289             \csgdef{footnotereading#1=typeset}{\the\c@footnote@typeset}%
4290         }%
4291         \def\@thefnmark{\thefootnote}%
4292     }%
4293     {%
4294         \setcounter{footnote}{#1}%
4295         \def\@thefnmark{\thefootnote}%
4296     }%
4297 }%
4298
4299 \newcommand{\vl@dbfnote}[2]{%
4300     \get@fnmark{#2}%
4301     \@footnotetext{#1}%
4302 }%
4303 %

```

### 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
4304 \newcommand*{\prebodyfootmark}{%
4305     \leavevmode
4306     \ifhmode
4307         \edef\@x@sf{\the\spacefactor}%
4308         \m@mmlf@check
4309         \nobreak
4310     \fi}

```

```

4311 \newcommand{\postbodyfootmark}{%
4312   \m@mmf@prepare
4313   \ifhmode\spacefactor\@x@sf\fi\relax}
4314
4315 %

```

## 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.

```

4316 \newcommand{\arrangementX}[2][1,usedefault]{%
4317   \def\do##1{%
4318     \csname arrangementX@#2\endcsname{##1}%
4319   }%
4320   \ifstrempy{#1}%
4321     {%
4322       \dolistloop{\@series}%
4323     }%
4324     {
4325       \docsvlist{#1}%
4326     }%
4327   }%
4328 %

```

### 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.

```

4329 \newcommand*{\normal@footnotemarkX}[1]{%
4330   \prebodyfootmark
4331   \wrapped@bodyfootmarkX{#1}%
4332   \postbodyfootmark}
4333
4334 %

```

**\normalbodyfootmarkX** The `\normalbodyfootmarkX{series}` *really* typesets the in-text marker. The style is the normal superscript.

```

4335 \newcommand*{\normalbodyfootmarkX}[1]{%
4336   \hbox{\textsuperscript{\normalfont\@nameuse{@thefnmark#1}}}}
4337 %

```

**\normalvfootnoteX** `\normalvfootnoteX{series}{text}` does the `\insert` for the *series* and calls the series' `\footfmt...` to format the *text*.

```

4338 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\normalvfootnoteX}[2]{%
4339   \beforeinsertion@X{#1}%
4340   \insert\@nameuse{footins#1}\bgroup
4341     \fontseries{\seriesdefault}%
4342     \fontshape{\shapedefault}%
4343     \hsize=\expandafter\dimexpr\csuse{widthX@#1}\relax%
4344     \noindent\csuse{bhooknoteX@#1}%
4345     \csuse{notefontsizeX@#1}%
4346     \footplitskips
4347     \ifl@dpairing\ifl@dpaging\else%
4348       \setnoteswidthliketwocolumnsX@{#1}%
4349     \fi\fi%
4350     \setnotesXpositionliketwocolumns@{#1}%
4351     \spaceskip=\z@skip \xspaceskip=\z@skip
4352     \csuse{\csuse{footnote@dir}}\@nameuse{footfmt#1}{#1}{#2}\egroup}
4353 %
4354 %

```

`\mpnormalvfootnoteX` The minipage version.

```

4355 \newcommand*{\mpnormalvfootnoteX}[3]{%
4356   \get@thisfootnoteX{#1}%
4357   \get@fnmarkX{#1}{\thisc@footnote}{#3}%
4358   \ifstrempy{#3}{%
4359     \edef\this@footnoteX@reading{\the\csname footnote#1@reading\endcsname}%
4360   }{%
4361     \edef\this@footnoteX@reading{###3}%
4362   }%
4363   \global\setbox\@nameuse{mpfootins#1}\vbox{%
4364     \unvbox\@nameuse{mpfootins#1}
4365     \noindent\csuse{bhooknoteX@#1}%
4366     \csuse{notefontsizeX@#1}%
4367     \hsize\columnwidth
4368     \@parboxrestore
4369     \color@begingroup
4370     \@nameuse{footfmt#1}{#1}{#2}\color@endgroup}}
4371 %
4372 %

```

`\normalfootfmtX` `\normalfootfmtX{<series>}{<text>}` typesets the footnote text, prepended by the marker.

```

4373 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\normalfootfmtX}[2]{%
4374   \ifluatex%
4375     \textdir\footnote@luatextextdir%
4376     \pardir\footnote@luatexpardir%
4377     \par%
4378   \fi%
4379   \protected@edef\@currentlabel{%
4380     \@nameuse{@thefnmark#1}%
4381   }%

```

```

4382 \ledsetnormalparstuffX{#1}%
4383 \sethangindentX{#1}%
4384 \rule{z@\splittopskip%
4385   {\csuse{notenumfontX@#1}\wrapped@footfootmarkX{#1}}}%
4386   \csuse{wrapcontentX@#1}{#2}%
4387   \strut\par}}
4388
4389 %

```

**\normalfootfootmarkX** `\normalfootfootmarkX{<series>}` is called by `\normalfootfmtX` to typeset the footnote marker in the footer before the footnote text.

```

4390 \newcommand*{\normalfootfootmarkX}[1]{%
4391   \textsuperscript{\@nameuse{@thefnmark#1}}}%
4392
4393 %

```

**\normalfootstartX** `\normalfootstartX{<series>}` is the `<series>` footnote starting macro used in the output routine.

```

4394 \newcommand*{\normalfootstartX}[1]{%
4395   \ifdimequal{Opt}{\prenotesX@}{}%
4396   {%
4397     \iftoggle{prenotesX@}{%
4398       \togglefalse{prenotesX@}%
4399       \skip\csname footins#1\endcsname=%
4400       \glueexpr\csuse{prenotesX@}+\csuse{afterruleX@#1}\relax%
4401     }%
4402   }%
4403 }%
4404 \vskip\skip\csname footins#1\endcsname%
4405 \leftskip=\z@
4406 \rightskip=\z@
4407 \ifl@dpairing\else%
4408   \hsize=\old@hsize%
4409 \fi%
4410 \setnoteswidthliketwocolumnsX@{#1}%
4411 \setnotesXpositionliketwocolumns@{#1}%
4412 \print@footnoteXrule{#1}%
4413 }%
4414
4415 %

```

**\normalfootnoteruleX** The rule drawn before the footnote series group.

```

4416 \let\normalfootnoteruleX=\footnoterule
4417
4418 %

```

`\normalfootgroupX` `\normalfootgroupX{⟨series⟩}` sends the contents of the `⟨series⟩` insert box to the output page without alteration.

```

4419 \newcommand*\normalfootgroupX}[1]{%
4420   \csuse{bhookgroupX@#1}%
4421   \unvbox\@nameuse{footins#1}%
4422   \hsize=\old@hsize%
4423 }%
4424
4425 %

```

`\mpnormalfootgroupX` The minipage version.

```

4426 \newcommand*\mpnormalfootgroupX}[1]{%
4427   \vskip\skip\@nameuse{mpfootins#1}
4428   \ifl@dpairing\ifparledgroup%
4429     \leavevmode\marks\parledgroup@{begin}%
4430     \marks\parledgroup@series{#1}%
4431     \marks\parledgroup@type{footnoteX}%
4432   \fi\fi\normalcolor
4433   \ifparledgroup%
4434     \ifl@dpairing%
4435     \else%
4436       \setnoteswidthliketwocolumnsX@{#1}%
4437       \setnotesXpositionliketwocolumns@{#1}%
4438       \print@footnoteXrule{#1}%
4439     \fi%
4440   \else%
4441     \setnoteswidthliketwocolumnsX@{#1}%
4442     \setnotesXpositionliketwocolumns@{#1}%
4443     \print@footnoteXrule{#1}%
4444   \fi%
4445   \csuse{bhookgroupX@#1}%
4446   \unvbox\@nameuse{mpfootins#1}}
4447
4448 %

```

`\normalbfnoteX` #1 = footnote series, #2 = footnote content, #3 manual footnote number

```

4449
4450 \newcommand{\normalbfnoteX}[3]{%
4451   \get@thisfootnoteX{#1}%
4452   \ifledRcol%
4453     \ifluatex
4454       \footnotelang@lua[R]%
4455     \fi
4456     \@ifundefined{xpg@main@language}%if polyglossia
4457     {}%
4458     {\footnotelang@poly[R]}%
4459   \xright@appenditem{%

```

```

4460 \noexpand\led@set@index@fornote{#1}%
4461 \noexpand\prepare@edindex@fornote{\l@d@nums}%
4462 \unexpanded{\def\this@footnoteX@reading}{\the\csname footnote#1
@reading\endcsname}%
4463 \noexpand\vbfnoteX{#1}{#2}{\thisc@footnote}{#3}%
4464 \noexpand\led@reinit@index@fornote%
4465 \unexpanded{\advance\@edindex@fornote@m@ne}%
4466 }%
4467 \to\inserts@listR
4468 \global\advance\insert@countR \@ne%
4469 \else%
4470 \ifluatex
4471 \footnotelang@lua%
4472 \fi
4473 \ifundefined{xpg@main@language}%if polyglossia
4474 {}%
4475 {\footnotelang@poly}%
4476 \xright@appenditem{%
4477 \noexpand\led@set@index@fornote{#1}%
4478 \noexpand\prepare@edindex@fornote{\l@d@nums}%
4479 \unexpanded{\def\this@footnoteX@reading}{\the\csname footnote#1
@reading\endcsname}%
4480 \noexpand\vbfnoteX{#1}{#2}{\thisc@footnote}{#3}%
4481 \noexpand\led@reinit@index@fornote%
4482 \unexpanded{\advance\@edindex@fornote@m@ne}%
4483 }%
4484 \to\inserts@list
4485 \global\advance\insert@count \@ne%
4486 \fi
4487 \ignorespaces}
4488
4489 %

```

**\get@thisfootnoteX** The macro `\get@thisfootnote` command just saves the footnote number in the `\thisfootnote` macro, depending on the use of pairing environments.

```

4490 \newcommand{\get@thisfootnoteX}[1]{%
4491 \ifbool{bool{l@d@pairing}} or bool{l@d@printingpages} or bool{
l@d@printingcolumns}}{%
4492 \protected@xdef\thisc@footnote{\the\csname footnote#1@reading\
endcsname}%
4493 }{%
4494 \protected@xdef\thisc@footnote{\the\csname c@footnote#1\endcsname}%
4495 }%
4496 }%
4497 %

```

**\vbfnoteX** This command calls the correct footnote-inserting commands. #1 footnote series, #2 footnote content, #3 footnote counter, #4 manual footnote mark

```

4498 \newcommand{\vbfnoteX}[4]{%
4499   \get@fnmarkX{#1}{#3}{#4}\relax%
4500   \@nameuse{regvfootnote#1}{#1}{#2}%
4501 }%
4502 %

```

**\get@fnmarkX** This command gets the correct footnote number when typesetting parallel texts. #1 footnote series, #2 footnote counter, #3 manual footnote number

```

4503 \newcommand{\get@fnmarkX}[3]{%
4504   \ifstrempy{#3}{%
4505     \ifboolexpr{bool{!@dpairing} or bool{!@dprintingpages} or bool{
4506       !@dprintingcolumns}}{%
4507       {%
4508         \stepcounter{footnote#1@typeset}%
4509         \setcounter{footnote#1}{\value{footnote#1@typeset}}%
4510         \@namedef{@thefnmark#1}{\csuse{thefootnote#1}}%
4511         \immediate\write\@mainaux{%
4512           \csgdef{footnote#1reading#2=typeset}{\the\csname c@footnote#1
4513             @typeset\endcsname}%
4514         }%
4515       }%
4516       {%
4517         \setcounter{footnote#1}{#2}%
4518         \@namedef{@thefnmark#1}{\csuse{thefootnote#1}}%
4519       }%
4520     }%
4521   }%
4522 }%
4523 %
4524 %

```

**\vnumfootnoteX** #1 footnote series, #2 footnote content, #3 manual footnote mark

```

4525 \newcommand{\vnumfootnoteX}[3]{%
4526   \ifnumberedpar@
4527     \edtext{}{\normalbfnoteX{#1}{#2}{#3}}%
4528   \else
4529     \def\this@footnoteX@reading{\the\csname footnote#1@reading\endcsname}%
4530     \get@thisfootnoteX{#1}%
4531     \get@fnmarkX{#1}{\expandonce\thisc@footnote}{#3}%
4532     \@nameuse{regvfootnote#1}{#1}{#2}%
4533   \fi}
4534 %
4535 %

```

**arrangementX@normal** `\arrangementX@normal{<series>}` initialises the settings for the <series> footnotes. This should always be called for each series.

```

4536 \newcommand*{\arrangementX@normal}[1]{%
4537   \csgdef{series@displayX#1}{normal}
4538   \expandafter\let\csname footstart#1\endcsname=\normalfootstartX
4539   \@namedef{@footnotemark#1}{\normal@footnotemarkX{#1}}
4540   \@namedef{bodyfootmark#1}{\normalbodyfootmarkX{#1}}
4541   \expandafter\let\csname regvfootnote#1\endcsname=\normalvfootnoteX
4542   \expandafter\let\csname vfootnote#1\endcsname=\vnumfootnoteX
4543   \expandafter\let\csname footfmt#1\endcsname=\normalfootfmtX
4544   \@namedef{footfootmark#1}{\normalfootfootmarkX{#1}}
4545   \expandafter\let\csname footgroup#1\endcsname=\normalfootgroupX
4546   \expandafter\let\csname footnoterule#1\endcsname=\normalfootnoteruleX
4547   \count\csname footins#1\endcsname=1000
4548   \csxdef{default@footins#1}{1000}%Use to have note only for one side
4549   \dimen\csname footins#1\endcsname=\csuse{maxhnotesX@#1}
4550   \skip\csname footins#1\endcsname=\csuse{beforenotesX@#1}%
4551   \advance\skip\csname footins#1\endcsname by\csuse{afterruleX@#1}%
4552   %

```

Additions for minipages.

```

4553   \ifnoledgroup@else%
4554     \expandafter\let\csname mpvfootnote#1\endcsname=\mpnormalvfootnoteX
4555     \expandafter\let\csname mpfootgroup#1\endcsname=\mpnormalfootgroupX
4556     \count\csname mpfootins#1\endcsname=1000
4557     \dimen\csname mpfootins#1\endcsname=\csuse{maxhnotesX@#1}
4558     \skip\csname mpfootins#1\endcsname=\csuse{beforenotesX@#1}%
4559     \advance\skip\csname mpfootins#1\endcsname by\csuse{afterruleX@#1}%
4560   \fi
4561 }
4562
4563 %

```

#### 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 4564 \newcommand*{\arrangementX@twocol}[1]{%
4565   \csgdef{series@displayX#1}{twocol}
4566   \expandafter\let\csname regvfootnote#1\endcsname=\twocolvfootnoteX
4567   \expandafter\let\csname footfmt#1\endcsname=\twocolfootfmtX
4568   \expandafter\let\csname footgroup#1\endcsname=\twocolfootgroupX
4569   \dimen\csname footins#1\endcsname=\csuse{maxhnotesX@#1}%
4570   \skip\csname footins#1\endcsname=\csuse{beforenotesX@#1}%
4571   \advance\skip\csname footins#1\endcsname by \csuse{afterruleX@#1}\relax%
4572   \twocolfootsetupX{#1}
4573   \ifnoledgroup@else%
4574     \expandafter\let\csname mpvfootnote#1\endcsname=\mpnormalvfootnoteX
4575     \expandafter\let\csname mpfootgroup#1\endcsname=\mptwocolfootgroupX
4576     \skip\csname mpfootins#1\endcsname=\csuse{beforenotesX@#1}%

```



```

4577 \advance\skip\csname mpfootins#1\endcsname by\csuse{afterruleX@#1}
4578 \mptwocolfootsetupX{#1}
4579 \fi%
4580 }
4581
4582 %

```

`\twocolfootsetupX` `\twocolfootsetupX{<series>}`

`\mptwocolfootsetupX`

```

4583 \newcommand*{\twocolfootsetupX}[1]{%
4584 \count\csname footins#1\endcsname 500
4585 \csxdef{default@footins#1}{500}%Use this to confine the notes to one
side only
4586 \multiply\dimen\csname footins#1\endcsname by \tw@}
4587 \newcommand*{\mptwocolfootsetupX}[1]{%
4588 \count\csname mpfootins#1\endcsname 500
4589 \multiply\dimen\csname mpfootins#1\endcsname by \tw@}
4590
4591 %

```

`\twocolvfootnoteX` `\twocolvfootnoteX{<series>}`

```

4592 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\twocolvfootnoteX}[2]{%
4593 \beforeinsertion@X{#1}%
4594 \insert\csname footins#1\endcsname\bgroup%
4595 \hsize=\expandafter\dimexpr\csuse{widthX@#1}\relax%
4596 \noindent\csuse{bhooknoteX@#1}%
4597 \csuse{notefontsizeX@#1}%
4598 \footsplitskips%
4599 \spaceskip=\z@skip \xspaceskip=\z@skip%
4600 \@nameuse{footfmt#1}{#1}{#2}\egroup}
4601
4602 %

```

`\twocolfootfmtX` `\twocolfootfmtX{<series>}`

```

4603 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\twocolfootfmtX}[2]{%
4604 \protected@edef\@currentlabel{%
4605 \@nameuse{@thefnmark#1}%
4606 }%
4607 \normal@pars%
4608 \sethangindentX{#1}%
4609 \hsize \csuse{hsizetwocolX@#1}%
4610 \nottoggle{parindentX@#1}{\parindent=\z@}{}%
4611 \tolerance=5000\relax%
4612 \par%
4613 \@tempdima=\parindent%
4614 \csuse{colalignX@#1}%
4615 \parindent=\@tempdima%
4616 {\hspace{\parindent}%

```

```

4617 \csuse{notenumfontX@#1}\wrapped@footfootmarkX{#1}\strut%
4618 \csuse{wrapcontentX@#1}{#2}%
4619 \strut\par}%
4620 \allowbreak%
4621 }%
4622 %
4623 %

\twocolfootgroupX \twocolfootgroupX{<series>}
\mptwocolfootgroupX
4624 \newcommand*{\twocolfootgroupX}[1]{\csuse{bhookgroupX@#1}\csuse{
notefontsizeX@#1}
4625 \splittopskip=\ht\strutbox
4626 \expandafter
4627 \rigidbalanceX\csname footins#1\endcsname \tw@ \splittopskip}}
4628 %
4629 \newcommand*{\mptwocolfootgroupX}[1]{\%
4630 \vskip\skip\@nameuse{mpfootins#1}
4631 \ifl@dpairing\ifparledgroup%
4632 \leavevmode\marks\parledgroup@{begin}%
4633 \marks\parledgroup@series{#1}%
4634 \marks\parledgroup@type{footnoteX}%
4635 \fi\fi\normalcolor
4636 \ifparledgroup%
4637 \ifl@dpairing%
4638 \else%
4639 \setnoteswidthliketwocolumnsX@{#1}%
4640 \setnotesXpositionliketwocolumns@{#1}%
4641 \print@footnoteXrule{#1}%
4642 \fi%
4643 \else%
4644 \setnoteswidthliketwocolumnsX@{#1}%
4645 \setnotesXpositionliketwocolumns@{#1}%
4646 \print@footnoteXrule{#1}%
4647 \fi%
4648 \csuse{bhookgroupX@#1}%
4649 \splittopskip=\ht\strutbox
4650 \expandafter
4651 \rigidbalanceX\csname mpfootins#1\endcsname \tw@ \splittopskip}}
4652 %
4653 %

```

#### 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
4654 \newcommand*{\arrangementX@threecol}[1]{%
4655 \csgdef{series@displayX#1}{threecol}

```

```

4656 \expandafter\let\csname regvfootnote#1\endcsname=\threecolvfootnoteX
4657 \expandafter\let\csname footfmt#1\endcsname=\threecolfootfmtX
4658 \expandafter\let\csname footgroup#1\endcsname=\threecolfootgroupX
4659 \dimen\csname footins#1\endcsname=\csuse{maxhnotesX@#1}%
4660 \skip\csname footins#1\endcsname=\csuse{beforenotesX@#1}%
4661 \advance\skip\csname footins#1\endcsname by \csuse{afterruleX@#1}\relax%
4662 \threecolfootsetupX{#1}
4663 \ifnolddgroup@else%
4664   \expandafter\let\csname mpvfootnote#1\endcsname=\mpnormalvfootnoteX
4665   \expandafter\let\csname mpfootgroup#1\endcsname=\mpthreecolfootgroupX
4666   \skip\csname mpfootins#1\endcsname=\csuse{beforenotesX@#1}%
4667   \advance\skip\csname mpfootins#1\endcsname by \csuse{afterruleX@#1}
4668   \mpthreecolfootsetupX{#1}
4669 \fi%
4670 }
4671
4672 %

```

`\threecolfootsetupX`    `\threecolfootsetupX{<series>}`  
`\mpthreecolfootsetupX`

```

4673 \newcommand*{\threecolfootsetupX}[1]{%
4674   \count\csname footins#1\endcsname 333
4675   \csxdef{default@footins#1}{333}%Use this to confine the notes to one
side only
4676   \multiply\dimen\csname footins#1\endcsname by \thr@@
4677   \newcommand*{\mpthreecolfootsetupX}[1]{%
4678     \count\csname mpfootins#1\endcsname 333
4679     \multiply\dimen\csname mpfootins#1\endcsname by \thr@@}
4680
4681 %

```

`\threecolvfootnoteX`    `\threecolvfootnoteX{<series>}{<text>}`

```

4682 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\threecolvfootnoteX}[2]{
%
4683   \beforeinsertion@X{#1}%
4684   \insert\csname footins#1\endcsname\bgroup%
4685   \hsize=\expandafter\dimexpr\csuse{widthX@#1}\relax%
4686   \noindent\csuse{bhooknoteX@#1}%
4687   \csuse{notefontsizeX@#1}%
4688   \footsplitskips%
4689   \@nameuse{footfmt#1}{#1}{#2}\egroup}
4690
4691 %

```

`\threecolfootfmtX`    `\threecolfootfmtX{<series>}`

```

4692 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\threecolfootfmtX}[2]{%
4693   \protected@edef\@currentlabel{%
4694     \@nameuse{@thefnmark#1}%

```

```

4695 }%
4696 \sethangindentX{#1}%
4697 \normal@pars%
4698 \hsize \csuse{hsizethreecolX@#1}%
4699 \nottoggle{parindentX@#1}{\parindent=\z@}{}%
4700 \tolerance=5000\relax%
4701 \@tempdima=\parindent%
4702 \csuse{colalignX@#1}%
4703 \parindent=\@tempdima%
4704 {\hspace{\parindent}%
4705 \csuse{notenumfontX@#1}\wrapped@footfootmarkX{#1}\strut%
4706 \csuse{wrapcontentX@#1}{#2}%
4707 \strut\par}\allowbreak}
4708
4709 %

```

```

\threecolfootgroupX \threecolfootgroupX{<series>}
\mpthreecolfootgroupX
4710 \newcommand*{\threecolfootgroupX}[1]{\csuse{bhookgroupX@#1}\csuse{
notefontsizeX@#1}
4711 \splittopskip=\ht\strutbox
4712 \expandafter
4713 \rigidbalanceX\csname footins#1\endcsname \thr@@ \splittopskip}}
4714
4715 \newcommand*{\mpthreecolfootgroupX}[1]{\csuse{
4716 \vskip\skip\@nameuse{mpfootins#1}
4717 \ifl@dpairing\ifparledgroup
4718 \leavevmode\marks\parledgroup@{begin}%
4719 \marks\parledgroup@series{#1}%
4720 \marks\parledgroup@type{footnoteX}%
4721 \fi\fi\normalcolor
4722 \ifparledgroup%
4723 \ifl@dpairing%
4724 \else%
4725 \setnoteswidthliketwocolumnsX@{#1}%
4726 \setnotesXpositionliketwocolumns@{#1}%
4727 \print@footnoteXrule{#1}%
4728 \fi%
4729 \else%
4730 \setnoteswidthliketwocolumnsX@{#1}%
4731 \setnotesXpositionliketwocolumns@{#1}%
4732 \print@footnoteXrule{#1}%
4733 \fi%
4734 \csuse{bhookgroupX@#1}%
4735 \splittopskip=\ht\strutbox
4736 \expandafter
4737 \rigidbalanceX\csname mpfootins#1\endcsname \thr@@ \splittopskip}}
4738
4739 %

```

## XIV.4.5 Paraphed footnotes

The following macros set footnotes as one paragraph.

```

\arrangementX@paragraph \footparagraphX{<series>}

4740 \newcommand*{\arrangementX@paragraph}[1]{%
4741   \csgdef{series@displayX#1}{paragraph}%
4742   \expandafter\let\csname footstart#1\endcsname=\parafootstartX
4743   \expandafter\let\csname regvfootnote#1\endcsname=\para@vfootnoteX
4744   \expandafter\let\csname footfmt#1\endcsname=\parafootfmtX
4745   \expandafter\let\csname footgroup#1\endcsname=\para@footgroupX
4746   \expandafter\let\csname footnoterule#1\endcsname=\normalfootnoteruleX
4747   \count\csname footins#1\endcsname=1000
4748   \csxdef{default@footins#1}{1000}%Use this to confine the notes to one
side only
4749   \dimen\csname footins#1\endcsname=\csuse{maxhnotesX@#1}
4750   \skip\csname footins#1\endcsname=\csuse{beforenotesX@#1}%
4751   \advance\skip\csname footins#1\endcsname by\csuse{afterruleX@#1}%
4752   \para@footsetupX{#1}
4753   \ifnoledgroup@else
4754     \expandafter\let\csname mpvfootnote#1\endcsname=\mppara@vfootnoteX
4755     \expandafter\let\csname mpfootgroup#1\endcsname=\mppara@footgroupX
4756     \count\csname mpfootins#1\endcsname=1000
4757     \dimen\csname mpfootins#1\endcsname=\csuse{maxhnotesX@#1}
4758     \skip\csname mpfootins#1\endcsname=\csuse{beforenotesX@#1}%
4759     \advance\skip\csname mpfootins#1\endcsname by\csuse{afterruleX@#1}%
4760   \fi
4761 }
4762
4763 %

\para@footsetupX \para@footsetupX{<series>}

4764 \newcommand*{\para@footsetupX}[1]{\csuse{bhookgroupX@#1}\csuse{
notefontsizeX@#1}
4765   \setnoteswidthliketwocolumnsX@{#1}%
4766   \ifcempty{widthX@#1}%
4767     {}%
4768     {\columnwidth=\expandafter\dimexpr\csuse{widthX@#1}\relax}%
4769   \dimen0=\baselineskip
4770   \multiply\dimen0 by 1024
4771   \divide\dimen0 by \columnwidth \multiply\dimen0 by \footfudgefiddle\relax
%
4772   \expandafter
4773   \xdef\csname footfudgefactor#1\endcsname{%
4774     \expandafter\strip@pt\dimen0 }}
4775
4776 %

\parafootstartX \parafootstartX{<series>}

```

```

4777 \newcommand*{\parafootstartX}[1]{%
4778   \ifdimequal{Opt}{\prenotesX@}{}%
4779   {%
4780     \iftoggle{prenotesX@}{%
4781       \togglefalse{prenotesX@}%
4782       \skip\csname footins#1\endcsname=%
4783       \glueexpr\csuse{prenotesX@}+\csuse{afterruleX@#1}\relax%
4784     }%
4785   }%
4786 }%
4787 \leftskip=\z@
4788 \rightskip=\z@
4789 \nottoggle{parindentX@#1}{\parindent=\z@}{}%
4790 \vskip\skip\@nameuse{footins#1}%
4791 \setnoteswidthliketwocolumnsX@{#1}%
4792 \setnotesXpositionliketwocolumns@{#1}%
4793 \print@footnoteXrule{#1}%
4794 }
4795
4796 %

```

`\para@vfootnoteX`    `\para@vfootnoteX{<series>}{<text>}`  
`\mppara@vfootnoteX`

```

4797 \newcommand*{\para@vfootnoteX}[2]{%
4798   \csuse{beforeinsertingX@#1}%
4799   \insert\csname footins#1\endcsname%
4800   \bgroup
4801     \csuse{notefontsizeX@#1}
4802     \footsplitskips
4803     \setbox0=\vbox{\hsize=\maxdimen%
4804       \let\bidirTL@everypar\@empty%
4805       \insert@txtbeforenotesX{#1}%
4806       \noindent\csuse{bhooknoteX@#1}%
4807       \@nameuse{footfmt#1}{#1}{#2}}%
4808     \setbox0=\hbox{\unvXH{0}{#1}}%
4809     \dp0=\z@
4810     \ht0=\csname footfudgefactor#1\endcsname\wd0
4811     \box0
4812     \penalty0
4813   \egroup}
4814 \newcommand*{\mppara@vfootnoteX}[3]{%
4815   \get@thisfootnoteX{#1}%
4816   \get@fnmarkX{#1}{\thisc@footnote}{#3}%
4817   \edef\thisc@footnoteX@reading{\the\csname footnote#1@reading\endcsname}%
4818   \global\setbox\@nameuse{mpfootins#1}\vbox{%
4819     \unvbox\@nameuse{mpfootins#1}
4820     \csuse{notefontsizeX@#1}
4821     \footsplitskips
4822     \setbox0=\vbox{\hsize=\maxdimen%
4823       \let\bidirTL@everypar\@empty%

```

```

4824 \noindent\color@begingroup%
4825 \csuse{bhooknoteX@#1}%
4826 \@nameuse{footfmt#1}{#1}{#2}\color@endgroup}%
4827 \setbox0=\hbox{\unvxhX{0}{#1}}%
4828 \dp0=\z@
4829 \ht0=\csname footfudgefactor#1\endcsname\wd0
4830 \box0
4831 \penalty0}}
4832
4833 %

```

```

\unvxhX34 \newcommand*{\unvxhX}[2]{% 2th is optional for retro-compatibility
4835 \setbox0=\vbox{\unvbox#1%
4836 \global\setbox1=\lastbox}%
4837 \unhbox1
4838 \unskip % remove \rightskip,
4839 \unskip % remove \parfillskip,
4840 \unpenalty % remove \penalty of 10000,
4841 \hskip\csuse{afternoteX@#2}%
4842 \relax}% but add the glue to go between the notes
4843
4844 %

```

`\parafootfmtX` `\parafootfmtX{<series>}`

```

4845 \newcommand*{\parafootfmtX}[2]{%
4846 \protected@edef\@currentlabel{%
4847 \@nameuse{@thefnmark#1}%
4848 }%
4849 \insertparafootsepX{#1}%
4850 \ledsetnormalparstuff@common%
4851 {\csuse{notenumfontX@#1}%
4852 \csuse{notenumfontX@#1}%
4853 \wrapped@footfootmarkX{#1}%
4854 \strut%
4855 \csuse{wrapcontentX@#1}{#2}%
4856 \penalty-10}}
4857
4858 %

```

`\para@footgroupX` `\para@footgroupX{<series>}`  
`\mppara@footgroupX`

```

4859 \newcommand*{\para@footgroupX}[1]{%
4860 \hsize=\expandafter\dimexpr\csuse{widthX@#1}\relax%
4861 \unvbox\csname footins#1\endcsname
4862 \ifcsstring{raggedX@#1}{L}{\RaggedLeft}{}%
4863 \ifcsstring{raggedX@#1}{R}{\RaggedRight}{}%
4864 \makehboxofhboxes
4865 \setbox0=\hbox{\unhbox0 \removehboxes}%

```

```

4866 \csuse{bhookgroupX@#1}
4867 \csuse{notefontsizeX@#1}
4868 \unhbox0\par}
4869
4870 \newcommand*{\mppara@footgroupX}[1]{%
4871   \setnoteswidthliketwocolumnsX@{#1}%
4872   \vskip\skip\@nameuse{mpfootins#1}
4873   \ifl@dpairing\ifparledgroup
4874     \leavevmode%
4875     \leavevmode\marks\parledgroup@{begin}%
4876     \marks\parledgroup@series{#1}%
4877     \marks\parledgroup@type{footnoteX}%
4878   \fi\fi\normalcolor
4879   \ifparledgroup%
4880     \ifl@dpairing%
4881     \else%
4882       \setnoteswidthliketwocolumnsX@{#1}%
4883       \setnotesXpositionliketwocolumns@{#1}%
4884       \print@footnoteXrule{#1}%
4885     \fi%
4886   \else%
4887     \setnoteswidthliketwocolumnsX@{#1}%
4888     \setnotesXpositionliketwocolumns@{#1}%
4889     \print@footnoteXrule{#1}%
4890   \fi%
4891   \unvbox\csname mpfootins#1\endcsname
4892   \ifcsstring{raggedX@#1}{L}{\RaggedLeft}{}%
4893   \ifcsstring{raggedX@#1}{R}{\RaggedRight}{}%
4894   \makehboxofhboxes
4895   \setbox0=\hbox{\unhbox0 \removehboxes}%
4896   \csuse{bhookgroupX@#1}%
4897   \csuse{notefontsizeX@#1}%
4898   \nottoggle{parindentX@#1}{\parindent=\z@}{}%
4899   \unhbox0\par}}
4900
4901 %

```

**Insertion of the footnotes separator** The command `\insertparafootsepX{<series>}` must be called at the beginning of `\parafootftmX`.

```

\insertparafootsepX%
4902 \newcommand{\insertparafootsepX}[1]{%
4903   \ifledRcol%
4904     \ifnumequal{\csuse{prevpage#1@numR}}{\page@numR}%
4905     {\csuse{Xparafootsep@#1}}%
4906     {}%
4907   \global\csname prevpage#1@numR\endcsname=\page@numR%
4908   \else%
4909     \ifnumequal{\csuse{prevpage#1@num}}{\page@num}%
4910     {\csuse{Xparafootsep@#1}}%

```



```

4911     {}%
4912     \global\csname prevpage#1@num\endcsname=\page@num%
4913     \fi%
4914 }
4915 %

```

## 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.

```

4916 \newcommand{\wrapped@footfootmarkX}[1]{%
4917   \ifdefined\hypertarget%
4918     \hyperlink%
4919       {@bodyfootmark#1@\this@footnoteX@reading}%
4920       {@\nameuse{footfootmark#1}}%
4921     \Hy@raisedlink{%
4922       \hypertarget%
4923         {@footnotemark#1@\this@footnoteX@reading}%
4924         {}%
4925     }%
4926   \else%
4927     \@nameuse{footfootmark#1}%
4928   \fi%
4929 }%
4930 %

```

**\wrapped@bodyfootmarkX** \wrapped@bodyfootmarkX prints the footnote mark of the text body, wrapped in hyperref package's commands, if needed.

```

4931 \newcommand{\wrapped@bodyfootmarkX}[1]{%
4932   \ifdefined\hypertarget%
4933     \hyperlink%
4934       {@footnotemark#1\expandafter\the\csname footnote#1@reading\
4935       endcsname}%
4936       {@\nameuse{bodyfootmark#1}}%
4937     \Hy@raisedlink{%
4938       \hypertarget%
4939         {@bodyfootmark#1\expandafter\the\csname footnote#1@reading\
4940         endcsname}%
4941         {}%
4942     }%
4943   \else%
4944     \@nameuse{bodyfootmark#1}%
4945   \fi%
4946 }%
4947 %

```

## 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 46 \newcommand*{\ledsetnormalparstuff@common}{%
\Xledsetnormalparstuff 47 \ifluatex%
\ledsetnormalparstuffX 48 \ifdefstring{\footnote@luatextextdir}{TLT}{}%
499 \{\textdir\footnote@luatextextdir}%
490 \pardir\footnote@luatexpardir%
491 \fi%
492 \csuse{\csuse{footnote@dir}}%
493 \normal@pars%
494 \parfillskip \z@ \@plus 1fil}%
495
496 \newcommand*{\Xledsetnormalparstuff}[1]{%
497 \ledsetnormalparstuff@common%
498 \nottoggle{Xparindent@#1}{\parindent=\z@}{\hspace{\parindent}}%
499 }%
4960
4961 \newcommand*{\ledsetnormalparstuffX}[1]{%
4962 \ledsetnormalparstuff@common%
4963 \nottoggle{parindentX@#1}{\parindent=\z@}{\hspace{\parindent}}%
4964 }%
4965 %

```

## 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. They set, if the option is enabled, the `\hsize`. They are also called at the on the setup for paragraphed notes.

`\setXnoteswidthliketwocolumns@`  
`\setnoteswidthliketwocolumnsX@`

```

4966 \newdimen\old@hsize%
4967
4968 \AtBeginDocument{\old@hsize=\hsize}%
4969

```

```

4970 \newcommand{\setXnoteswidthliketwocolumns@}[1]{%
4971   \global\let\hsize@fornote=\hsize%
4972   \global\old@hsize=\hsize%
4973   \let\old@columnwidth=\columnwidth%
4974   \iftoggle{Xnoteswidthliketwocolumns@#1}{%
4975     {%
4976       \setwidthliketwocolumns%
4977       \global\let\hsize@fornote=\hsize%
4978     }%
4979   }%
4980   \let\hsize=\hsize@fornote%
4981   \let\columnwidth=\old@columnwidth%
4982 }%
4983
4984 \newcommand{\setnoteswidthliketwocolumnsX@}[1]{%
4985   \global\let\hsize@fornote=\hsize%
4986   \global\old@hsize=\hsize%
4987   \let\old@columnwidth=\columnwidth%
4988   \iftoggle{noteswidthliketwocolumnsX@#1}{%
4989     {%
4990       \setwidthliketwocolumns%
4991       \global\let\hsize@fornote=\hsize%
4992     }%
4993   }%
4994   \let\hsize=\hsize@fornote%
4995   \let\columnwidth=\old@columnwidth%
4996 }%
4997
4998 %

```

`\setXnotespositionliketwocolumns@` 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`.

```

4999 \newcommand{\setXnotespositionliketwocolumns@}[1]{%
5000   \iftoggle{Xnoteswidthliketwocolumns@#1}{%
5001     \csuse{setnotespositionliketwocolumns@\columns@position}%
5002   }%
5003 }%
5004
5005 \newcommand{\setnotesXpositionliketwocolumns@}[1]{%
5006   \iftoggle{noteswidthliketwocolumnsX@#1}{%
5007     \csuse{setnotespositionliketwocolumns@\columns@position}%
5008   }%
5009 }%
5010
5011 %

```

## 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.

```

5012 \def\@fnpos{familiar-critical}
5013 \def\@mpfnpos{critical-familiar}
5014 \newcommand{\fnpos}[1]{\xdef\@fnpos{#1}}
5015 \newcommand{\mpfnpos}[1]{\xdef\@mpfnpos{#1}}
5016 %

```

## 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@Xfootnoterule17 \newcommand{\print@Xfootnoterule}[1]{%
\print@footnoterule18 \vskip-\csuse{Xafterrule@#1}%Because count in \dimen\csuse{#1footins}
5019 \nointerlineskip%
5020 \moveleft-\leftskip\vbox{\csuse{#1footnoterule}}%
5021 \nointerlineskip%
5022 \vskip\csuse{Xafterrule@#1}%
5023 }%
5024
5025 \newcommand{\print@footnoterule}[1]{%
5026 \vskip-\csuse{afterruleX@#1}%Because count in \dimen\csuse{footins#1}
5027 \nointerlineskip%
5028 \moveleft-\leftskip\vbox{\csuse{footnoterule#1}}%
5029 \nointerlineskip%
5030 \vskip\csuse{afterruleX@#1}%
5031 }%
5032
5033 %

```

## 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.

```

5034 \newtoggle{Xprenotes@}%
5035 \toggletrue{Xprenotes@}%
5036 \newcommand{Xprenotes@}{Opt}%
5037 \newcommand*{\Xprenotes}[1]{\renewcommand{\Xprenotes@}{#1}}%
5038 \newcommand{\preXnotes}[1]{\led@warning@preXnotes@deprecated\Xprenotes{#1}}
    %For compatibility
5039 %

```

The same, but for familiar footnotes.

```

\Xprenotes@40 \newtoggle{prenotesX@}
\Xprenotes@41 \toggletrue{prenotesX@}
5042 \newcommand{prenotesX@}{Opt}
5043 \newcommand*{\prenotesX}[1]{\renewcommand{\prenotesX@}{#1}}
5044 %

```

## XIX.3 Internal commands

```

firstXseries@45 \gdef\firstXseries@{}
prepare@Xprenotes@46 \newcommand{\prepare@Xprenotes}[1]{%
5047   \ifdimequal{Opt}{\Xprenotes@}%

```

```

5048 {}%
5049 {%
5050   \IfStrEq{\firstXseries@}{}{%
5051     \global\skip\csuse{#1footins}=\Xprenotes@%
5052     \global\advance\skip\csname #1footins\endcsname by\csuse{Xafterterrule@
#1}%
5053     \gdef\firstXseries@{#1}%
5054   }%
5055   {%
5056     \ifseriesbefore{#1}{\firstXseries@}%
5057     {%
5058       \global\skip\csuse{#1footins}=\csuse{Xbeforenotes@\firstXseries@}%
5059       \global\advance\skip\csname #1footins\endcsname by\csuse{Xafterterrule@
#1}%
5060       \gdef\firstXseries@{#1}%
5061     }%
5062     {}%
5063   }%
5064 }%
5065 }
5066 %

```

The same thing is required for familiar notes and \prenotesX.

```

firstseriesX@67 \gdef\firstseriesX@{}
prepare@prenotesX68 \newcommand{\prepare@prenotesX}[1]{%
5069   \ifdimequal{0pt}{\prenotesX@}%
5070   {}%
5071   {%
5072     \IfStrEq{\firstseriesX@}{}{%
5073       \global\skip\csuse{footins#1}=\prenotesX@%
5074       \global\advance\skip\csname footins#1\endcsname by\csuse{afterterruleX@
#1}%
5075       \gdef\firstseriesX@{#1}%
5076     }%
5077     {%
5078       \ifseriesbefore{#1}{\firstseriesX@}%
5079       {%
5080         \global\skip\csuse{footins#1}=\csuse{beforenotesX@\firstseriesX@}%
5081         \global\advance\skip\csname footins#1\endcsname by\csuse{afterterruleX@
#1}%
5082         \gdef\firstXseries@{#1}%
5083       }%
5084       {}%
5085     }%
5086   }%
5087 }
5088 %

```

## XX Endnotes

First, check the noend option.

```
5089 \ifbool{noend@}{}{%Used instead of \ifnoend@ to prevent expansion problem
5090 %
```

### 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.

```
5091 \newcommand{\l@dend@open}[1]{%
5092   \global\booltrue{l@dend@#1}%
5093   \expandafter\immediate%
5094   \expandafter\openout%
5095   \csname l@d@#1end\endcsname%
5096   =\l@auxdir\jobname.#1end\relax%
5097 }%
5098 \newcommand{\l@dend@close}[1]{%
5099   \global\boolfalse{l@dend@#1}%
5100   \expandafter\immediate%
5101   \expandafter\closeout\csname l@d@#1end\endcsname%
5102 }%
5103
5104 %
```

`\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.

```
5105 \newcommand{\l@dend@stuff}{%
5106   \def\do##1{%
5107     \ifbool{l@dend@##1}{}%
5108     {\l@dend@open{##1}}%
5109     \expandafter\immediate\expandafter\write\csname l@d@##1end\endcsname{\
string\l@d@section{\the\section@num}\@percentchar}%
5110   }%
5111   \dolistloop{\@series}%
5112 }%
5113
5114 %
```

`\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.

```

5115 \global\newboolean\parapparatus@{ }\def\endprint#1#2#3#4#5#6#7{
5116 \iftoggle{Xendfirstnote@#4}{
5117 \global\togglefalse{Xendfirstnote@#4}%
5118 \csuse{Xendtxtbeforenotes@#4}%
5119 }{ }%
5120 \csuse{Xendbhooknote@#4}%
5121 \csuse{Xendnotefontsize@#4}%
5122 \hangindent=\csuse{Xendhangindent@#4}%
5123 \ifXendinsertsep%
5124 \hskip\csuse{Xendafternote@#4}\relax%
5125 \csuse{Xendsep@#4}%
5126 \else%
5127 \iftoggle{Xendparagraph@#4}%
5128 { \global\Xendinsertsep@true}%
5129 {}%
5130 \fi%
5131 \xdef\@currentseries{#4}%
5132 \def\do##1{%
5133 \setkeys[mac]{truefootnoteoption}{##1}%
5134 }%
5135 \notblank{#5}{\docsvlist{#5}}{ }%
5136 \IfStrEq{#6}{R}{\ledRcol@true}{ }%
5137 \def\@this@crossref@start{#7:start}%
5138 \def\@this@crossref@end{#7:end}%
5139 \printlineendnote{#1}{#4}%
5140 \IfStrEq{#6}{R}{\ledRcol@false}{ }%
5141 \nottoggle{Xendlemmadisablefontselection@#4}%
5142 { \select@lemmafont#1}%
5143 {}%
5144 \bgroup%
5145 \csuse{Xendlemmafont@#4}%
5146 \csuse{Xendwraplemma@#4}{#2}%
5147 \egroup%
5148 \ifboolexpr{

```



```

5149   togl {nosep@}%
5150   or test{\ifcsemtty{Xendlemmaseparator@#4}}%
5151   }%
5152   {\hskip\csuse{Xendinplaceoflemmaseparator@#4}\relax}%
5153   {\nobreak%
5154   \hskip\csuse{Xendbeforelemmaseparator@#4}%
5155   \csuse{Xendlemmaseparator@#4}%
5156   \hskip\csuse{Xendafterlemmaseparator@#4}%
5157   \relax%
5158   }%
5159   \csuse{Xendwrapcontent@#4}{#3}%
5160   \nottoggle{Xendparagraph@#4}{\par}{}%
5161   \def\do##1{%
5162     \setkeys[mac]{falsefootnoteoption}{##1}%
5163   }%
5164   \notblank{#5}{\docsvlist{#5}}{%}%
5165 }}%
5166
5167 \let\l@d@section=\@gobble
5168
5169 %

```

**\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.

```

5170 \newcommand{\printlineendnote}[2]{%
5171   \l@dp@rsefeet@spec#1|{%
5172     \iftoggle{Xendnumberonlyfirstintwolines@#2}{%
5173       \xdef\lineinfo@{\l@dparsedstartpage - \l@dparsedstartline - \
5174       \l@dparsedstartsub - \l@dparsedendpage - \l@dparsedendline - \
5175       \l@dparsedendsub}%
5176     }%
5177     {%
5178       \xdef\lineinfo@{\l@dparsedstartpage - \l@dparsedstartline - \
5179       \l@dparsedstartsub}%
5180     }%
5181     \ifboolexpr{%
5182       togl {nonum@}%
5183       or togl {Xendnonumber@#2}%
5184     }%
5185     {\hspace{\csuse{Xendinplaceofnumber@#2}}}%
5186     {%
5187       \iftoggle{Xendnumberonlyfirstinline@#2}%
5188       {\ifcsdef{prevendline#2}%
5189         {\ifcsequal{prevendline#2}{\lineinfo@}%
5190         }%
5191         \csuse{Xendbhookinplaceofnumber@#2}%
5192         \ifcsemtty{Xendsymnum@#2}%
5193         {\hspace{\csuse{Xendinplaceofnumber@#2}}}%

```

```

5191         {\printsymlineendnotearea{#2}}%
5192         \csuse{Xendahookinplaceofnumber@2}%
5193     }%
5194     {\printlineendnotearea{#1}{#2}}}%
5195     {\printlineendnotearea{#1}{#2}}%
5196 }%
5197 {\printlineendnotearea{#1}{#2}}%We keep every time line
5198 \csxdef{prevendline#2}{\lineinfo@}%
5199 }%
5200 }%
5201 %

```

```

\printsymlineendnotearea \newcommand{\printsymlineendnotearea}[1]{%
5203     \hspace{\csuse{Xendbeforenum@#1}}%
5204     \csuse{Xendnotenumfont@#1}%
5205     \ifdimequal{\csuse{Xendboxsymlinenum@#1}}{\z@}%
5206         {\csuse{Xendsymlinenum@#1}}%
5207         {\hbox to \csuse{Xendboxsymlinenum@#1}%
5208             {\csuse{Xendsymlinenum@#1}\hfill}%
5209         }%
5210     \hspace{\csuse{Xendaftersymlinenum@#1}}%
5211 }%
5212 %

```

**\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.)

```

5213 \newcommand{\printlineendnotearea}[2]{%
5214     \csuse{Xendbhooklinenumber@#2}%
5215     \hspace{\csuse{Xendbeforenumber@#2}}%
5216     \bgroup%
5217     \csuse{Xendnotenumfont@#2}%
5218     \ifdimequal{\csuse{Xendboxlinenum@#2}}{\Opt}%
5219         {\printendlines#1||\ifledRcol@{\Rlineflag\fi}%
5220         {\leavevmode%
5221             \hbox to \csuse{Xendboxlinenum@#2}%
5222             {%
5223                 \IfSubStr{RC}{\csuse{Xendboxlinenumalign@#2}}{\hfill}{}}%
5224                 \printendlines#1||\ifledRcol@{\Rlineflag\fi%
5225                 \IfSubStr{LC}{\csuse{Xendboxlinenumalign@#2}}{\hfill}{}}%
5226             }}%
5227     \egroup%
5228     \hspace{\csuse{Xendafternumber@#2}}%
5229     \csuse{Xendahooklinenumber@#2}%
5230 }%
5231 %

```

## 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.

```

\toendnotes* \newcommandx{\toendnotes}[2][1,usedefault]{%
\toendnotes* \ifboolexpr{bool{numbering} or bool{numberingR}}{%
5234   \def\do##1{%
5235     \expandafter\immediate\expandafter\write\csname l@d@##1end\endcsname%
5236     {\unexpanded{#2}\@percentchar}%
5237   }%
5238   \ifstrempy{#1}%
5239     {\dolistloop{\@series}}%
5240     {\docsvlist{#1}}%
5241   }\led@err@toendnotes@outsidenumbering}%
5242 }%
5243 \WithSuffix\newcommandx\toendnotes*[2][1,usedefault]{%
5244   \ifboolexpr{bool{numbering} or bool{numberingR}}{%
5245     \def\do##1{%
5246       \expandafter\immediate\expandafter\write\csname l@d@##1end\endcsname%
5247       {#2\@percentchar}%
5248     }%
5249     \ifstrempy{#1}%
5250       {\dolistloop{\@series}}%
5251       {\docsvlist{#1}}%
5252     }\led@err@toendnotes@outsidenumbering}%
5253 }%
5254 %

```

### 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.

```

5255 \newif\ifXendinsertsep@
5256 \newcommand*\doendnotes[1]{%
5257   \l@dend@close{#1}%
5258   \begingroup
5259     \csxdef{prevpagenum@#1}{}%
5260     \csxdef{prevpagerange@#1}{}%
5261     \global\toggletrue{Xendfirstnote@#1}%
5262     \makeatletter
5263     \expandafter\let\csname #1end\endcsname=\endprint
5264     \input\l@auxdir\jobname.#1end%
5265     \global\Xendinsertsep@false%

```

```

5266 \endgroup}
5267 %

```

**\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.

```

5268 \newcommand*{\doendnotesbysection}[1]{%
5269 \l@dend@close{#1}%
5270 \csxdef{prevpagenum@#1}{}%
5271 \csxdef{prevpagerange@#1}{}%
5272 \global\expandafter\advance\csname #1end@bysection\endcsname by 1%
5273 \begingroup%
5274 \makeatletter%
5275 \def\l@d@section##1{%
5276 \global\toggletrue{Xendfirstnote@#1}%
5277 \ifnumequal{##1}{\csname #1end@bysection\endcsname}%
5278 {\cslet{#1end}{\endprint}}%
5279 {\cslet{#1end}{\@gobbleseven}}%
5280 }%
5281 \input\l@auxdir\jobname.#1end%
5282 \global\Xendinsertsep@false%
5283 \endgroup%
5284 }%
5285 %

```

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.

```

5286 }%
5287 %

```

**\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.

```

5288 \newcommand*{\setprintendlines}[6]{%
5289 \let\@annot@start@print\relax%
5290 \let\@annot@end@print\relax%
5291 \l@d@pnumfalse%
5292 \l@d@dashfalse%
5293 \l@d@elinfalse%
5294 %

```

First of all, we print the second page number only if the ending page number is different

```

5295 \ifnum#4=#1 \else
5296   \l@d@pnumtrue
5297   \l@d@dashtrue
5298 \fi
5299 %

```

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.

```

5300 \ifboolexpr{%
5301   bool{\l@d@pnum}%
5302   or not test{\ifnumequal{#2}{#5}}%
5303 }{%
5304   \l@d@elintrue%
5305   \l@d@dashtrue%
5306   \l@d@elintrue
5307   \l@d@dashtrue
5308   \unless\ifx\relax\annot@end%
5309   \def\@annot@end@print{%
5310     \l@wrapcs@ifnotemptybox{Xendwraplinenumannotation@\@currentseries
5311 }{\@annot@end}%
5312   }%
5313   \fi%
5314 }%
5315 {}%
5316 %

```

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.

```

5316 \ifl@d@elin%
5317   \def\@annot@start@print{%
5318     \l@wrapcs@ifnotemptybox{Xendwraplinenumannotation@\@currentseries}{\
5319 annot@start}%
5320   }%
5321   \else%
5322     \ifx\annot@start\annot@end%
5323       \unless\ifx\@annot@start\relax%
5324       \def\@annot@start@print{%
5325         \l@wrapcs@ifnotemptybox{Xendwraplinenumannotation@\@currentseries
5326 }{\@annot@start}%
5327       }%
5328       \fi%
5329     \else%
5330     \ifx\@annot@end@print\relax%
5331     \def\@annot@start@print{%
5332       \l@wrapcs@ifnotemptybox{Xendwraplinenumannotation@\@currentseries
5333 }{%
5334       \ifx\annot@start\empty\else%

```

```

5332         \annot@start%
5333         \ifdefined\linangesep@%
5334             \linangesep@%
5335         \else%
5336             \csuse{Xendlinangeseparator@}\@currentseries}%
5337         \fi%
5338     \fi%
5339     \annot@end%
5340 }%
5341 }%
5342 \else%
5343     \let\@annot@start@print\@annot@end@print%
5344     \let\@annot@end@print\relax%
5345 \fi%
5346 \fi%
5347 \fi%
5348 %

```

We print the starting sub-line if it is nonzero.

```

5349 \l@d@ssubfalse
5350 \ifnum#3=0 \else
5351     \l@d@ssubtrue
5352 \fi
5353 %

```

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.

```

5354 \l@d@eslfalse
5355 \ifnum#6=0 \else
5356     \ifnum#6=#3
5357         \ifl@d@elin \l@d@esltrue \else \l@d@eslfalse \fi
5358     \else
5359         \l@d@esltrue
5360         \l@d@dashtrue
5361     \fi
5362 \fi%
5363 %

```

```

5364 \ifl@d@dash%
5365     \ifbool{expr}{togl{fulllines@} or test{\ifcsemt{Xendtwolines@}\@currentseries}}}%
5366     {}%
5367     {%
5368     \setistwofollowinglines{#1}{#2}{#4}{#5}%
5369     \ifbool{expr}%
5370     (%
5371         togl {Xendtwolinesbutnotmore@\@currentseries}%
5372         and not%
5373         (%
5374             bool {istwofollowinglines@}%

```

```

5375         )%
5376     )%
5377     or%
5378     (%
5379         (not test{\ifnumequal{#1}{#4}})%
5380         and togl{Xendtwolinesonlyinsamepage@ \@currentseries}%
5381     )%
5382 }%
5383 {}%
5384 {%
5385     \l@d@dashfalse%
5386     \l@d@Xtwolinestrue%
5387     \l@d@elinlinefalse%
5388     \l@d@eselfalse%
5389     \ifcseempty{Xendmorethantwolines@ \@currentseries}%
5390     {%
5391         {\ifistwofollowinglines@ \else%
5392             \l@d@Xmorethantwolinestrue%
5393         }%
5394     }%
5395 }%
5396 }%
5397 \fi%
5398 %

```

If the `\Xendnoidenticallinenumannotation` 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

```

5399     \iftoggle{Xendnoidenticallinenumannotation@ \@currentseries}{%
5400         \ifx\annot@start\annot@end%
5401             \let\annot@end@print\relax%
5402             \ifx\linenumrep\@gobble%Don't print the dash if we're not printing
the line number
5403                 \l@d@dashfalse%
5404             \fi%
5405         \fi%
5406     }{}%
5407 %

```

Finally, we check for `\Xendlinenumannotationonlyfirst` and `\Xendlinenumannotationonlyfirstintwo`, and we redefine, if required, `\annot@start@print` and `\annot@end@print`. We also store the current line number annotations.

```

5408     \iftoggle{Xendlinenumannotationonlyfirst@ \@currentseries}{%
5409         \ifboolexpr{%
5410             (%
5411                 togl{Xendlinenumannotationonlyfirstintwo@ \@currentseries}%
5412                 and test{\ifcsequal{annot@start}{prevannot@start@ \@currentseries}}%
5413                 and test{\ifcsequal{annot@end}{prevannot@end@ \@currentseries}}%
5414             )%

```

```

5415     or%
5416     (%
5417         not tog1{Xendlinenumannotationonlyfirstintwo@\\@currentseries}%
5418         test{\\ifcsequal{annot@start}{prevannot@start@\\@currentseries}}%
5419     )%
5420 }{%
5421     \\def\\@annot@start@print{%
5422         \\l@wrapcs@ifnotemptybox{Xendwraplinenumannotation@\\@currentseries
5423     }{\\csuse{Xendsym\\linenumannotation@\\@currentseries}}%
5424     }%
5425     \\let\\@annot@end@print\\relax%
5426     \\ifx\\linenumrep\\@gobble%Don't print the dash if we're not printing
the line number
5427     \\l@d@dashfalse%
5428     \\fi%
5429 }{%
5430     \\global\\cslet{prevannot@start@\\@currentseries}{\\@annot@start}%
5431     \\global\\cslet{prevannot@end@\\@currentseries}{\\@annot@end}%
5432 }%
5433 }%

```

End of \\setprintendlines.

```

5434 }%
5435 %

```

**\\printendlines** Now we are ready to print it all.

```

5436 \\def\\printendlines#1|#2|#3|#4|#5|#6|#7|#8|{%
5437     \\begingroup
5438     \\setprintendlines{#1}{#2}{#3}{#4}{#5}{#6}%
5439 %

```

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.

```

5440     \\ifdimequal{\\csuse{Xendboxstartlinenum@\\@currentseries}}{0pt}%
5441     {\\bgroup}%
5442     {\\leavevmode\\hbox to \\csuse{Xendboxstartlinenum@\\@currentseries}\\bgroup
\\hfill}%
5443 %

```

Then, print the starting page number-

```

5444     \\ifboolexpr{%
5445     (%
5446         test{\\ifcsstring{prevpagenum@\\@currentseries}{#1}}%
5447         and not%

```



```

5448      (togl{Xendpagenumberonlyfirstifsingle@\@currentseries} and bool{
1@d@pnum})%
5449    )%
5450    or%
5451    (%
5452    test {\ifcsstring{prevpagerange@\@currentseries}{#1-#4}}%
5453    )%
5454  }%
5455  {%
5456  \ifcsemtyp{Xendsympagenum@\@currentseries}%
5457    {\hspace{\csuse{Xendinplaceofpagenumber@\@currentseries}}}%
5458    {\csuse{Xendsympagenum@\@currentseries}}%
5459  }%
5460  {%
5461  \wrap@edcrossref{\@this@crossref@start}{\printnpnum{#1}}%
5462  }%
5463  %

```

Then, determine what must be printed before the start line.

```

5464  \ifl@d@dash%
5465  \ifl@d@pnum%
5466    \csuse{Xendlineprefixsingle@\@currentseries}%
5467  \else%
5468    \ifcsemtyp{Xendlineprefixmore@\@currentseries}%
5469    {\csuse{Xendlineprefixsingle@\@currentseries}}%
5470    {\csuse{Xendlineprefixmore@\@currentseries}}%
5471  \fi%
5472  \else%
5473    \csuse{Xendlineprefixsingle@\@currentseries}%
5474  \fi%
5475  %

```

Then print the starting line, followed, if needed, by the side flag and the starting sub line number, then the line number annotation.

```

5476  \ifcsstring{Xendlinenumannotationposition@\@currentseries}{before}%
5477  {\@annot@start@print}%
5478  {}%
5479  \wrap@edcrossref{\@this@crossref@start}{%
5480    \ifledRcol@%
5481    \linenumrepR{#2}%
5482  \else%
5483    \linenumrep{#2}%
5484  \fi%
5485  }%
5486  \iftoggle{Xendlineflag@\@currentseries}{\ifledRcol@\@Rlineflag\fi}{}%
5487  \ifl@d@ssub%
5488    \csuse{Xendsublinesep@\@currentseries}%
5489    \wrap@edcrossref{\@this@crossref@start}{%
5490      \ifledRcol@%
5491      \sublinenumrepR{#3}%

```

```

5492     \else%
5493         \sublinenumrep{#3}%
5494     \fi%
5495 }%
5496 \fi%
5497 \ifcsstring{Xendlinenumannotationposition@ \@currentseries}{after}%
5498     {\@annot@start@print}%
5499     {}%
5500 %

```

Close the box.

```

5501 \egroup%
5502 %

```

Open the box for the ending line number.

```

5503 \ifdimequal{\csuse{Xendboxendlinenum@ \@currentseries}}{0pt}%
5504     {\bgroup}%
5505     {\hbox to \csuse{Xendboxendlinenum@ \@currentseries}\bgroup}%
5506 %

```

Print the dash + the ending line number, or the line number range symbol.

```

5507 \ifl@d@Xtwolines%
5508     \ifl@d@Xmorethantwolines%
5509         \csuse{Xendmorethantwolines@ \@currentseries}%
5510     \else%
5511         \csuse{Xendtwolines@ \@currentseries}%
5512     \fi%
5513 \else%
5514     \ifl@d@dash%
5515         \ifdefined\linerrangesep%
5516             \linerrangesep%
5517         \else%
5518             \csuse{Xendlinerrangeseparator@ \@currentseries}%
5519         \fi%
5520     \fi%
5521 %

```

Print the ending page number.

```

5522 \ifl@d@pnum%
5523     \ifcsstring{prevpagerange@ \@currentseries}{#1-#4}%
5524     {%
5525         \ifcsempy{Xendsympagenum@ \@currentseries}%
5526             {\hspace{\csuse{Xendinplaceofpagenumber@ \@currentseries}}}%
5527             {\csuse{Xendsympagenum@ \@currentseries}}%
5528         }%
5529     {%
5530         \wrap@edcrossref{\@this@crossref@end}\printnpnum{#4}%
5531     }%
5532 \fi%
5533 %

```

Print the ending line number, with, if needed, the line prefix, and followed by the side flag, the subline number, and line number annotation.

```

5534 \ifcsstring{Xendlinenumannotationposition@\@currentseries}{before}%
5535 {\@annot@end@print}%
5536 }%
5537 \ifl@d@elin%
5538 \ifl@d@pnum\csuse{Xendlineprefixsingle@\@currentseries}\fi%
5539 \wrap@edcrossref{\@this@crossref@end}{%
5540 \ifledRcol%
5541 \linenumrepR{#5}%
5542 \else%
5543 \linenumrep{#5}%
5544 \fi%
5545 }%
5546 \iftoggle{Xendlineflag@\@currentseries}{\ifledRcol@\@Rlineflag\fi}{%
5547 \fi%
5548 \ifl@d@esl%
5549 \ifl@d@elin%
5550 \csuse{Xendsublinesep@\@currentseries}%
5551 \fi%
5552 \wrap@edcrossref{\@this@crossref@end}{%
5553 \ifledRcol%
5554 \sublinenumrepR{#6}%
5555 \else%
5556 \sublinenumrep{#6}%
5557 \fi%
5558 }%
5559 \fi%
5560 \ifcsstring{Xendlinenumannotationposition@\@currentseries}{after}%
5561 {\@annot@end@print}%
5562 }%
5563 \fi%
5564 %

```

Close the ending line box.

```

5565 \ifdimequal{\csuse{Xendboxendlinenum@\@currentseries}}{0pt}%
5566 {}%
5567 {\hfill}%Prevent underfull hbox
5568 \egroup%
5569 %

```

And, finally, save, if needed, the current page number for the Xendpagenumberonlyfirst hooks.

```

5570 \iftoggle{Xendpagenumberonlyfirst@\@currentseries}%
5571 {\iftoggle{Xendpagenumberonlyfirstintwo@\@currentseries}%
5572 {\csxdef{prevpagerange@\@currentseries}{#1-#4}}%
5573 {\csxdef{prevpagenum@\@currentseries}{#4}}%
5574 }%
5575 {}%

```

```

5576 %
    Now, the end of \printendlines macro.
5577 \endgroup%
5578 }%
5579
5580 %

```

**\printnnum** A macro to print a page number in an endnote. Should not be override anymore

```

5581 \newcommand*\printnnum[1]{\csuse{Xendbeforepagenumber@\@currentseries
    }#1\csuse{Xendafterpagenumber@\@currentseries}}
5582
5583 %

```

## XXI Generate series of notes

In this section, X means the name of the series (A, B etc.)

**\series** **\series** creates one more new series. It is a public command, which just loops on the private command **\newseries@**.

```

5584 \newcommand{\newseries}[1]{%
5585     \def\do##1{\newseries@{##1}}%
5586     \docsvlist{#1}
5587 }
5588 %

```

**\@series** The **\series@** macro is an etoolbox list, which contains the name of all series.

```

5589 \newcommand{\@series}{}
5590 %

```

The command **\newseries@****\series** creates a new series of the footnote.

```

\newseries@91 \newcommand{\newseries@}[1]{
5592 %

```

### XXI.1 Test if series is still existing

```

5593 \xifinlist{#1}{\@series}{\led@warn@SeriesStillExist{#1}}%
5594 {%
5595 %

```

### XXI.2 Init specific to reledpar

When calling **\newseries@** after having loaded **reledpar**, we need to load specific setting.

```

5596 \ifdefined\newseries@par%
5597 \newseries@par{#1}%
5598 \fi%
5599 %

```

### XXI.3 For critical footnotes

Critical footnotes are those which start with letters. We look for the `\nocritical` option of `reledmac`.

```

5600 \unless\ifnocritical@
5601 %

```

#### XXI.3.1 Options

```

5602 \newtoggle{Xlineflag@#1}
5603 \newtoggle{Xparindent@#1}
5604 \newtoggle{Xlemmadisablefontselection@#1}
5605 \csgdef{Xwrapcontent@#1}{}%
5606 \csgdef{Xbeforeinserting@#1}{}%
5607 \csgdef{Xhangindent@#1}{Opt}%
5608 \csgdef{Xragged@#1}{}%
5609 \csgdef{Xhsizetwocol@#1}{0.45 \hsize}%
5610 \csgdef{Xhsizethreecol@#1}{.3 \hsize}%
5611 \csgdef{Xcolalign@#1}{\raggedright}%
5612 \csgdef{Xnotenumfont@#1}{\normalfont}%
5613 \csgdef{Xnotefontsize@#1}{\footnotesize}%
5614 \csgdef{Xbhooknote@#1}{}%
5615 \csgdef{Xbhookgroup@#1}{}%
5616
5617 \csgdef{Xboxlinenum@#1}{Opt}%
5618 \csgdef{Xboxlinenumalign@#1}{L}%
5619
5620 \csgdef{Xboxstartlinenum@#1}{Opt}%
5621 \csgdef{Xboxendlinenum@#1}{Opt}%
5622
5623 \csgdef{Xboxsymlinenum@#1}{Opt}%
5624 \newtoggle{Xgroupbyline@#1}%
5625 \newtoggle{Xgroupbylineseparetwolines@#1}%
5626
5627 \newtoggle{Xnumberonlyfirstinline@#1}%
5628 \newtoggle{Xnumberonlyfirstintwolines@#1}%
5629
5630 \newtoggle{Xlinenumannotationonlyfirst@#1}%
5631 \newtoggle{Xlinenumannotationonlyfirstintwo@#1}%
5632
5633 \csgdef{Xtwolines@#1}{}%
5634 \csgdef{Xmorethantwolines@#1}{%

```

```

5635 \csgdef{Xsublinesep@#1}{\fullstop}%
5636 \csgdef{Xpagelinesep@#1}{\csname Xsublinesep@#1\endcsname}%for
backward compatibility, call Xsublinesep@#1
5637 \newtoggle{Xtwolinesbutnotmore@#1}%
5638 \newtoggle{Xtwolinesonlyinsamepage@#1}%
5639 \newtoggle{Xonlypstart@#1}%
5640 \newtoggle{Xpstarteverytime@#1}%
5641 \newtoggle{Xpstart@#1}%
5642 \newtoggle{Xstanza@#1}%
5643 \csgdef{Xstanzaseparator@#1}{}%
5644
5645 \csgdef{Xsymlinenum@#1}{}%
5646 \csgdef{Xsymlinenumannotation@#1}{}%
5647
5648 \newtoggle{Xnonnumber@#1}%
5649 \csgdef{Xbeforenumber@#1}{Opt}%
5650 \csgdef{Xtxtbeforenumber@#1}{}%
5651 \csgdef{Xafternumber@#1}{0.5em}%
5652 \newtoggle{Xnonbreakableafternumber@#1}%
5653 \csgdef{Xbeforesymlinenum@#1}{\csuse{Xbeforenumber@#1}}%
5654 \csgdef{Xaftersymlinenum@#1}{\csuse{Xafternumber@#1}}%
5655 \csgdef{Xinplaceofnumber@#1}{1em}%
5656 \global\cslet{Xlemmaseparator@#1}{\rbracket}%
5657 \csgdef{Xbeforelemmaseparator@#1}{0em}%
5658 \csgdef{Xafterlemmaseparator@#1}{0.5em}%
5659 \csgdef{Xinplaceoflemmaseparator@#1}{1em}%
5660 \csgdef{Xbeforenotes@#1}{1.2em \@plus .6em \@minus .6em}%
5661 \csgdef{Xafterrule@#1}{Opt}%
5662
5663 \csgdef{Xtxtbeforenotes@#1}{%
5664 \newtoggle{Xtxtbeforenotes@#1@typeset}%Not directly used by user,
but internal
5665 \newtoggle{Xtxtbeforenotesonlyonce@#1}%
5666
5667 \csgdef{Xmaxhnotes@#1}{0.8\vsizer%
5668 \newtoggle{Xnoteswidthliketwocolumns@#1}%
5669 \csgdef{Xparafootsep@#1}{}%
5670 \csgdef{Xafternote@#1}{1em plus.4em minus.4em}%
5671 \csgdef{Xlinrangeseparator@#1}{\endashchar}%
5672
5673 \csgdef{Xlemmafont@#1}{}%
5674 \csgdef{Xwraplemma@#1}{%
5675 \csgdef{Xwidth@#1}{\hsizer%
5676 \csgdef{Xwraplinenumannotation@#1}{\textsuperscript}%
5677 \csgdef{Xlinenumannotationposition@#1}{after}%
5678 \newtoggle{Xnoidenticallinenumannotation@#1}%
5679 %

```

## XXI.3.2 Create inserts, needed to add notes in foot

As regards inserts, see chapter 15 of *The TeXbook* by D. Knuth.

```

5680 \expandafter\newinsert\csname #1footins\endcsname%
5681 \unless\ifnoledgroup@%
5682 \expandafter\newinsert\csname mp#1footins\endcsname%
5683 \fi%
5684 %

```

## 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.

```

5685 \global\newbool{parapparatus@}\expandafter\newcommand\expandafter
*}{\expandafter\newcommand\csname #1footnote\endcsname[2][\{%
5686 \if@edtext@secondarg@%
5687 \ifledRcol%
5688 \ifcsstring{Xonlyside@#1}{L}{\%
led@error@note@called@onrightside{#1footnote}}}%
5689 \else%
5690 \ifcsstring{Xonlyside@#1}{R}{\%
led@error@note@called@onleftside{#1footnote}}}%
5691 \fi%
5692 \begingroup%
5693 \newcommand{\content}{##2}%
5694 \ifnumberedpar@%
5695 \ifledRcol%
5696 \ifluatex%
5697 \footnotelang@lua[R]%
5698 \fi%
5699 \@ifundefined{xpg@main@language}%if polyglossia
5700 {}%
5701 {\footnotelang@poly[R]}%
5702 \footnoteoptions@{R}{##1}{true}%
5703 \xright@appenditem{%
5704 \ifbool{indtl@innote}%
5705 {\unexpanded{\let\index\nindex}}%
5706 {}%
5707 \ifbool{indtl@notenumber}%
5708 {\unexpanded{\let\index\nindex}}%There is no note
number... so
5709 {}%
5710 \noexpand\Xnote@true%
5711 \noexpand\prepare@Xprenotes{#1}%
5712 \noexpand\prepare@edindex@fornote{\l@d@nums}%
5713 \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.

```

```

5714 \noexpand\setcounter{stanzaR}{\the\c@stanzaR}%Save
stanzaR counter for footnote
5715 \unexpanded{\def\@this@crossref@start}{\theedtext:
start}%
5716 \unexpanded{\def\@this@crossref@end}{\theedtext:end}%
5717 \expandonce{\@beforeinsertofthisedtext}% Internal for
now, no reason to make it public
5718 \noexpand\parse@annot\l@current@annotR|
5719 \noexpand\csuse{v#1footnote}{#1}%
5720 {\l@d@nums}{\expandonce\@tag}{\expandonce\content}}
%
5721 \noexpand\Xnote@false%
5722 \unexpanded{\advance\@edindex@fornote@m@one}%
5723 \ifbool{indtl@innote}%
5724 {\unexpanded{\let\index\orig@@index}}%
5725 {}%
5726 \ifbool{indtl@notenumber}%
5727 {\unexpanded{\let\index\orig@@index}}%
5728 {}%
5729 }\to\inserts@listR
5730 \footnoteoptions@{R}{##1}{false}%
5731 \global\advance\insert@countR \@ne%
5732 \else%
5733 \ifluatex%
5734 \footnotelang@lua%
5735 \fi%
5736 \@ifundefined{xpg@main@language}%if polyglossia
5737 {}%
5738 {\footnotelang@poly}%
5739 \footnoteoptions@{L}{##1}{true}%
5740 \xright@appenditem{%
5741 \ifbool{indtl@innote}%
5742 {\unexpanded{\let\index\nindex}}%
5743 {}%
5744 \ifbool{indtl@notenumber}%
5745 {\unexpanded{\let\index\nindex}}%There is no note
number... so
5746 {}%
5747 \noexpand\Xnote@true%
5748 \noexpand\prepare@Xprenotes{#1}%
5749 \noexpand\prepare@edindex@fornote{\l@d@nums}%
5750 \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.
5751 \ifl@dpairing%
5752 \noexpand\setcounter{stanzaL}{\the\c@stanzaL}%Save
stanzaR counter for footnote
5753 \fi%

```



```

5754 \unexpanded{\def\@this@crossref@start}{\theedtext:
start}%
5755 \unexpanded{\def\@this@crossref@end}{\theedtext:end}%
5756 \expandonce{\@beforeinsertofthisedtext}%Internal for
now, no reason to make it public
5757 \noexpand\parse@annot\l@current@annot|
5758 \noexpand\csuse{v#1footnote}%
5759 {#1}%
5760 {{\l@d@nums}{\expandonce\@tag}{\expandonce\content
}}%
5761 \noexpand\Xnote@false%
5762 \unexpanded{\advance\@edindex@fornote@m@ne}%
5763 \ifbool{indtl@innote}%
5764 {\unexpanded{\let\index\orig@@index}}%
5765 {}%
5766 \ifbool{indtl@notenumber}%
5767 {\unexpanded{\let\index\orig@@index}}%
5768 {}%
5769 }\to\inserts@list
5770 \global\advance\insert@count \@ne%
5771 \footnoteoptions@{L}{#1}{false}%
5772 \fi
5773 \else
5774 \csuse{v#1footnote}{#1}{\{0|0|0|0|0|0|0|0\}}{#1}%
5775 \fi%
5776 \endgroup%
5777 \else%
5778 \led@err@FootnoteNotInSecondArgEdtext{#1}%
5779 \fi%
5780 \ignorespaces%
5781 }
5782 %

```

Create counter used to determine on which page the previous note was called.

```

5783 \expandafter\newcount\csname #1prevpage@num\endcsname%
5784 \expandafter\newcount\csname #1prevpage@numR\endcsname%
5785 %

```

We need to be able to modify reledmac's footnote macros and restore their

```

5786 \global\csletcs{#1@@footnote}{#1footnote}
5787 %

```

### XXI.3.4 Set standard display

```

5788 \Xarrangement@normal{#1}%
5789 %

```

End of for critical footnotes.

```
5790 \fi
5791 %
```

## XXI.4 For familiar footnotes

Familiar footnotes are those which end with letters. We look for the `nofamiliar` option of `reledmac`.

```
5792 \unless\ifnofamiliar@
5793 %
```

### XXI.4.1 Options

```
5794 \newtoggle{parindentX@#1}
5795 \csgdef{wrapcontentX@#1}{}%
5796 \csgdef{hangindentX@#1}{Opt}%
5797 \csgdef{beforeinsertingX@#1}{}%
5798 \csgdef{raggedX@#1}{}%
5799 \csgdef{hsizetwocolX@#1}{0.45 \hsize}%
5800 \csgdef{hsizethreecolX@#1}{.3 \hsize}%
5801 \csgdef{colalignX@#1}{\raggedright}%
5802 \csgdef{notenumfontX@#1}{\normalfont}%
5803 \csgdef{notefontsizeX@#1}{\footnotesize}%
5804 \csgdef{bhooknoteX@#1}{}%
5805 \csgdef{bhookgroupX@#1}{}%
5806 \csgdef{afterruleX@#1}{Opt}
5807 \csgdef{beforenotesX@#1}{1.2em \@plus .6em \@minus .6em}
5808 \csgdef{maxhnotesX@#1}{0.8\vsize}%
5809 \newtoggle{noteswidthliketwocolumnsX@#1}%
5810 \csgdef{parafootsepX@#1}{}%
5811 \csgdef{afternoteX@#1}{1em plus.4em minus.4em}
5812 \csgdef{widthX@#1}{\hsize}%
5813 \csgdef{txtbeforenotesX@#1}{}%
5814 \newtoggle{txtbeforenotesX@#1@typeset}%Not directly used by user,
but internal
5815 \newtoggle{txtbeforenotesonlyonceX@#1}%
5816 % End of for familiar footnotes.
5817 % \subsubsection{Create inserts, needed to add notes in foot}
5818 % As regards inserts, see chapter 15 of the TeXBook by D. Knuth.
5819 % \begin{macrocode}
5820 \expandafter\newinsert\csname footins#1\endcsname%
5821 \unless\ifnoledgroup@%
5822 \expandafter\newinsert\csname mpfootins#1\endcsname%
5823 \fi%
5824 %
```

### 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.

```

5825 \global\expandafter\newcommand\csname footnote#1\endcsname [2] [] {%
5826     \begingroup%
5827     \prepare@prenotesX{#1}%
5828     \newcommand{\content}{##2}%
5829 %
5830 %

```

If we are using the `\csquotes` package, we reset the quotation level.

```

5831 \ifdefined\csq@qllevel%
5832     \csq@qllevel=0\relax%
5833 \fi%
5834 %

```

If we are preparing parallel typesetting, we cannot just increase the footnote counter. Read `reledpar`'s handbook about that (V.1.2 p. 53). If we have a manual footnote mark, use it.

```

5835 \global\expandafter\advance\csname footnote#1@reading\
endcsname by \@ne%
5836 \ifstrempy{##1}{%
5837     \ifboolexpr{bool{!l@dpairing} or bool{!l@dprintingpages} or
bool{!l@dprintingcolumns}}{%
5838         \ifcsdef{footnote#1@reading\the\csname footnote#1@reading
\endcsname=typeset}%
5839             {\setcounter{footnote#1}{\csuse{footnote#1@reading\the
\csname footnote#1@reading\endcsname=typeset}}}%
5840             {\setcounter{footnote#1}{\the\csname footnote#1@reading\
endcsname}}}%
5841     }{%
5842         \stepcounter{footnote#1}%
5843     }%
5844 }{%
5845 %

```

We also have to check consistency with `\onlysideX` setting.

```

5846 \ifledRcol%
5847     \ifcsstring{onlysideX@#1}{L}{\
led@error@note@called@onrightside{footnote#1}}}%
5848     \else%
5849     \ifcsstring{onlysideX@#1}{R}{\
led@error@note@called@onleftside{footnote#1}}}%
5850     \fi%
5851 %

```

And now, the feature not depending of whether we are preparing parallel typesetting

```

5852 \ifstrempy{##1}%
5853     {\protected@csxdef{@thefnmark#1}{\csuse{thefootnote#1}}}%
5854     {\protected@csxdef{@thefnmark#1}{##1}}%
5855 \nottoggle{nomk@}%Nomk is set to true when using \
footnoteXnomk with \parpackage

```

```

5856         {\csuse{@footnotemark#1}}%
5857         {}%
5858         \ifluatex%
5859         \xdef\footnote@luatextextdir{\the\textdir}%
5860         \xdef\footnote@luatexpardir{\the\pardir}%
5861         \fi%
5862         \if@ledgroup%
5863         \ledset@index@fornote{#1}%
5864         \fi%
5865         \csuse{vfootnote#1}{#1}{\expandonce\content}{##1}\
m@mmf@prepare%
5866         \ifbool{indtl@innote}%
5867         {\let\index\orig@@index}%
5868         {}%
5869         \ifbool{indtl@notenumber}%
5870         {\let\index\orig@@index}%
5871         {}%
5872         \endgroup%
5873     }
5874 %

```

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. 53).

```

5875     \newcounter{footnote#1}
5876     \global\expandafter\renewcommand\csname thefootnote#1\endcsname{\
arabic{footnote#1}}
5877     \expandafter\newcount\csname footnote#1@reading\endcsname%
5878 %

```

Create counter used to determine on which page the previous note was called.

```

5879     \expandafter\newcount\csname prevpage#1@num\endcsname%
5880     \expandafter\newcount\csname prevpage#1@numR\endcsname%
5881 %

```

Add `\let\footnoteX\@gobble` to `\no@expands`.

```

5882     \expandafter\gappto\expandafter\no@expands\expandafter{\expandafter\
let\csname footnote#1\endcsname\@gobble}%
5883 %

```

And now, define `\footnoteXmark` and `\footnoteXtext`, equivalent to classical `\footnotemark` and `\footnotetext`.

```

5884     \expandafter\newcommand\csname footnote#1mark\endcsname{%
5885         \begingroup%
5886         \prepare@prenotesX{#1}%
5887         \stepcounter{footnote#1}%
5888         \protected@csxdef{@thefnmark#1}{\csuse{thefootnote#1}}%
5889         \csuse{@footnotemark#1}%

```

```

5890      \m@mmf@prepare%
5891      \endgroup%
5892    }%
5893    \expandafter\newcommand\csname footnote#1text\endcsname[1]{%
5894      \begingroup%
5895        \csuse{vfootnote#1}{#1}{\expandonce{##1}}%
5896      \endgroup%
5897    }%
5898  %

```

Do not forget to initialize the series.

```

5899      \arrangementX@normal{#1}%
5900      \fi
5901    %

```

## XXI.5 The endnotes

Endnotes are commands like `\Xendnote`, where `X` is a series letter. First, we check for the `noend` options.

```

5902      \unless\ifnoend@
5903    %

```

### 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`

```

5904      \expandafter\newwrite\csname l@d@#1end\endcsname%
5905      \expandafter\newif\csname ifl@dend@#1\endcsname%
5906    %

```

### 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.

```

5907      \global\expandafter\newcommandx\csname #1endnote\endcsname[2][1,
5908      usedefault]{%
5909        \bgroup%
5910        \newlinechar='40%
5911        \global\@noneed@Footnotetrue%
5912        \newcommand{\content}{##2}%
5913        \stepcounter{labidx}%

```

```

5914 \expandafter\immediate\expandafter\write\csname l@d#1end\
endcsname{%
5915 \unexpanded{\def\sw@list@inedtext}%
5916 {\expandafter\unexpanded\expandafter{\sw@inthisedtext}}}%
5917 \@percentchar\space%Explicit space, to add a linebreak in the
output file
5918 \noexpand\parse@annot\l@current@annot|\@percentchar\space%
5919 \expandafter\string\csname #1end\endcsname%
5920 {\ifnumberedpar@l@d@nums\fi}%
5921 {\ifnumberedpar@\expandonce\@tag\fi}%
5922 {\expandonce\content}%
5923 {\#1}%
5924 {\unexpanded{##1}}%
5925 {\ifledRcol R\else L\fi}%
5926 {\theedtext}%
5927 \@percentchar%
5928 }%
5929 \egroup%
5930 \ignorespaces%
5931 }%
5932 %

```

### XXI.5.3 Tools

The `\Xtoendnotes` command inserts any arbitrary content into the endnote file. It is an alias of the more generalist `\addtoendnotes`

```

5933 \global\expandafter\newcommand\csname #1toendnotes\endcsname[1]{%
5934 \toendnotes[#1]{##1}%
5935 }%
5936
5937 \expandafter\WithSuffix\expandafter\newcommand\csname #1toendnotes\
endcsname*[1]{%
5938 \toendnotes*[#1]{##1}%
5939 }%
5940
5941
5942 %

```

### 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.

```

5943 \global\cslet{#1end}{\@gobbleseven}
5944
5945 %

```

We need to store the number of times `\doendnotesbysection` is called for one series.

```
5946 \global\expandafter\newcount\csname #1end@bysection\endcsname%
5947 %
```

### XXI.5.5 The options

```
5948 \csgdef{Xendwraplemma@#1}{%
5949 \csgdef{Xendwrapcontent@#1}{}%
5950 \csgdef{Xendtwolines@#1}{}%
5951 \csgdef{Xendmorethantwolines@#1}{}%
5952 \newtoggle{Xendtwolinesbutnotmore@#1}{}%
5953 \newtoggle{Xendtwolinesonlyinsamepage@#1}{}%
5954 \newtoggle{Xendlemmadisablefontselection@#1}{}%
5955 \csgdef{Xendnotenumfont@#1}{\normalfont}%
5956 \csgdef{Xendnotefontsize@#1}{\footnotesize}%
5957 \csgdef{Xendbhooknote@#1}{}%
5958
5959 \csgdef{Xendsublinesep@#1}{\fullstop}%
5960
5961 \csgdef{Xendbeforenumber@#1}{0pt}
5962 \csgdef{Xendafternumber@#1}{0.5em}
5963
5964 \csgdef{Xendboxlinenum@#1}{0pt}%
5965 \csgdef{Xendboxlinenumalign@#1}{L}%
5966
5967 \csgdef{Xendboxstartlinenum@#1}{0pt}%
5968 \csgdef{Xendboxendlinenum@#1}{0pt}%
5969
5970 \csgdef{Xendlemmaseparator@#1}{}%
5971 \csgdef{Xendbeforelemmaseparator@#1}{0em}%
5972 \csgdef{Xendafterlemmaseparator@#1}{0.5em}%
5973 \csgdef{Xendinplaceoflemmaseparator@#1}{0.5em}%
5974
5975 \newtoggle{Xendparagraph@#1}%
5976 \csgdef{Xendafternote@#1}{1em plus.4em minus.4em}%
5977 \csgdef{Xendsep@#1}{}%
5978
5979 \csgdef{Xendinplaceofnumber@#1}{0pt}%
5980 \newtoggle{Xendnonumber@#1}%
5981
5982 \csgdef{Xendhangindent@#1}{0pt}%
5983
5984 \newtoggle{Xendnumberonlyfirstinline@#1}%
5985 \newtoggle{Xendnumberonlyfirstintwolines@#1}%
5986
5987 \newtoggle{Xendlinenumannotationonlyfirst@#1}%
5988 \newtoggle{Xendlinenumannotationonlyfirstintwo@#1}%
5989
```

```

5990 \csgdef{Xendbeforesymlinenum@#1}{\csuse{Xendbeforenumber@#1}}%
5991 \csgdef{Xendaftersymlinenum@#1}{\csuse{Xendafternumber@#1}}%
5992
5993 \csgdef{Xendsymlinenum@#1}{}%
5994 \csgdef{Xendsymlinenumannotation@#1}{}%
5995
5996 \csgdef{Xendboxsymlinenum@#1}{Opt}%
5997
5998 \csgdef{Xendbhooklinenum@#1}{}%
5999 \csgdef{Xendehooklinenum@#1}{}%
6000 \csgdef{Xendbhookinplaceofnumber@#1}{}%
6001 \csgdef{Xendehookinplaceofnumber@#1}{}%
6002
6003 \csgdef{Xendlinerangeseparator@#1}{\endashchar}%
6004
6005 \csgdef{Xendbeforepagenumber@#1}{p.}%
6006 \csgdef{Xendafterpagenumber@#1}{) }%
6007 \csgdef{Xendlineprefixsingle@#1}{}%
6008 \csgdef{Xendlineprefixmore@#1}{}%
6009
6010 \newtoggle{Xendlineflag@#1}
6011
6012 \csgdef{Xendlemmafont@#1}{}%
6013
6014 \csgdef{Xendlinenumannotationposition@#1}{after}%
6015 \csgdef{Xendwraplinenumannotation@#1}{\textsuperscript}%
6016 \newtoggle{Xendnoidenticallinenumannotation@#1}%
6017
6018 \newtoggle{Xendpagenumberonlyfirst@#1}%
6019 \newtoggle{Xendpagenumberonlyfirsttifsingl@#1}%
6020 \newtoggle{Xendpagenumberonlyfirstintwo@#1}%
6021 \csgdef{Xendsympagenum@#1}{}%
6022 \csgdef{Xendinplaceofpagenumber@#1}{Opt}%
6023
6024 \csgdef{Xendtxtbeforenotes@#1}{}%
6025 \newtoggle{Xendfirstnote@#1}%Not a hook, but used to apply
Xendtxtbeforenotes
6026
6027 %

```

End of endnotes declaration

```

6028 \fi%
6029 %

```

Dump series in \@series

```

6030 \listxadd{\@series}{#1}
6031 }
6032 }% End of \newseries
6033 %

```



## XXI.6 Init standards series (A,B,C,D,E)

```
6034 \expandafter\newseries\expandafter{\default@series}
6035 %
```

## XXII Setting series display

### XXII.1 Change series order

**\seriesatbegin** `\seriesatbegin{<s>}` changes the order of series, to put the series `<s>` at the beginning of the list. The series can be the result of a command.

```
6036 \newcommand{\seriesatbegin}[1]{%
6037   \StrDel{\@series}{#1}[\@series]%
6038   \edef\@new{%
6039     \listadd{\@new}{#1}%
6040     \listadd{\@new}{\@series}%
6041     \xdef\@series{\@new}%
6042   }
6043   %
```

**\seriesatend** And `\seriesatend` moves the series to the end of the list.

```
6044 \newcommand{\seriesatend}[1]{%
6045   \StrDel{\@series}{#1}[\@series]%
6046   \edef\@new{%
6047     \listadd{\@new}{\@series}%
6048     \listadd{\@new}{#1}%
6049     \xdef\@series{\@new}%
6050   }
6051   %
```

### XXII.2 Test series order

**\ifseriesbefore** `\ifseriesbefore{<seriesA>}{<seriesB>}{<true>}{<false>}` expands to `<true>` if `<seriesA>` is printed before `<seriesB>`, or to `<false>` otherwise.

```
6052 \newcommand{\ifseriesbefore}[4]{%
6053   \StrPosition{\@series}{#1}[\@first]%
6054   \StrPosition{\@series}{#2}[\@second]%
6055   \ifnumgreater{\@second}{\@first}{#3}{#4}%
6056 }
6057 %
```

#### XXII.2.1 Get the first series

In some specific case, we need to know the first series of the list of series.

```

\@getfirstseries\newcommand{\@getfirstseries}{%
6059 \ifdefempty{\@series}%
6060 {\xdef\@firstseries{}}%
6061 {\StrChar{\@series}{1}[\@firstseries]}%
6062 }%
6063 %

```

## 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`.

```

6064 \newcommandx{\settoggle@series}[5][4,5,usedefault]{%
6065   \def\do##1{%
6066     \global\settoggle{#2@##1}{#3}%
6067     \ifstrequal{#4}{critical}{
6068       \csuse{Xarrangement@}\csuse{series@display##1}}{##1}%
6069     }{}
6070     \ifstrequal{#4}{familiar}{
6071       \csuse{arrangementX@}\csuse{series@displayX##1}}{##1}%
6072     }{}
6073   }%
6074   \ifstreempty{#1}{%
6075     \dolistloop{\@series}%
6076     \ifstreempty{#5}{}%
6077       \docsvlist{#5}%
6078     }
6079   }%
6080   {%
6081     \docsvlist{#1}%
6082   }%
6083 }
6084 %

```

**\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`.

```

6085 \newcommandx{\setcommand@series}[5][4,5,usedefault]{%
6086   \def\do##1{
6087     \csgdef{#2@##1}{#3}
6088     \ifstrequal{#4}{critical}{%
6089       \csuse{Xarrangement@}\csuse{series@display##1}}{##1}%
6090     }{}
6091     \ifstrequal{#4}{familiar}{%
6092       \csuse{arrangementX@}\csuse{series@displayX##1}}{##1}%
6093     }{}%
6094   }%
6095   \ifstreempty{#1}{%
6096     \dolistloop{\@series}%

```

```

6097      \ifstrempy{#5}{}{%
6098          \docsvlist{#5}
6099      }
6100  }%
6101  {%
6102      \docsvlist{#1}%
6103  }%
6104  }%
6105  %

```

### 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><pseudoseries>`, like `\Xtwolinesappref`.

```

6106 \newcommand{\newhookcommand@series}[2][2,usedefault]{%
6107   \global\expandafter\newcommand\expandafter*\csname #1\endcsname[2][]{%
6108     \setcommand@series{##1}{#1}{##2}[] [#2]%
6109   }%
6110   \ifstrempy{#2}{-}{%
6111     \def\do##1{%
6112       \global\expandafter\newcommand\expandafter*\csname #1##1\endcsname
6113       [1]{%
6114         \csuse{#1}[##1]{###1}%
6115       }%
6116     }%
6117     \docsvlist{#2}%
6118   }%
6119 }%

```

`\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><pseudoseris>`, like `\Xtwolinesbutnotmoreappref`.

```

6120 \newcommandx{\newhooktoggle@series}[2][2,usedefault]{%
6121   \global\expandafter\newcommandx\expandafter*\csname #1\endcsname[2][1,2={
true},usedefault]{%
6122     \settoggle@series{##1}{##2}[][#2]%
6123   }%
6124   \ifstrepty{#2}{}%
6125   \def\do##1{%
6126     \global\expandafter\newcommand\expandafter*\csname #1##1\endcsname{%
6127       \csuse{#1}[##1]%
6128     }%

```

```

6129 }%
6130 \docsvlist{#2}%
6131 }%
6132 }
6133 %

```

**\newhooktoggle@series@reload** `\newhookcommand@toggle@reload` does the same thing as `\newhooktoggle@series` but the commands created by this macro also reload the series arrangement, depending of type os notes

```

6134 \newcommand{\newhooktoggle@series@reload}[2]{%
6135 \global\expandafter\newcommandx\expandafter*\csname #1\endcsname[2][1,2={
true},usedefault]{%
6136 \settoggle@series{##1}{#1}{##2}[#2]%
6137 }%
6138 }%
6139 %

```

**\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.

```

6140 \newcommand{\newhookcommand@series@reload}[2]{%
6141 \global\expandafter\newcommand\expandafter*\csname #1\endcsname[2][]{%
6142 \setcommand@series{##1}{#1}{##2}[#2]%
6143 }%
6144 }
6145 %

```

### 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.

```

6146 \unless\ifnocritical@
6147 \newhookcommand@series{Xwrapcontent}%
6148 \newhookcommand@series{Xbeforeinserting}%
6149 \newhookcommand@series{Xlemmafont}%
6150 \newhookcommand@series{Xwraplemma}%
6151 \newhooktoggle@series{Xparindent}
6152 \newhookcommand@series{Xhangindent}
6153 \newhookcommand@series{Xragged}
6154 \newhookcommand@series{Xhsizetwocol}
6155 \newhookcommand@series{Xhsizethreecol}
6156 \newhookcommand@series{Xcolalign}%
6157 \newhookcommand@series{Xnotenumfont}
6158 \newhookcommand@series{Xbhooknote}
6159 \newhookcommand@series@reload{Xbhookgroup}{critical}
6160 \newhookcommand@series{Xboxsymlinenum}%

```

```

6161 \newhookcommand@series{Xsymlinenum}
6162 \newhookcommand@series{Xsymlinenumannotation}%
6163 \newhookcommand@series{Xbeforenumber}
6164 \newhookcommand@series{Xtxtbeforenumber}
6165 \newhookcommand@series{Xafternumber}
6166 \newhookcommand@series{Xbeforesymlinenum}
6167 \newhookcommand@series{Xaftersymlinenum}
6168 \newhookcommand@series{Xinplaceofnumber}
6169 \newhookcommand@series{Xlemmaseparator}
6170 \newhookcommand@series{Xbeforelemmaseparator}
6171 \newhookcommand@series{Xafterlemmaseparator}
6172 \newhookcommand@series{Xinplaceoflemmaseparator}
6173 \newhookcommand@series{Xtxtbeforenotes}
6174 \newhooktoggle@series{Xtxtbeforenotesonlyonce}%
6175 \newhookcommand@series@reload{Xafterrule}{critical}
6176 \newhooktoggle@series{Xnumberonlyfirstinline}
6177 \newhooktoggle@series{Xnumberonlyfirstintwolines}
6178 \newhooktoggle@series{Xlinenumannotationonlyfirst}%
6179 \newhooktoggle@series{Xlinenumannotationonlyfirstintwo}%
6180 \newhooktoggle@series{Xgroupbyline}%
6181 \newhooktoggle@series{Xgroupbylineseparatetwolines}%
6182 \newhooktoggle@series{Xnonumber}
6183 \newhooktoggle@series{Xpstart}
6184 \newhooktoggle@series{Xpstarteverytime}%
6185
6186 \newhooktoggle@series{Xstanza}%
6187 \newhookcommand@series{Xstanzaseparator}%
6188
6189 \newhooktoggle@series{Xonlypstart}
6190 \newhooktoggle@series{Xnonbreakableafternumber}
6191 \newhooktoggle@series{Xlemmadisablefontselection}
6192 \newhookcommand@series@reload{Xmaxhnotes}{critical}
6193 \newhookcommand@series@reload{Xbeforenotes}{critical}
6194 \newhooktoggle@series@reload{Xnoteswidthliketwocolumns}{critical}%
6195 \newhookcommand@series@reload{Xnotefontsize}{critical}
6196
6197 \newhookcommand@series{Xboxlinenum}%
6198 \newhookcommand@series{Xboxlinenumalign}%
6199
6200 \newhookcommand@series{Xboxstartlinenum}%
6201 \newhookcommand@series{Xboxendlinenum}%
6202
6203 \newhookcommand@series{Xafternote}%
6204 \newhookcommand@series{Xparafootsep}
6205 \newhookcommand@series@reload{Xwidth}{critical}%
6206
6207 \ifundef{\Xhsize}%
6208 {%
6209   \newcommandx{\Xhsize}[2][1,usedefault]{%
6210     \led@warning@Xhsize@deprecated%

```

```

6211 \Xwidth[#1]{#2}%
6212 }%
6213 }%
6214 {}%
6215 \fi
6216 %

```

Now, hooks for critical notes which also apply to crossreferencing and line numbering at the sides of the page.

```

6217 \newhooktoggle@series{Xlineflag}[appref,SEref]
6218 \newhookcommand@series{Xtwolines}[appref,SEref]
6219 \newhookcommand@series{Xmorethantwolines}[appref,SEref]
6220 \newhookcommand@series{Xsublinesep}[appref,SEref,side]%
6221 \newhookcommand@series{Xpagelinesep}[appref,SEref,side]%
6222 \newhooktoggle@series{Xtwolinesbutnotmore}[appref,SEref]
6223 \newhooktoggle@series{Xtwolinesonlyinsamepage}[appref,SEref]
6224 \newhookcommand@series{Xlinrangeseparator}[appref,SEref]
6225 \newhookcommand@series{Xlinenumannotationposition}[side,appref,SEref]%
6226 \newhookcommand@series{Xwraplinenumannotation}[side,ref,appref,SEref]%
6227 \newhooktoggle@series{Xnoidenticallinenumannotation}[side,ref,appref,SEref]
6228 %

```

### XXII.3.5 Options for familiar notes

Before generating the optional commands for familiar notes, we check the `\nofamiliar` option.

```

6229 \unless\ifnofamiliar@
6230 \newhookcommand@series{wrapcontentX}%
6231 \newhookcommand@series{beforeinsertingX}%
6232 \newhooktoggle@series{parindentX}
6233 \newhookcommand@series{hangindentX}
6234 \newhookcommand@series{raggedX}
6235 \newhookcommand@series{hsizetwocolX}
6236 \newhookcommand@series{hsizethreecolX}
6237 \newhookcommand@series{colalignX}%
6238 \newhookcommand@series{notenumfontX}
6239 \newhookcommand@series{bhooknoteX}
6240 \newhookcommand@series@reload{bhookgroupX}{familiar}
6241 \newhookcommand@series@reload{beforenotesX}{familiar}
6242 \newhookcommand@series@reload{maxhnotesX}{familiar}
6243 \newhooktoggle@series@reload{noteswidthliketwocolumnsX}{familiar}%
6244 \newhookcommand@series@reload{afterruleX}{familiar}
6245 \newhookcommand@series@reload{notefontsizeX}{familiar}
6246 \newhookcommand@series{afternoteX}
6247 \newhookcommand@series{parafootsepX}
6248 \newhookcommand@series{txtbeforenotesX}%
6249 \newhooktoggle@series{txtbeforenotesonlyonceX}%

```

```

6250 \newhookcommand@series@reload{widthX}{familiar}%
6251 \ifundef{\hsizeX}%
6252 {%
6253   \newcommandx{\hsizeX}[2][1,usedefault]{%
6254     \led@warning@hsizeX@deprecated%
6255     \widthX[#1]{#2}%
6256   }%
6257 }%
6258 {}%
6259 \fi
6260 %

```

### 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.

```

6261 \unless\ifnoend@
6262   \newhookcommand@series{Xendwraplemma}
6263   \newhookcommand@series{Xendwrapcontent}
6264   \newhookcommand@series{Xendnotenumfont}
6265   \newhookcommand@series{Xendlemmafont}%
6266   \newhookcommand@series{Xendbhooknote}
6267
6268   \newhookcommand@series{Xendboxlinenum}%
6269   \newhookcommand@series{Xendboxlinenumalign}%
6270
6271   \newhookcommand@series{Xendboxstartlinenum}%
6272   \newhookcommand@series{Xendboxendlinenum}%
6273
6274   \newhookcommand@series{Xendnotefontsize}
6275   \newhooktoggle@series{Xendlemmadisablefontselection}
6276   \newhookcommand@series{Xendlemmaseparator}
6277   \newhookcommand@series{Xendbeforelemmaseparator}
6278   \newhookcommand@series{Xendafterlemmaseparator}
6279   \newhookcommand@series{Xendinplaceoflemmaseparator}
6280
6281   \newhookcommand@series{Xendbeforenumber}%
6282   \newhookcommand@series{Xendafternumber}%
6283
6284   \newhooktoggle@series{Xendparagraph}
6285   \newhookcommand@series{Xendafternote}
6286   \newhookcommand@series{Xendsep}
6287
6288   \newhookcommand@series{Xendinplaceofnumber}%
6289   \newhooktoggle@series{Xendnonumber}%
6290
6291   \newhooktoggle@series{Xendnumberonlyfirstinline}%
6292   \newhooktoggle@series{Xendnumberonlyfirstintwolines}%
6293

```



```

6294 \newhooktoggle@series{Xendlinenumannotationonlyfirst}%
6295 \newhooktoggle@series{Xendlinenumannotationonlyfirstintwo}%
6296
6297 \newhookcommand@series{Xendsymmlinenum}%
6298 \newhookcommand@series{Xendbeforesymmlinenum}%
6299 \newhookcommand@series{Xendaftersymmlinenum}%
6300 \newhookcommand@series{Xendboxsymmlinenum}%
6301
6302 \newhookcommand@series{Xendsymmlinenumannotation}%
6303
6304 \newhookcommand@series{Xendbhooklinenumber}%
6305 \newhookcommand@series{Xendahooklinenumber}%
6306 \newhookcommand@series{Xendbhookinplaceofnumber}%
6307 \newhookcommand@series{Xendahookinplaceofnumber}%
6308
6309 \newhookcommand@series{Xendhangindent}%
6310
6311 \newhooktoggle@series{Xendpagenumberonlyfirst}%
6312 \newhooktoggle@series{Xendpagenumberonlyfirstifsingle}%
6313 \newhooktoggle@series{Xendpagenumberonlyfirstintwo}%
6314 \newhookcommand@series{Xendsympagenum}%
6315 \newhookcommand@series{Xendinplaceofpagenumber}%
6316
6317 \newhookcommand@series{Xendtxtbeforenotes}%
6318
6319 \fi
6320 \newhooktoggle@series{Xendlineflag}[apprefwithpage,SErefwithpage]
6321 \newhookcommand@series{Xendtwolines}[apprefwithpage,SErefwithpage]
6322 \newhookcommand@series{Xendmorethantwolines}[apprefwithpage,SErefwithpage]
6323 \newhooktoggle@series{Xendtwolinesbutnotmore}[apprefwithpage,SErefwithpage]
6324 \newhooktoggle@series{Xendtwolinesonlyinsamepage}[apprefwithpage,
SErefwithpage]
6325 \newhookcommand@series{Xendlinerangeseparator}[apprefwithpage,SErefwithpage
]
6326 \newhookcommand@series{Xendbeforepagenumber}[apprefwithpage,SErefwithpage,
SErefonlypage]
6327 \newhookcommand@series{Xendafterpagenumber}[apprefwithpage,SErefwithpage]
6328 \newhookcommand@series{Xendlineprefixsingle}[apprefwithpage,SErefwithpage]
6329 \newhookcommand@series{Xendlineprefixmore}[apprefwithpage,SErefwithpage]
6330 \newhookcommand@series{Xendsublinesep}[apprefwithpage,SErefwithpage]
6331
6332 \newhookcommand@series{Xendlinenumannotationposition}[apprefwithpage,
SErefwithpage]%
6333 \newhookcommand@series{Xendwraplinenumannotation}[apprefwithpage,
SErefwithpage]%
6334 \newhooktoggle@series{Xendnoidenticallinenumannotation}[apprefwithpage,
SErefwithpage]%
6335 %

```

## XXII.4 Hooks for a particular footnote

`\newhooktoggle@specific` `\newhooktoggle@specific` is a generic command to create boolean hook specific to a note.

```

6336 \newcommand{\newhooktoggle@specific}[1]{%
6337   \newtoggle{#1}%
6338   \listgadd{\hooktoggle@specific}{#1}%
6339   \define@key[mac]{truefootnoteoption}{#1}[]{\global\settoggle{#1}{true}}%
When enabling footnote option
6340   \define@key[mac]{falsefootnoteoption}{#1}[]{\global\settoggle{#1}{false
}}
6341 }
6342 %

```

`\newhookarg@specific` `\newhookarg@specific` is a generic command to create argumen hook specific to a note.

```

6343 \newcommand{\newhookarg@specific}[1]{%
6344   \listgadd{\hookarg@specific}{#1}%
6345   \define@key[mac]{truefootnoteoption}{#1}{\csgdef{#1}{##1}}%When enabling
footnote option
6346   \define@key[mac]{falsefootnoteoption}{#1}{\global\csundef{#1}}%When
disabling footnote option
6347 }
6348 %

```

`\hooktoggle@specific` `\add@hooktoggle@specific@to@cs` 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.

```

6349 \def\hooktoggle@specific{%
6350 \newcommand{\add@hooktoggle@specific@to@cs}[1]{%
6351   \def\do##1{%
6352     \iftoggle{##1}{%
6353       \ifcsdef{#1}{%
6354         \csgappto{#1}{\toggletrue{##1}}%
6355       }{%
6356         \csgdef{#1}{\toggletrue{##1}}%
6357       }%
6358     }{%
6359       \ifcsdef{#1}{%
6360         \csgappto{#1}{\togglefalse{##1}}%
6361       }{%
6362         \csgdef{#1}{\togglefalse{##1}}%
6363       }%
6364     }%
6365   }%
6366   \dolistloop{\hooktoggle@specific}%

```

```
6367 }%
6368 %
```

`\hookarg@specific` The same, but for optional argument of critical footnotes with assigned value.  
`dd@hookarg@specific@to@cs`

```
6369 \def\hookarg@specific{}%
6370 \newcommand{\add@hookarg@specific@to@cs}[1]{%
6371   \def\do##1{%
6372     \ifcsvoid{##1@}{}%
6373     \ifcsdef{##1}{%
6374       \csxappto{##1}{%
6375         \noexpand\csdef%
6376           {##1@}{\csname##1@\endcsname}%
6377       }%
6378     }{%
6379       \csxdef{##1}{%
6380         \noexpand\csdef%
6381           {##1@}{\csname##1@\endcsname}%
6382       }%
6383     }%
6384   }%
6385 }%
6386 \dolistloop{\hookarg@specific}%
6387 }%
6388 %
```

And now, we define some hooks specific to a note.

```
6389 \newhooktoggle@specific{fulllines}%
6390 \newhooktoggle@specific{nonum}
6391 \newhooktoggle@specific{nosep}
6392 \newhooktoggle@specific{noprefix}%
6393 \newhooktoggle@specific{prefixmore}%
6394 \newhookarg@specific{linrangesep}
6395 %
```

`linrangesep@` `\linrangesep@` is defined by the option `linrangesep` of critical notes to change temporarily the line range separator for a specific line. As we have to define it before typesetting the line and undefine it after, we use the family of `xkeyval` package's key.

```
6396 %
```

`\nomk@` `\nomk@` toggle is used by `reledpar` to remove the footnote mark in the text when using `\footnotexmk`. Read `reledpar` handbook.

```
6397 \newtoggle{nomk@}%
6398 %
```

## XXII.5 Alias

`\Xnolemmaseparator` `\Xnolemmaseparator[⟨series⟩]` is just an alias for `\Xlemmaseparator[⟨series⟩]{}`.

```
6399 \newcommand*{\Xnolemmaseparator}[1][1]{\Xlemmaseparator[#1]{}}
6400 %
```

## 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.

```
6401 \newcommand*{\l@ddoxtrafeet}{%
6402   \IfStrEq{familiar-critical}{\@fnpos}
6403     {\do@feetX\do@Xfeet}%
6404   {%
6405     \IfStrEq{critical-familiar}{\@fnpos}%
6406       {\do@Xfeet\do@feetX}%
6407     {%
6408       \setbox\@outputbox \vbox{%
6409         \unvbox\@outputbox%
6410         \do@feet@custom@order}{\@fnpos}%
6411       }%
6412     }%
6413   }%
6414 }%
6415
6416 %
```

`\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.

```
6417 \newcommand{\do@feet@custom@order}[2]{%
6418   \def\do##1{%
6419     \edef\@notesseries{\@firstoftwo##1}%
6420     \edef\@notetype{\@secondoftwo##1}%
6421     \ifdefstring{\@notetype}{critical}%
6422       {\csuse{#1append@Xnotes}{\@notesseries}}%
6423   }
```

```

6423     {\ifdefstring{\@@notetype}{familiar}%
6424      {\csuse{#1append@notesX}{\@@notesseries}}}%
6425      {}%
6426    }%
6427  }%
6428  \expandafter\docsvlist\expandafter{#2}%
6429 }%
6430 %

```

**\do@Xfeet** \do@Xfeet is the code extending \@makecol to cater to the extra critical feet.

```

6431 \newcommand*{\do@Xfeet}{%
6432   \setbox\@outputbox \vbox{%
6433     \unvbox\@outputbox
6434     \@opXfeet}}
6435 %

```

**\@opXfeet** The extra critical feet to be added to the output. . A macro which appends critical

**\append@Xnotes** notes to the output's routine, also adding vertical space before notes

```

6436 \print@Xnotes \newcommand{\append@Xnotes}[1]{%
6437   \ifvoid\csuse{#1footins}\else%
6438     \global\skip\csuse{#1footins}=\csuse{Xbeforenotes@#1}%
6439     \global\advance\skip\csuse{#1footins} by\csuse{Xafterrule@#1}%
6440     \print@Xnotes{#1}%
6441   \fi%
6442 }%
6443 %

```

The normal way to add one series, \print@Xnotes, is replaced by reledpar when using \Pages.

```

6444 \newcommand\print@Xnotes[1]{%
6445   \xdef\@currentseries{#1}%
6446   \csuse{#1footstart}{#1}%
6447   \csuse{#1footgroup}{#1}%
6448 }%
6449 %

```

We print all series of notes by looping on them. We check before printing them that they are not voided.

```

6450 \newcommand*{\@opXfeet}{%
6451   \unless\ifnocritical@%
6452     \gdef\firstXseries@{}%
6453     \def\do##1{%
6454       \append@Xnotes{##1}%
6455     }%
6456     \dolistloop{\@series}%
6457   \fi%
6458 }%
6459 %

```

`\l@ddodoreinxtrafeet` `\l@ddodoreinxtrafeet` is the code catering for the extra footnotes within `\@reinserts`. We use the same category and ordering as in `\l@ddoxtrafeet`.

```

6460 \newcommand*{\l@ddodoreinxtrafeet}{%
6461   \IfStrEq{familiar-critical}{\@fnpos}
6462     {\@doreinfeetX\X@doreinfeet}%
6463   {%
6464     \IfStrEq{critical-familiar}{\@fnpos}%
6465       {\X@doreinfeet\@doreinfeetX}%
6466       {\@doreinfeetX\X@doreinfeet}%
6467   }%
6468 }
6469
6470 %

```

`\X@doreinfeet` `\X@doreinfeet` is the code for catering for the extra critical footnotes within `\@reinserts`.

```

6471 \newcommand*{\X@doreinfeet}{%
6472   \unless\ifnocritical@%
6473     \def\do##1{%
6474       \ifvoid\csuse{##1footins}\else%
6475         \insert\csuse{##1footins}{\unvbox\csuse{##1footins}}%
6476       \fi}%
6477     \dolistloop{\@series}
6478   \fi%
6479 }
6480
6481 %

```

`\print@notesX` We have to add all the new kinds of familiar footnotes to the output routine. A macro which appends the familiar footnotes of one series onto the output routine, also adding vertical skip before notes.

`\append@notesX`

`\do@feetX`

`\@doreinfeetX`

```

6482 \newcommand{\append@notesX}[1]{%
6483   \ifvoid\csuse{footins#1}\else%
6484     \global\skip\csuse{footins#1}=\csuse{beforenotesX@#1}%
6485     \global\advance\skip\csuse{footins#1} by\csuse{afterruleX@#1}%
6486     \print@notesX{#1}%
6487   \fi%
6488 }%
6489 %

```

The normal way to print one series of notes. `\print@Xnotes` is replaced by `reledpar` when using `\Pages`.

```

6490 \newcommand\print@notesX[1]{%
6491   \xdef\@currentseries{#1}%
6492   \csuse{footstart#1}{#1}%
6493   \csuse{footgroup#1}{#1}%
6494 }%
6495 %

```

We print all the series of notes by looping over them. We check before printing them that they are not voided.

```

6496 \newcommand*{\do@feetX}{%
6497   \unless\ifnofamiliar%
6498   \gdef\firstseriesX@{}%
6499   \setbox\@outputbox \vbox{%
6500     \unvbox\@outputbox%
6501     \def\do##1{%
6502       \append@notesX{##1}%
6503     }%
6504     \dolistloop{\@series}}%
6505   \fi%
6506 }%
6507
6508 \newcommand{\@doreinfeetX}{%
6509   \unless\ifnofamiliar%
6510   \def\do##1{%
6511     \ifvoid\csuse{footins##1}\else
6512       \insert%
6513         \csuse{footins##1}
6514         {\unvbox\csuse{footins##1}}%
6515     \fi%
6516   }%
6517   \dolistloop{\@series}%
6518   \fi%
6519 }%
6520
6521 %

```

## 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.)

```

6522 \@ifclassloaded{memoir}{%
6523   %
6524   memoir is loaded so we use memoir's built in hooks.
6525   \g@addto@macro{\m@mdoextrafeet}{\l@ddoxtrafeet}%
6526   \g@addto@macro{\m@mdodoreinextrafeet}{\l@ddodoreinextrafeet}%
6527   }%

```

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.

```

6528 \ifboolexpr{%
6529   test{\@ifpackageloaded{fancyhdr}}%
6530   and test {\ifdef{\latex@makecol}}%
6531 }{%
6532   \patchcmd%
6533     {\latex@makecol}%
6534     {\xdef\@freelist{\@freelist\@midlist}}%
6535     {\xdef\@freelist{\@freelist\@midlist}\l@ddoxtrafeet}%
6536     {}%
6537     {\led@error@fail@patch@@@makecol}}%
6538   {%
6539     \patchcmd%
6540       {\@makecol}%
6541       {\xdef\@freelist{\@freelist\@midlist}}%
6542       {\xdef\@freelist{\@freelist\@midlist}\l@ddoxtrafeet}%
6543       {}%
6544       {\led@error@fail@patch@@@makecol}}%
6545     }%
6546
6547   \patchcmd%
6548     {\@reinserts}%
6549     {\ifvbox}%
6550     {\l@ddodoreinextrafeet\ifvbox}%
6551     {}%
6552     {\led@error@fail@patch@@@reinserts}}%
6553 }
6554
6555 %

```

It turns out that \@docclearpage also needs modifying.

**\if@led@nofoot** We have to check if there are any leftover feet.

```

6556 \newif\if@led@nofoot
6557
6558 %

```

```

6559 \@ifclassloaded{memoir}{%
6560 %

```

If the memoir class is loaded, we hook into its modified \@docclearpage.

```

\@mem@extranofeet \g@addto@macro{\@mem@extranofeet}{%%
6562   \def\do#1{%
6563     \unless\ifnocritical@%
6564       \ifvoid\csuse{#1footins}\else\@mem@nofootfalse\fi%
6565       \fi%
6566     \unless\ifnofamiliar@%

```



```

6567 \ifvoid\csuse{footins#1}\else\@mem@nofootfalse\fi%
6568 \fi%
6569 }
6570 \dolistloop{\@series}%
6571 }%
6572 }%
6573 %

```

As memoir is not loaded we have patch \@doclearpage.

```

\@led@testifnofoot 74 \newcommand*{\@led@testifnofoot}{%
\@doclearpage 75 \@led@nofoottrue%
6576 \ifvoid\footins\else%
6577 \@led@nofootfalse%
6578 \fi%
6579 \def\do##1{%
6580 \unless\ifnocritical@%
6581 \ifvoid\csuse{##1footins}\else%
6582 \@led@nofootfalse%
6583 \fi%
6584 \fi%
6585 \unless\ifnofamiliar@%
6586 \ifvoid\csuse{footins##1}\else%
6587 \@led@nofootfalse%
6588 \fi%
6589 \fi%
6590 }%
6591 \dolistloop{\@series}%
6592 }%
6593
6594 \pretocmd%
6595 {\@doclearpage}%
6596 {\@led@testifnofoot}%
6597 {}%
6598 {\led@error@fail@patch@@doclearpage}%
6599
6600 \patchcmd%
6601 {\@doclearpage}%
6602 {\ifvoid\footins}%
6603 {\if@led@nofoot}%
6604 {}%
6605 {\led@error@fail@patch@@doclearpage}%
6606
6607 }
6608
6609 %

```

## 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. 130). The `sameparallelpagenumbers` 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.

```

6610
6611 \newcommand{\par@patch@thepage}{%
6612   \ifboolexpr{%
6613     bool{sameparallelpagenumbers}%
6614     or bool{prevpgnotnumbered}%
6615   }%
6616   {%
6617     \patchcmd{\thepage}%
6618       {page}{par@page}%
6619       {}%
6620       {\led@error@fail@patch@thepage}%
6621   }{}%
6622 }%
6623
6624 \newcommand{\par@patch@pagenumbering}{%
6625   \setcounter{par@page}{1}%
6626   \par@patch@thepage%
6627 }%
6628
6629 \ifl@dmemoir%
6630   \apptocmd{\@mempnum}%
6631     {\par@patch@pagenumbering}%
6632     {}%

```

```

6633     {\led@error@fail@patch@memnum}%
6634
6635 \else%
6636   \apptocmd{\pagenumbering}%
6637     {\par@patch@pagenumbering}%
6638     {}%
6639     {\led@error@fail@patch@pagenumbering}%
6640 \fi%
6641
6642 \AtBeginDocument{\par@patch@thepage}%
6643 %

```

**\@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.

```

6644 \AtBeginDocument{%
6645   \apptocmd{\@outputpage}{%
6646     \ifsameparallelpagelapnumber%
6647       \ifl@dprintingpages%
6648         \ifodd\c@page\else%
6649           \stepcounter{par@page}%
6650         \fi%
6651       \else%
6652         \stepcounter{par@page}%
6653       \fi%
6654     \else%
6655       \stepcounter{par@page}%
6656     \fi%
6657   }%
6658   {}%
6659   {\led@error@fail@patch@outputpage}%
6660 }%
6661 %

```

**\thepar@page** The par@page counter.

```

6662 \newcounter{par@page}%
6663 \setcounter{par@page}{1}%
6664 %

```

## 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.

```
6665 \list@create{\labelref@list}
6666 %
```

**\zz@@@** Two convenience macros to zero three / four labeling counters in one go.

```
\zz@@@
6667 \newcommand*\zz@@@{-{000|000|000}}% Set three counters to zero in one go
6668 \newcommand*\zz@@@{-{000|000|000|000}}% Set four counters to zero in one go
6669 %
6670 %
```

**\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>36</sup>

This version of the original `edmac \label` uses `\@bsphack` and `\@esphack` to eliminate extra space problems and also use the  $\TeX$  write methods for the `.aux` file.

Jesse Billett<sup>37</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.

```
6671 \newcommand*\edlabel}[1]{%
6672   \leavevmode%
6673   \@bsphack%
6674   \ifboolexpr{bool{ledRcol} or bool{ledRcol@}}{%
6675     \ifXnote@%
6676     \protected@write\@auxout{%
6677       {\string\l@dmake@labelsR\space\thepage|\l@dparsedstartline|\l@dparsedstartsub||\annot@start|\the\c@pstartR|{#1}}%
6678       \ifdef{\hypertarget}%
6679       {\Hy@raisedlink{\hypertarget{#1}{}}}%
6680       {}%
6681     \else%
6682     \write\linenum@outR{\string\@lab}%
6683     \ifx\labelref@listR\empty%
6684     \xdef\label@refs{\zz@@@}%
6685     \else%
6686     \gl@p\labelref@listR\to\label@refs%
6687     \fi%
6688   }
```

<sup>36</sup>The remaining macros in this section were kindly revised by Wayne Sullivan, who substantially improved their efficiency and flexibility.

<sup>37</sup>(jdb43@cam.ac.uk) via the ctt thread ‘ledmac cross referencing’, 25 August 2003.

```

6688 \ifvmode%
6689 \advancelabel@refs%
6690 \fi%
6691 %

```

Use code from the kernel `\label` command to write the correct page number. Also define an `hypertarget` if `hyperref` package is loaded.

```

6692 \protected@write\@auxout{}%
6693 {\string\l@dmake@labelsR\space\thepage|\label@refs|\the\c@pstartR
|{#1}}%
6694 \ifdef\hypertarget%
6695 {\Hy@raisedlink{\hypertarget{#1}{}}}%
6696 {}%
6697 \fi%
6698 }{%
6699 \ifXnote@%
6700 \ifl@dpairing%pstart or pstartL?
6701 \protected@write\@auxout{}%
6702 {\string\l@dmake@labels\space\thepage|\l@dparsedstartline|\
l@dparsedstartsub||\annot@start|\the\c@pstartL|{#1}}%
6703 \ifdef\hypertarget%
6704 {\Hy@raisedlink{\hypertarget{#1}{}}}%
6705 {}%
6706 \else%
6707 \protected@write\@auxout{}%
6708 {\string\l@dmake@labels\space\thepage|\l@dparsedstartline|\
l@dparsedstartsub||\annot@start|\the\c@pstart|{#1}}%
6709 \ifdef\hypertarget%
6710 {\Hy@raisedlink{\hypertarget{#1}{}}}%
6711 {}%
6712 \fi%
6713 \else%
6714 \write\linenum@out{\string\@lab}%
6715 \ifx\labelref@list\empty%
6716 \xdef\label@refs{\zz@@@}%
6717 \else%
6718 \gl@p\labelref@list\to\label@refs%
6719 \fi%
6720 \ifvmode%
6721 \advancelabel@refs%
6722 \fi%
6723 \ifl@dpairing%Pstart or PstartL?
6724 \protected@write\@auxout{}%
6725 {\string\l@dmake@labels\space\thepage|\label@refs|\the\c@pstartL
|{#1}}%
6726 \ifdef\hypertarget%
6727 {\Hy@raisedlink{\hypertarget{#1}{}}}%
6728 {}%
6729 \else%
6730 \protected@write\@auxout{}%

```

```

6731      {\string\l@dmake@labels\space\thepage|\label@refs|\the\c@pstart
|{#1}}}%
6732      \ifdef\hypertarget}%
6733      {\Hy@raisedlink{\hypertarget{#1}{}}}%
6734      }%
6735      \fi%
6736      \fi%
6737      }%
6738      \@esphack}%
6739
6740      %

```

`\advancelabel@refs` 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.

```

6741 \newcounter{line}%
6742 \newcounter{subline}%
6743 \newcounter{absline}%
6744 \newcommand{\advancelabel@refs}{%
6745     \setcounter{line}{\expandafter\labelrefsparseline\label@refs}%
6746     \stepcounter{line}%
6747     \setcounter{absline}{\expandafter\labelrefsparseabsline\label@refs}%
6748     \stepcounter{absline}%
6749     \ifsublines@%
6750         \setcounter{subline}{\expandafter\labelrefsparsesubline\label@refs}%
6751         \stepcounter{subline}{1}%
6752         \def\label@refs{\theline|\thesubline|\theabsline}%
6753     \else%
6754         \def\label@refs{\theline|0|\theabsline}%
6755     \fi%
6756 }
6757 \def\labelrefsparseline#1|#2|#3{#1}%
6758 \def\labelrefsparsesubline#1|#2|#3{#2}%
6759 \def\labelrefsparseabsline#1|#2|#3{#3}%
6760 %

```

`\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.

```

6761 \newcommand*{\l@dmake@labels}{%
6762 \def\l@dmake@labels#1|#2|#3|#4|#5|#6|#7{%
6763   \expandafter\ifx\csname the@label\csuse{XR@prefix}#7\endcsname%
6764   \relax%
6765   \else%
6766     \led@warn@DuplicateLabel{\csuse{XR@prefix}#7}%
6767   \fi
6768   \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
6769   \csgdef{the@label\csuse{XR@prefix}#7}{#1|#2|#3|#4|#5|#6|\relax}%
6770   \ignorespaces}
6771 %
6772 %

```

TeX reads the aux file at both the beginning and end of the document, so we have to switch off duplicate label checking after the first time the file is read.

```

6773 \AtBeginDocument{%
6774   \def\l@dmake@labels#1|#2|#3|#4|#5|#6|#7{%
6775   }
6776 %
6777 %

```

**\@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.

```

6778
6779 \newcommand*{\@lab}{%
6780   \ifledRcol
6781     \xright@appenditem{\linenumr@p{\line@numR}}|%
6782     \ifsublines@ \sublinenumr@p{\subline@numR}\else 0\fi|%
6783     \the\absline@numR|%
6784     \current@annot%
6785   }%
6786   \to\labelref@listR
6787 \else
6788   \xright@appenditem{%
6789     \linenumr@p{\line@num}|%
6790     \ifsublines@ \sublinenumr@p{\subline@num}\else 0\fi|%
6791     \the\absline@num|%
6792     \current@annot%
6793   }%

```

```

6794 \to\labelref@list
6795 \fi}
6796 %

```

**\applabel** \applabel, if called in \edtext will insert automatically both a starting and an ending label for the current edtext lines.

```

6797 \newcommand*\applabel[1]{%
6798 \if@edtext@secondarg%
6799 %

```

Label should not be already defined.

```

6800 \ifcsundef{the@label#1}{%
6801 \csdef{the@label#1}{applabel}%
6802 }%
6803 {%
6804 \led@warn@DuplicateLabel{#1 (applabel)}%
6805 }%
6806 %

```

Parse the \edtext line numbers and annotations.

```

6807 \expandafter\l@dp@rsefootsspec\l@d@nums|%
6808 \expandafter\parse@annot\l@current@annot|%
6809 %

```

Use the  $\TeX$  standard hack for label.

```

6810 \@bsphack%
6811 %

```

And now, write the data in the auxiliary file.

```

6812 \ifl@Rcol%
6813 \protected@write\@auxout{%
6814 {\string\l@dmake@labelsR\space\l@dparsedstartpage|\l@dparsedstartline|\l@dparsedstartsub|\l@annot@start|\the\c@pstartR|{#1:
start}}}%
6815 \ifdef\hypertarget{%
6816 {\Hy@raisedlink{\hypertarget{#1:start}}}%
6817 }%
6818 \protected@write\@auxout{%
6819 {\string\l@dmake@labelsR\space\l@dparsedendpage|\l@dparsedendline
||\l@dparsedendsub|\l@annot@end|\the\c@pstartR|{#1:end}}}%
6820 \else%
6821 \ifl@dpairing%pstart or pstartL?
6822 \protected@write\@auxout{%
6823 {\string\l@dmake@labelsR\space\l@dparsedstartpage|\l@dparsedstartline|\l@dparsedstartsub|\l@annot@start|\the\c@pstartL|{#1:
start}}}%
6824 \ifdef\hypertarget{%
6825 {\Hy@raisedlink{\hypertarget{#1:start}}}%

```



```

6826     {}%
6827     \protected@write\@auxout{}%
6828     {\string\l@dmake@labels\space\l@dparsedendpage|\
l@dparsedendline|\l@dparsedendsub|\annot@end|\the\c@pstartL{#1:end}}%
6829     \else%
6830     \protected@write\@auxout{}%
6831     {\string\l@dmake@labels\space\l@dparsedstartpage|\
l@dparsedstartline|\l@dparsedstartsub|\annot@start|\the\c@pstart{#1:start}}%
6832     \ifdef{\hypertarget}%
6833     {\Hy@raisedlink{\hypertarget{#1:start}}}%
6834     {}%
6835     \protected@write\@auxout{}%
6836     {\string\l@dmake@labels\space\l@dparsedendpage|\
l@dparsedendline|\l@dparsedendsub|\annot@end|\the\c@pstart{#1:end}}%
6837     \fi%
6838     \fi%
6839     %

```

Use the  $\TeX$  standard hack for label.

```

6840     \esphack%
6841     %

```

Warning if `\applabel` is called outside of `\edtext`.

```

6842     \else%
6843     \led@warn@AppLabelOutSecondArgEdtext{#1}%
6844     \fi%
6845     %

```

End of `\applabel`

```

6846 }%
6847 %

```

**`\edlabels`** `\edlabels` and `\edlabelE` are just used to mark the beginning and the end of a passage.

```

\edlabelE
\edlabelSE
6848 \newcommand{\edlabels}[1]{%
6849   \edlabel{#1:start}%
6850 }
6851 \newcommand{\edlabelE}[1]{%
6852   \edlabel{#1:end}%
6853 }
6854 \newcommand{\edlabelSE}[1]{%
6855   \edlabels{#1}%
6856   \edlabelE{#1}%
6857 }
6858 %

```

**`\wrap@edcrossref`** `\wrap@edcrossref` is called around all `reledmac` crossref commands, except those which start with `x`. It adds the hyperlink.

```

6859 \newrobustcmd{\wrap@edcrossref}[2]{%
6860   \ifdef{\hyperlink}%
6861     {\hyperlink{#1}{#2}}%
6862     {#2}%
6863 }
6864 %

```

**\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.

ℒ<sub>TEX</sub> already defines a \pageref, so changing the name to \edpageref.

```

6865 \newcommand*{\edpageref}[1]{\l@dref@undefined{#1}\wrap@edcrossref{#1}{\l@dgetref@num{1}{#1}}}
6866 \newcommand*{\xpageref}[1]{\l@dgetref@num{1}{#1}}
6867
6868 %

```

**\edlineref** If the specified label exists, \lineref gives its line number.

```

6869 \newcommand*{\edlineref}[1]{%
6870   \l@dref@undefined{#1}%
6871   \wrap@edcrossref{#1}{%
6872     \providetoggle{label@#1@ledRcol}%Required for the first run, when the
label has not yet been parsed on the .aux file
6873     \iftoggle{label@#1@ledRcol}%
6874       {\linenumrepR{\l@dgetref@num{2}{#1}}}%
6875       {\linenumrep{\l@dgetref@num{2}{#1}}}%
6876     \xflagref{#1}%
6877   }%
6878 }%
6879 \newcommand*{\xlineref}[1]{\l@dgetref@num{2}{#1}}%
6880
6881 %

```

**\sublineref** If the specified label exists, \sublineref gives its sub-line number.

```

6882 \newcommand*{\sublineref}[1]{%
6883   \l@dref@undefined{#1}%
6884   \wrap@edcrossref{#1}{%
6885     \providetoggle{label@#1@ledRcol}%Required for the first run, when the
label has not yet been parsed on the .aux file
6886     \iftoggle{label@#1@ledRcol}%
6887       {\sublinenumrepR{\l@dgetref@num{3}{#1}}}%
6888       {\sublinenumrep{\l@dgetref@num{3}{#1}}}%

```

```

6889 }%
6890 }%
6891 \newcommand*{\xsublineref}[1]{\l@getref@num{3}{#1}}
6892
6893 %

```

**\xabslineref** If the specified label exists, \xabslineref gives its absolute line number. This is generally used only by reledmac internal macros.

```

6894 \newcommand*{\xabslineref}[1]{\l@getref@num{4}{#1}}%
6895 %

```

**\annotationref** If the specified label exists, \edannotationref gives the line number annotation with which it is associated.

```

6896 \newcommand*{\annotationref}[1]{%
6897   \l@dref@undefined{#1}%
6898   \wrap@edcrossref{#1}{%
6899     \l@wrap@ifnotemptybox{\Xwraplinenumannotation@ref}{%
6900       \l@getref@num{5}{#1}%
6901     }%
6902   }%
6903 }%
6904 \newcommand*{\xannotationref}[1]{%
6905   \l@getref@num{5}{#1}%
6906 }%
6907 %

```

**\pstartref** If the specified label exists, \pstartref gives its pstart number.

```

6908 \newcommand*{\pstartref}[1]{\l@dref@undefined{#1}\wrap@edcrossref{#1}{\l@getref@num{6}{#1}}}%
6909 \newcommand*{\xpstartref}[1]{\l@getref@num{6}{#1}}%
6910
6911 %

```

**\xflagref** \xflagref finds the side flag of any ref defined with \edlabel.

```

6912 \newcommand*{\xflagref}[1]{\l@getref@num{7}{#1}}%
6913 %

```

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.

```

6914 \newcommand*{\l@dref@undefined}[1]{%
6915   \expandafter\ifx\csname the@label#1\endcsname\relax
6916     \led@warn@RefUndefined{#1}%
6917   \fi}
6918
6919 %

```

**\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.

```

6920 \newcommand*{\l@dgetref@num}[2]{%
6921   \expandafter
6922   \ifx\csname the@label#2\endcsname \relax
6923     000%
6924   \else
6925     \expandafter\expandafter\expandafter
6926     \l@dlabel@parse\csname the@label#2\endcsname|#1%
6927   \fi}
6928
6929 %

```

**\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`.)

```

6930 \newcommand*{\l@dlabel@parse}{%
6931   \def\l@dlabel@parse#1|#2|#3|#4|#5|#6|#7|#8{%
6932     \ifcase #8%
6933       \or #1%
6934       \or #2%
6935       \or #3%
6936       \or #4%
6937       \or #5%
6938       \or #6%
6939       \or #7%
6940     \fi}
6941 %

```

**\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.

```

6942 \newcommand*{\xxref}[2]{%
6943   {%
6944     \expandafter\ifx\csname the@label#1\endcsname\relax%
6945       \let\@tempa\zz@@@%
6946       \def\@tempc{}%
6947     \else%
6948       \def\@tempa{%
6949         \l@ldgetref@num{1}{#1}|%
6950         \l@ldgetref@num{2}{#1}|%
6951         \l@ldgetref@num{3}{#1}%
6952       }%
6953       \edef\@tempc{%
6954         \l@ldgetref@num{5}{#1}%
6955       }%
6956     \fi%
6957     \expandafter\ifx\csname the@label#2\endcsname\relax%
6958       \let\@tempb\zz@@@%
6959       \def\@tempd{}%
6960     \else%
6961       \def\@tempb{%
6962         \l@ldgetref@num{1}{#2}|%
6963         \l@ldgetref@num{2}{#2}|%
6964         \l@ldgetref@num{3}{#2}%
6965       }%
6966       \edef\@tempd{%
6967         \l@ldgetref@num{5}{#2}%
6968       }%
6969     \fi%
6970     \global\appto\@beforeinsertofthisedtext{%
6971       \def\@this@crossref@start{#1}%
6972     }%
6973     \global\appto\@beforeinsertofthisedtext{%
6974       \def\@this@crossref@end{#2}%
6975     }%
6976     \linenum{\@tempa|\@tempb}%
6977     \edef\@tempe{\@tempc|\@tempd}%
6978     \expandafter\lineannot\expandafter{\@tempe}%
6979   }%
6980 }%
6981
6982
6983 %

```

`\appref`    `\appref`, `\Seref`, `\apprefwithpage`, `\Serefwithpage` and `\SEonlypage` print cross-  
`\apprefwithpage`  
`\Seref`  
`\Serefwithpage`  
`\Serefwithpage`

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).

Here we define hooks similar to some those related to critical footnotes or endnotes.

So, first declare the default value of the hooks for the pseudo-series. Also declare the internal toggle which are switch by `reledmac`.

```

6984 \def\Xtwolines@appref{}%
6985 \def\Xtwolines@SEref{}%
6986
6987 \def\Xmorethantwolines@appref{}%
6988 \def\Xmorethantwolines@SEref{}%
6989
6990 \def\Xlinerangeseparator@appref{\endashchar}%
6991 \def\Xlinerangeseparator@SEref{\endashchar}%
6992
6993 \def\Xsublinesep@appref{\fullstop}%
6994 \def\Xsublinesep@SEref{\fullstop}%
6995
6996 \def\Xpagelinesep@appref{\fullstop}%
6997 \def\Xpagelinesep@SEref{\fullstop}%
6998
6999
7000 \newtoggle{Xtwolinesbutnotmore@appref}%
7001 \newtoggle{Xtwolinesbutnotmore@SEref}%
7002
7003 \newtoggle{Xtwolinesonlyinsamepage@appref}%
7004
7005 \newtoggle{Xtwolinesonlyinsamepage@SEref}%
7006
7007 \newtoggle{Xlineflag@appref}%
7008 \toggletrue{Xlineflag@appref}%Here exception
7009 \newtoggle{Xlineflag@SEref}%
7010 \toggletrue{Xlineflag@SEref}%%Here exception
7011
7012 \newtoggle{Xlinenumannotationonlyfirst@SEref}%Never changed, just for the
test in \setprintlines
7013 \newtoggle{Xlinenumannotationonlyfirst@appref}%Idem
7014
7015 \def\Xlinenumannotationposition@appref{after}%
7016 \def\Xlinenumannotationposition@SEref{after}%
7017
7018 \def\Xwraplinenumannotation@appref{\textsuperscript}%
7019 \def\Xwraplinenumannotation@SEref{\textsuperscript}%
7020 \newtoggle{Xnoidenticallinenumannotation@appref}%
7021 \newtoggle{Xnoidenticallinenumannotation@SEref}%
7022
7023 \def\Xendtwolines@apprefwithpage{}%
7024 \def\Xendtwolines@SErefwithpage{}%
7025

```

```

7026 \def\Xendmorethantwolines@apprefwithpage{%
7027 \def\Xendmorethantwolines@SErefwithpage{%
7028
7029 \def\Xendlinrangeseparator@apprefwithpage{\endashchar}
7030 \def\Xendlinrangeseparator@SErefwithpage{\endashchar}
7031 \def\Xendlinrangeseparator@SErefonlypage{\endashchar}
7032
7033 \def\Xendbeforepagenumber@apprefwithpage{p.}%
7034 \def\Xendbeforepagenumber@SErefwithpage{p.}%
7035 \def\Xendbeforepagenumber@SEonlypage{p.}%
7036
7037 \def\Xendafterpagenumber@apprefwithpage{ }%
7038 \def\Xendafterpagenumber@SErefwithpage{ }%
7039
7040
7041 \def\Xendlineprefixsingle@apprefwithpage{%
7042 \def\Xendlineprefixsingle@SErefwithpage{%
7043
7044 \def\Xendlineprefixmore@apprefwithpage{%
7045 \def\Xendlineprefixmore@SErefwithpage{%
7046
7047 \newtoggle{Xendtwolinesbutnotmore@apprefwithpage}%
7048 \newtoggle{Xendtwolinesbutnotmore@SErefwithpage}%
7049
7050 \def\Xendsublinesep@apprefwithpage{\fullstop}%
7051 \def\Xendsublinesep@SErefwithpage{\fullstop}%
7052
7053 \newtoggle{Xendtwolinesonlyinsamepage@apprefwithpage}%
7054 \newtoggle{Xendtwolinesonlyinsamepage@SErefwithpage}%
7055
7056 \newtoggle{Xendlineflag@apprefwithpage}
7057 \toggletrue{Xendlineflag@apprefwithpage}%Here, exception
7058 \newtoggle{Xendlineflag@SErefwithpage}
7059 \toggletrue{Xendlineflag@SErefwithpage}%Here, exception
7060
7061
7062 \newtoggle{Xendlinenumannotationonlyfirst@SEref}%Never changed, just for
the test in \setprintendlines
7063 \newtoggle{Xendlinenumannotationonlyfirst@appref}%Idem
7064
7065 \def\Xendlinenumannotationposition@apprefwithpage{after}%
7066 \def\Xendlinenumannotationposition@SErefwithpage{after}%
7067
7068 \def\Xendwraplinenumannotation@apprefwithpage{\textsuperscript}%
7069 \def\Xendwraplinenumannotation@SErefwithpage{\textsuperscript}%
7070
7071 \newtoggle{Xendnoidenticallinenumannotation@apprefwithpage}%
7072 \newtoggle{Xendnoidenticallinenumannotation@SErefwithpage}%
7073 %

```

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.

```

7074
7075 \gdef\Xboxstartlinenum@appref{Opt}
7076 \gdef\Xboxstartlinenum@Seref{Opt}
7077
7078 \gdef\Xboxendlinenum@appref{Opt}
7079 \gdef\Xboxendlinenum@Seref{Opt}
7080
7081 \gdef\Xendboxstartlinenum@apprefwithpage{Opt}
7082 \gdef\Xendboxstartlinenum@Serefwithpage{Opt}
7083
7084 \gdef\Xendboxendlinenum@apprefwithpage{Opt}
7085 \gdef\Xendboxendlinenum@Serefwithpage{Opt}
7086
7087 \newtoggle{Xendpagenumberonlyfirst@apprefwithpage}
7088 \newtoggle{Xendpagenumberonlyfirst@Serefwithpage}
7089
7090 \newtoggle{Xendpagenumberonlyfirstifsingle@apprefwithpage}
7091 \newtoggle{Xendpagenumberonlyfirstifsingle@Serefwithpage}
7092
7093 \newtoggle{Xendpagenumberonlyfirstintwo@apprefwithpage}
7094 \newtoggle{Xendpagenumberonlyfirstintwo@Serefwithpage}
7095
7096 \gdef\Xendsympagenum@apprefwithpage{}
7097 \gdef\Xendsympagenum@Serefwithpage{}
7098
7099 \gdef\Xendinplaceofpagenumber@apprefwithpage{}
7100 \gdef\Xendinplaceofpagenumber@Serefwithpage{}
7101
7102 %

```

Now, declare the default values of `\@apprefprefixsingle` and `\@apprefprefixmore`, `\@Serefprefix`, `\@Serefprefixmore` and the commands which defines them.

```

7103 \newcommand\@apprefprefixsingle{}%
7104 \newcommand\@Serefprefixsingle{}%
7105
7106 \newcommand\@apprefprefixmore{}%
7107 \newcommand\@Serefprefixmore{}%
7108
7109 \newcommand{\setapprefprefixsingle}[1]{%
7110   \gdef\@apprefprefixsingle{#1}%
7111 }
7112 \newcommand{\setSerefprefixsingle}[1]{%
7113   \gdef\@Serefprefixsingle{#1}%
7114 }
7115
7116 \newcommand{\setapprefprefixmore}[1]{%

```



```

7117 \gdef\@apprefprefixmore{#1}%
7118 }
7119 \newcommand{\setSErefprefixmore}[1]{%
7120 \gdef\SErefprefixmore{#1}%
7121 }
7122 %
7123 %

```

And not `\setSErefonlypageprefixsingle` and `\setSErefonlypageprefixmore`.

```

7124 \newcommand{\setSErefonlypageprefixsingle}[1]{%
7125 \gdef\SErefonlypage@prefixsingle{#1}%
7126 }%
7127 \newcommand{\setSErefonlypageprefixmore}[1]{%
7128 \gdef\SErefonlypage@prefixmore{#1}%
7129 }%
7130 %

```

And now, the main commands: `\appref`, `\apprefwithpage`, `\SEref` and `\SErefwithpage`. These commands call `\reformatted@` and `\reformattedwithpage`, which calls `\printlines` and `\printendlines`. That is why we have previously declared all hooks values tested inside these last commands.

```

7131
7132 \newcommandx{\appref}[2][1,usedefault]{\reformatted@{#1}{#2}{appref}}
7133 \newcommandx{\SEref}[2][1,usedefault]{\reformatted@{#1}{#2}{SEref}}
7134
7135 \newcommandx{\apprefwithpage}[2][1,usedefault]{\reformattedwithpage@
7136 {#1}{#2}{appref}}
7137 \newcommandx{\SErefwithpage}[2][1,usedefault]{\reformattedwithpage@
7138 {#1}{#2}{SEref}}
7139
7140 \newcommand{\reformatted@}[3]{%
7141 \def\do##1{%
7142 \setkeys[mac]{truefootnoteoption}{##1}%
7143 }%
7144 \notblank{#1}{\docsvlist{#1}}{}%
7145 \xdef\@currentseries{#3}%
7146 \iftoggle{noprefix@}{}%
7147 {%
7148 \ifcempty{@#3prefixmore}%
7149 {\@apprefprefixsingle}%
7150 {%
7151 \IfEq{\xlineref{#2:start}}{\xlineref{#2:end}}%
7152 {\iftoggle{prefixmore@}%
7153 {\csuse{@#3prefixmore}}%
7154 {\csuse{@#3prefixsingle}}}%
7155 }%

```

```

7156     {\csuse{@#3prefixmore}}}%
7157   }%
7158 }%
7159 \ifboolexpr{%
7160   test{\ifcsundef{the@label#2:start}}}%
7161   or test{\ifcsundef{the@label#2:end}}}%
7162 }%
7163 {\led@warn@pairRefUndefined{#2}\nfss@text{\reset@font\bfseries ??}}%
7164 {%
7165   \def\@this@crossref@start{#2:start}%
7166   \def\@this@crossref@end{#2:end}%
7167   \xdef\annot@start{\xannotationref{#2:start}}%
7168   \xdef\annot@end{\xannotationref{#2:end}}%
7169   \printlines\xpageref{#2:start}|\xlineref{#2:start}|\xsublineref{#2:
start}|\xpageref{#2:end}|\xlineref{#2:end}|\xsublineref{#2:end}|\relax|
xflagref{#2:start}}|}%
7170 }%
7171 \def\do##1{%
7172   \setkeys[mac]{falsefootnoteoption}{##1}%
7173 }%
7174 \notblank{#1}{\docsvlist{#1}}}%
7175 }%
7176
7177 \newcommand{\reformattedwithpage@}[3]{%
7178   \def\do##1{%
7179     \setkeys[mac]{truefootnoteoption}{##1}%
7180   }%
7181   \notblank{#1}{\docsvlist{#1}}}%
7182   \xdef\@currentseries{#3withpage}%
7183   \ifboolexpr{%
7184     test{\ifcsundef{the@label#2:start}}}%
7185     or test{\ifcsundef{the@label#2:end}}}%
7186   }%
7187   {\led@warn@pairRefUndefined{#2}\nfss@text{\reset@font\bfseries ??}}%
7188   {%
7189     \def\@this@crossref@start{#2:start}%
7190     \def\@this@crossref@end{#2:end}%
7191     \printendlines\xpageref{#2:start}|\xlineref{#2:start}|\xsublineref{#2:
start}|\xpageref{#2:end}|\xlineref{#2:end}|\xsublineref{#2:end}|\relax|
xflagref{#2:start}}|}%
7192   }%
7193   \def\do##1{%
7194     \setkeys[mac]{falsefootnoteoption}{##1}%
7195   }%
7196   \notblank{#1}{\docsvlist{#1}}}%
7197 }%
7198
7199 \newcommand{\reformattedonlypage@}[3]{%
7200   \def\do##1{%
7201     \setkeys[mac]{truefootnoteoption}{##1}%

```

```

7202 }%
7203 \notblank{#1}{\docsvlist{#1}}{}}%
7204 \xdef\@currentseries{#3onlypage}%
7205 \ifboolexpr{%
7206   test{\ifcsundef{the@label#2:start}}}%
7207   or test{\ifcsundef{the@label#2:end}}}%
7208 }%
7209 {\led@warn@pairRefUndefined{#2}\nfss@text{\reset@font\bfseries ??}}%
7210 {\ifnumequal{\xpageref{#2:end}}{\xpageref{#2:start}}}%
7211   {%
7212     \ifcsvoid{#3onlypage@prefixsingle}%
7213     {}%
7214     {\csletcs{Xendbeforepagenumber@#3onlypage}{#3onlypage@prefixsingle
7215 }}%
7216     \printnpnum{%
7217       \wrap@edcrossref{#2:start}{\xpageref{#2:start}}}%
7218     }%
7219   {%
7220     \ifcsvoid{#3onlypage@prefixmore}%
7221     {}%
7222     {\csletcs{Xendbeforepagenumber@#3onlypage}{#3onlypage@prefixmore}}}%
7223   \ifdefined\linangesep{%
7224     \printnpnum{%
7225       \wrap@edcrossref{#2:start}{\xpageref{#2:start}}}%
7226     \linangesep%
7227     \wrap@edcrossref{#2:end}{\xpageref{#2:end}}}%
7228   }%
7229   \else%
7230     \printnpnum{%
7231       \wrap@edcrossref{#2:start}{\xpageref{#2:start}}}%
7232     \csuse{Xendlinangeseparator@{\@currentseries}}%
7233     \wrap@edcrossref{#2:end}{\xpageref{#2:end}}}%
7234   }%
7235   \fi%
7236 }%
7237 }%
7238 \def\do##1{%
7239   \setkeys[mac]{falsefootnoteoption}{##1}%
7240 }%
7241 \notblank{#1}{\docsvlist{#1}}{}}%
7242 }%
7243 %

```

**\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, fol-

lowed 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`.

```
7244 \newcommand*{\edmakelabel}[2]{\expandafter\xdef\csname the@label#1\
endcsname{#2}}
7245
7246 %
```

(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. 145 and V.9 p. 109), 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`.

```
7247 \pretocmd{\XR@test}{%
7248   {\XR@test@mac+++#1#2#3#4+++}%
7249   {}%
7250   {}%
7251   %
```

**\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`.

```
7252 \long\def\xR@test@mac+++#1+++{\XR@test@mac@test#1}
7253 %
```

**\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, not that the `\XR@prefix` is not called here, but it is integrated directly in `\l@dmake@labels` and `\l@dmake@labelsR`.

```
7254 \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
7255   \ifx#1\l@dmake@labels%
7256     \l@dmake@labels#2%
7257   \else
7258     \ifx#1\l@dmake@labelsR%
7259       \l@dmake@labelsR #2%
7260     \fi%
7261   \fi%
7262 }%
7263 %
```

## XXVI Sidenotes

Regular `\marginpar` 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 `\marginpar` in numbered text do not disturb the flow.

```

7264 \pretocmd{\@xympar}%
7265   {\ifnumberedpar@
7266     \led@warn@NoMarginpars
7267     \@esphack
7268   \else}%
7269   {}%
7270   {}%
7271
7272 \apptocmd{\@xympar}%
7273   {\fi}%
7274   {}
7275   {}
7276
7277 %

```

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

```

7278 \newcount\sidenote@margin
7279 \newcommand*{\sidenotemargin}[1]{\{%
7280   \l@dgetsidenote@margin{#1}%
7281   \ifnum\@l@tempcntb>\m@ne
7282     \ifledRcol
7283       \global\sidenote@marginR=\@l@tempcntb
7284     \else
7285       \global\sidenote@margin=\@l@tempcntb
7286     \fi
7287   \fi}}
7288 \newcommand*{\l@dgetsidenote@margin}[1]{\%
7289   \def\@tempa{#1}\def\@tempb{left}}%

```

```

7290 \ifx\@tempa\@tempb
7291 \l@ldtempcntb \z@
7292 \else
7293 \def\@tempb{right}%
7294 \ifx\@tempa\@tempb
7295 \l@ldtempcntb \@ne
7296 \else
7297 \def\@tempb{outer}%
7298 \ifx\@tempa\@tempb
7299 \l@ldtempcntb \tw@
7300 \else
7301 \def\@tempb{inner}%
7302 \ifx\@tempa\@tempb
7303 \l@ldtempcntb \thr@@
7304 \else
7305 \led@warn@BadSidenotemargin
7306 \l@ldtempcntb \m@ne
7307 \fi
7308 \fi
7309 \fi
7310 \fi}
7311 \sidenotemargin{right}
7312
7313 %

```

`\l@ldlp@rbox` We need two boxes to store sidenote texts.

```

\l@drp@rbox
7314 \newbox\l@ldlp@rbox
7315 \newbox\l@drp@rbox
7316
7317 %

```

`\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
7318 \newdimen\ledlsnotewidth \ledlsnotewidth=\marginparwidth
\ledrsnotesep
7319 \newdimen\ledrsnotewidth \ledrsnotewidth=\marginparwidth
\ledlsnotefontsetup
7320 \newdimen\ledlsnotesep \ledlsnotesep=\linenumsep
\ledrsnotefontsetup
7321 \newdimen\ledrsnotesep \ledrsnotesep=\linenumsep
7322 \newcommand*\ledlsnotefontsetup{\raggedleft\footnotesize}
7323 \newcommand*\ledrsnotefontsetup{\raggedright\footnotesize}
7324
7325 %

```

`\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`

```

7326 \newcommand*\ledleftnote}[1]{\edtext{}\l@dlsnote{#1}}
7327 \newcommand*\ledrightnote}[1]{\edtext{}\l@drsnote{#1}}
7328 \newcommand*\ledsidenote}[1]{\edtext{}\l@dcsnote{#1}}%
7329 \newcommand*\ledinnernote}[1]{\edtext{}\l@disnote{#1}}%
7330 \newcommand*\ledouternote}[1]{\edtext{}\l@dosnote{#1}}%
7331 %

```

`\l@dlsnote` . The ‘footnotes’ for left, right, and moveable sidenotes. The whole scheme is reminiscent of the critical footnotes code.

```

7332 \newif\ifrightrightnoteup
7333 \l@drsnote \rightnoteuptrue
7334 \l@disnote
7335 \newcommand*\l@dlsnote}[1]{%
7336   \begingroup%
7337   \newcommand{\content}{#1}%
7338   \ifnumberedpar@
7339     \ifledRcol%
7340       \xright@appenditem{\noexpand\l@dlsnote{\expandonce\content}}%
7341         \to\inserts@listR
7342       \global\advance\insert@countR \@ne%
7343     \else%
7344       \xright@appenditem{\noexpand\l@dlsnote{\expandonce\content}}%
7345         \to\inserts@list
7346       \global\advance\insert@count \@ne%
7347     \fi
7348   \fi%
7349   \ignorespaces%
7350 \endgroup%
7351 }%
7352
7353 \newcommand*\l@drsnote}[1]{%
7354   \begingroup%
7355   \newcommand{\content}{#1}%
7356   \ifnumberedpar@
7357     \ifledRcol%
7358       \xright@appenditem{\noexpand\l@drsnote{\expandonce\content}}%
7359         \to\inserts@listR
7360       \global\advance\insert@countR \@ne%
7361     \else%
7362       \xright@appenditem{\noexpand\l@drsnote{\expandonce\content}}%
7363         \to\inserts@list
7364       \global\advance\insert@count \@ne%
7365     \fi
7366   \fi\ignorespaces%
7367 \endgroup%
7368 }%
7369
7370 \newcommand*\l@dcsnote}[1]{%
7371   \begingroup%

```

```

7372 \newcommand{\content}{#1}%
7373 \ifnumberedpar@
7374   \ifledRcol%
7375     \xright@appenditem{\noexpand\vl@dcnote{\expandonce\content}}%
7376       \to\inserts@listR
7377     \global\advance\insert@countR \@ne%
7378   \else%
7379     \xright@appenditem{\noexpand\vl@dcnote{\expandonce\content}}%
7380       \to\inserts@list
7381     \global\advance\insert@count \@ne%
7382   \fi
7383 \fi\ignorespaces%
7384 \endgroup%
7385 }%
7386
7387 \newcommand*{\l@disnote}[1]{%
7388   \begingroup%
7389   \newcommand{\content}{#1}%
7390   \ifnumberedpar@%
7391     \ifledRcol%
7392       \xright@appenditem{\noexpand\vl@disnote{\expandonce\content}}%
7393         \to\inserts@listR%
7394       \global\advance\insert@countR \@ne%
7395     \else%
7396       \xright@appenditem{\noexpand\vl@disnote{\expandonce\content}}%
7397         \to\inserts@list%
7398       \global\advance\insert@count \@ne%
7399     \fi%
7400   \fi\ignorespaces%
7401   \endgroup%
7402 }%
7403
7404 \newcommand*{\l@dosnote}[1]{%
7405   \begingroup%
7406   \newcommand{\content}{#1}%
7407   \ifnumberedpar@%
7408     \ifledRcol%
7409       \xright@appenditem{\noexpand\vl@dosnote{\expandonce\content}}%
7410         \to\inserts@listR%
7411       \global\advance\insert@countR \@ne%
7412     \else%
7413       \xright@appenditem{\noexpand\vl@dosnote{\expandonce\content}}%
7414         \to\inserts@list%
7415       \global\advance\insert@count \@ne%
7416     \fi%
7417   \fi\ignorespaces%
7418   \endgroup%
7419 }%
7420
7421 %

```



`\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 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.

```

7422 \newcommand*{\vl@dlsnote}[1]{%
7423   \ifledRcol{%
7424     \@l@tempcntb=\sidenote@marginR%
7425     \ifnum\@l@tempcntb>\@ne%
7426       \advance\@l@tempcntb by\page@numR%
7427     \fi%
7428   \else%
7429     \@l@tempcntb=\sidenote@margin%
7430     \ifnum\@l@tempcntb>\@ne%
7431       \advance\@l@tempcntb by\page@num%
7432     \fi%
7433   \fi%
7434   \ifodd\@l@tempcntb%
7435     \listgadd{\l@dcsnotetext@l}{#1}%
7436   \else%
7437     \listgadd{\l@dcsnotetext}{#1}%
7438   \fi
7439 }
7440 \newcommand*{\vl@drsnote}[1]{%
7441   \ifledRcol{%
7442     \@l@tempcntb=\sidenote@marginR%
7443     \ifnum\@l@tempcntb>\@ne%
7444       \advance\@l@tempcntb by\page@numR%
7445     \fi%
7446   \else%
7447     \@l@tempcntb=\sidenote@margin%
7448     \ifnum\@l@tempcntb>\@ne%
7449       \advance\@l@tempcntb by\page@num%
7450     \fi%

```

```

7451 \fi%
7452 \ifodd\l@dttempcntb%
7453   \listgadd{\l@dcsnotetext}{#1}%
7454 \else%
7455   \listgadd{\l@dcsnotetext@r}{#1}%
7456 \fi%
7457 }
7458 \newcommand*{\vl@dcsnote}[1]{\listgadd{\l@dcsnotetext}{#1}}
7459
7460 \newcommand{\vl@disnote}[1]{%
7461   \ifledRcol%
7462     \@tempcnta=\page@numR%
7463   \else%
7464     \@tempcnta=\page@num%
7465   \fi%
7466   \ifodd\@tempcnta% ODD => right page => inner side = left side
7467     \vl@dlsnote{#1}%
7468   \else%
7469     \vl@drsnote{#1}%
7470   \fi%
7471 }%
7472
7473 \newcommand{\vl@dosnote}[1]{%
7474   \ifledRcol%
7475     \@tempcnta=\page@numR%
7476   \else%
7477     \@tempcnta=\page@num%
7478   \fi%
7479   \ifodd\@tempcnta% ODD => right page => outer side = right side
7480     \vl@drsnote{#1}%
7481   \else%
7482     \vl@dlsnote{#1}%
7483   \fi%
7484 }%
7485
7486 %

```

`\setl@dlp@rbox` `\setl@dlprbox{<lednums>}{<tag>}{<text>}` puts `<text>` into the `\l@dlp@rbox` box. `\setl@drpr@box` And similarly for the right side box. It is these boxes that finally get displayed in the margins.

```

7487 \newcommand*{\setl@dlp@rbox}[1]{%
7488   \begingroup%
7489     \parindent\z@\hsize=\ledlsnotewidth%
7490     \ledlsnotefontsetup% We kept it outside of the vbox, because can affect
the ragging
7491     \global\setbox\l@dlp@rbox%
7492     \ifleftnoteup%
7493       =\vbox to\z@{\ledlsnotefontsetup\vss #1}}% We put \
ledlsnotefontsetup inside footnote because required for color command. Note

```

```

the {} to keep setting local.
7494   \else%
7495     =\vbox to 0.70\baselineskip{{\ledrsnotefontsetup\strut#1\vss}}%
7496   \fi%
7497   \endgroup%
7498 }
7499
7500 \newcommand*{\setl@drp@rbox}[1]{%
7501   \begingroup%
7502     \parindent\z@\hsize=\ledrsnotewidth%
7503     \ledrsnotefontsetup%We kept it outside of the vbox, because can affect
the ragging
7504     \global\setbox\l@drp@rbox%
7505     \ifrightnoteup%
7506       =\vbox to\z@{{\ledrsnotefontsetup\vss#1}}%We put \ledrsnotefontsetup
inside footnote because required for color command. Note the {} to keep
setting local.
7507     \else%
7508       =\vbox to0.7\baselineskip{{\ledrsnotefontsetup\strut#1\vss}}%
7509     \fi%
7510   \endgroup%
7511 }%
7512 \newif\ifleftnoteup
7513   \leftnoteuptrue
7514 %

```

**\@sidenoteseq** This macro is used to separate sidenotes of the same line.

```

7515 \newcommand{\setsidenoteseq}[1]{\gdef\@sidenoteseq{#1}}
7516 \newcommand{\@sidenoteseq}{, }
7517 %

```

**\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 **\@sidenoteseq** as separator. It is the result that we put on the sidenote.

```

7518 \newcommand*{\affixside@note}{%
7519   \prepare@edindex@fornote{\the\page@num|\the\line@num|\the\subline@num|\
the\page@num|\the\line@num|\the\subline@num|}%
7520   \def\sidenotecontent@{}%
7521   \numgdef{\itemcount@}{0}%
7522   \def\do##1{%
7523     \ifnumequal{\itemcount@}{0}%
7524       {%
7525         \appto\sidenotecontent@{##1}}% Not print not separator before
the 1st note
7526       {\appto\sidenotecontent@{\@sidenoteseq ##1}}%
7527     }%

```

```

7528         \numgdef{\itemcount@}{\itemcount@+\@ne}%
7529     }%
7530     \dolistloop{\l@dcstotetext}%
7531     \ifnumgreater{\itemcount@}{1}{\led@err@ManySidenotes}{}%
7532 %

```

And we do the same for left and right notes (not movable).

```

7533 \gdef\@templ@d{%
7534 \gdef\@templ@n{\l@dcstotetext\l@dcstotetext@l\l@dcstotetext@r}%
7535 \ifx\@templ@d\@templ@n \else%
7536   \if@twocolumn%
7537     \if@firstcolumn%
7538       \setl@dlp@rbox{##1}{\sidenotecontent@}%
7539     \else%
7540       \setl@drp@rbox{\sidenotecontent@}%
7541     \fi%
7542   \else%
7543     \@l@tempcntb=\sidenote@margin%
7544     \ifnum\@l@tempcntb>\@ne%
7545       \advance\@l@tempcntb by\page@num%
7546     \fi%
7547     \ifodd\@l@tempcntb%
7548       \setl@drp@rbox{\sidenotecontent@}%
7549       \gdef\sidenotecontent@{%
7550         \numgdef{\itemcount@}{0}%
7551         \dolistloop{\l@dcstotetext@l}%
7552         \ifnumgreater{\itemcount@}{1}{\led@err@ManyLeftnotes}{}%
7553         \setl@dlp@rbox{\sidenotecontent@}%
7554       \else%
7555         \setl@dlp@rbox{\sidenotecontent@}%
7556         \gdef\sidenotecontent@{%
7557           \numgdef{\itemcount@}{0}%
7558           \dolistloop{\l@dcstotetext@r}%
7559           \ifnumgreater{\itemcount@}{1}{\led@err@ManyRightnotes}{}%
7560           \setl@drp@rbox{\sidenotecontent@}%
7561         \fi%
7562       \fi%
7563     \fi%
7564     \advance\@edindex@fornote@\m@ne%
7565   }
7566 %

```

## 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 `\@iiiminipage` and `\endminipage` macros. We will arrange this so that additional series can be easily added.

`\l@dfeetbeginmini` These will be the hooks in `\@iiiminpage` and `\endminipage`.  
`\l@dfeetendmini` They can be extended to handle other things if necessary.

```

7567 \ifnoledgroup@else%
7568 \newcommand*{\l@dfeetbeginmini}{\@ledgrouptrue\l@dedbeginmini\
l@dfambeginmini}
7569 \newcommand*{\l@dfeetendmini}{%
7570   \IfStrEq{critical-familiar}{\@mpfnpos}%
7571     {\l@dedendmini\l@dfamendmini}%
7572     {%
7573       \IfStrEq{familiar-critical}{\@mpfnpos}%
7574         {\l@dfamendmini\l@dedendmini}%
7575         {\do@feet@custom@order{mp@}{\@mpfnpos}}}%
7576   }%
7577 }%
7578 %

```

`\l@dedbeginmini` These handle the initiation and closure of critical footnotes in a minipage environment.

```

\l@dedendmini
\mp@append@Xnotes
7579 \newcommand*{\l@dedbeginmini}{%
7580   \unless\ifnocritical@%
7581     \def\do##1{%
7582       \csletcs{v##1footnote}{mpv##1footnote}%
7583     }%
7584     \dolistloop{\@series}%
7585   \fi%
7586 }
7587 \newcommand*{\l@dedendmini}{%
7588   \unless\ifnocritical@%
7589     \ifl@dpairing%
7590       \ifledRcol%
7591         \flush@notesR%
7592       \else%
7593         \flush@notes%
7594       \fi%
7595     \fi
7596     \def\do##1{%
7597       \mp@append@Xnotes{##1}%
7598     }%
7599     \dolistloop{\@series}%
7600   \fi%
7601 }%
7602 \newcommand{\mp@append@Xnotes}[1]{%
7603   \ifvoid\csuse{mp#1footins}\else%
7604     \ifl@dpairing%
7605       \ifparledgroup%
7606         \ifledRcol%
7607           \dingdef{\parledgroup@beforenotesR}{\parledgroup@beforenotesR+\skip
\@nameuse{mp#1footins}}%
7608         \else%

```

```

7609 \dingdef{\parledgroup@beforenotesL}{\parledgroup@beforenotesL+\
skip\@nameuse{mp#1footins}}%
7610 \fi%
7611 \fi%
7612 \fi%
7613 \ifcsstring{series@display#1}{paragraph}{}{%
7614 \setbox\@nameuse{mp#1footins}=\vbox{%
7615 \csuse{Xnotefontsize@#1}%
7616 \ifcsdef{Xhsize\csuse{series@display#1}@#1}{%
7617 \hsize\csuse{Xhsize\csuse{series@display#1}@#1}%
7618 }{}%
7619 \noindent\csuse{Xtxtbeforenotes@#1}%
7620 \unvbox\@nameuse{mp#1footins}%
7621 \@parboxrestore%
7622 }%
7623 }%
7624 \csuse{mp#1footgroup}{#1}%
7625 \fi%
7626 }%
7627 %

```

`\l@dfambeginmini` These handle the initiation and closure of familiar footnotes in a minipage environment.

```

\l@dfamendmini
\mp@append@notesX
7628 \newcommand*\l@dfambeginmini{%
7629 \unless\ifnofamiliar%
7630 \def\do##1{\csletcs{vfootnote##1}{mpvfootnote##1}}%
7631 \dolistloop{\@series}%
7632 \fi%
7633 }%
7634
7635 \newcommand*\l@dfamendmini{%
7636 \unless\ifnofamiliar%
7637 \def\do##1{%
7638 \mp@append@notesX{##1}%
7639 }%
7640 \dolistloop{\@series}%
7641 \fi%
7642 }%
7643 \newcommand{\mp@append@notesX}[1]{%
7644 \ifvoid\csuse{mpfootins#1}\else%
7645 \ifcsstring{series@displayX#1}{paragraph}{}{%
7646 \setbox\@nameuse{mpfootins#1}=\vbox{%
7647 \csuse{notefontsizeX@#1}%
7648 \ifcsdef{hsize\csuse{series@display#1}X@#1}{%
7649 \hsize\csuse{hsize\csuse{series@display#1}X@#1}%
7650 }{}%
7651 \noindent\csuse{txtbeforenotesX@#1}%
7652 \unvbox\@nameuse{mpfootins#1}%
7653 \@parboxrestore%
7654 }%

```

```

7655 }%
7656 \csuse{mpfootgroup#1}{#1}%
7657 \fi%
7658 }%
7659 %

```

`\@iiiminipage` This is our extended form of the kernel `\@iiiminipage` defined in `ltboxes.dtx`.

```

7660 \patchcmd%
7661 {\@iiiminipage}%
7662 {\let\@footnotetext\@mpfootnotetext}%
7663 {\let\@footnotetext\@mpfootnotetext\l@dfetbeginmini}%
7664 {}%
7665 {\led@error@fail@patch@\@iiiminipage}%
7666 %

```

`\endminipage` This is our extended form of the kernel `\endminipage` defined in `ltboxes.dtx`.

```

7667 \patchcmd%
7668 {\endminipage}%
7669 {\footnoterule}%
7670 {\footnoterule\l@advance@parledgroup@beforenormalnotes}%
7671 {}%
7672 {\led@error@fail@patch@\endminipage}
7673
7674 \patchcmd%
7675 {\endminipage}%
7676 {\@minipagefalse}%
7677 {\l@dfetendmini\@minipagefalse}%
7678 {}%
7679 {\led@error@fail@patch@\endminipage}
7680
7681 %

```

`\l@dunboxmpfoot` `\l@dunboxmpfoot` insert normal footnotes for `ledgroup`.  
`edgroup@beforenormalnotes`

```

7682 \newcommand*{\l@dunboxmpfoot}{%
7683 \vskip\skip\@mpfootins
7684 \normalcolor
7685 \footnoterule
7686 \l@advance@parledgroup@beforenormalnotes
7687 \unvbox\@mpfootins}%
7688 }
7689 %

```

When using parallel `ledgroup`, we need to store the vertical space added before footnote, in order to compensate them between left and right pages.

```

7690 \newcommand{\l@advance@parledgroup@beforenormalnotes}{%
7691 \ifparledgroup
7692 \ifl@pairing

```

```

7693     \ifledRcol
7694         \dimgdef{\parledgroup@beforenotesR}{\parledgroup@beforenotesR+
skip\@mpfootins}
7695     \else
7696         \dimgdef{\parledgroup@beforenotesL}{\parledgroup@beforenotesL+
skip\@mpfootins}
7697     \fi
7698     \fi
7699     \fi
7700 }
7701 %

```

**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.

```

7702
7703 \newenvironment{ledgroup}{%
7704     \resetprevpage@num%
7705     \def\@mpfn{mpfootnote}\def\thempfn{\thempfootnote}\c@mpfootnote\z@%
7706     \let\@footnotetext\@mpfootnotetext
7707     \l@dfetbeginmini%
7708 }{%
7709     \par
7710     \unskip
7711     \ifvoid\@mpfootins\else
7712         \l@dunboxmpfoot
7713     \fi
7714     \l@dfetendmini%
7715     \@ledgroupfalse%
7716 }
7717
7718
7719 %

```

**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.

```

7720 \newenvironment{ledgroupsize}[2][1]{%
7721 %

```

Set the various text measures.

```

7722     \hsize #2\relax
7723 %

```

Initialize fills for centering.

```

7724     \let\ledllfill\hfil
7725     \let\ledrfill\hfil

```



```

7726 \def\@tempa{#1}\def\@tempb{1}%
7727 %

```

Left adjusted numbered lines

```

7728 \ifx\@tempa\@tempb
7729 \let\ledllfill\relax
7730 \else
7731 \def\@tempb{r}%
7732 \ifx\@tempa\@tempb
7733 %

```

Right adjusted numbered lines

```

7734 \let\ledrfill\relax
7735 \fi
7736 \fi
7737 %

```

Set up the footnoting.

```

7738 \def\@mpfn{mpfootnote}\def\thempfn{\thempfootnote}\c@mpfootnote\z@
7739 \let\@footnotetext\@mpfootnotetext
7740 \l@dfetbeginmini%
7741 }{
7742 \par
7743 \unskip
7744 \ifvoid\@mpfootins\else
7745 \l@duboxmpfoot
7746 \fi
7747 \l@dfetendmini%
7748 }
7749
7750 %

```

Close the \ifnoledgroup@else.

```

7751 \fi%
7752 %

```

`\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`.

```

7753 \newif\ifledgroupnotesL@
7754 \newif\ifledgroupnotesR@
7755 %

```

## 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.

```

7756 \AtBeginDocument{%
7757   \unless\ifl@imakeidx%
7758     \ifpackageloaded{imakeidx}{\led@error@PackageAfterEledmac{imakeidx}}{}
7759   %
7759   \fi%
7760   \unless\ifl@indextools%
7761     \ifpackageloaded{indextools}{\led@error@PackageAfterEledmac{indextools}}{}%
7762   \fi%
7763   \unless\ifl@footmisc%
7764     \ifpackageloaded{footmisc}{\led@error@PackageAfterEledmac{footmisc}}{}
7765   %
7765   \fi%
7766 }
7767 %

```

### XXVIII.2 Auxiliary macros for \edindex

**\pagelinesep** In order to get a correct line number we have to use the label/ref mechanism. These macros are for that.

**\edindexlab**

```

\c@labidx
7768 \newcommand{\pagelinesep}{-}
7769 \newcommand{\edindexlab}{\&}
7770 \newcounter{labidx}
7771 \setcounter{labidx}{0}
7772
7773 %

```

**\doedindexlabel** This macro sets an \edlabel.

```

7774 \newcommand{\doedindexlabel}{%
7775   \stepcounter{labidx}%
7776   \edlabel{\edindexlab\thelabidx}%
7777 }
7778
7779 %

```

**\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.

```

7780 \newcounter{pageline}%
7781 \renewcommand{\thepageline}{%
7782   \thepage%
7783   \pagelinesep%
7784   \xlineref{\edindexlab\thelabidx}%

```

```
7785 }
7786 %
```

**\thestartpageline** These macros make up the page/line start/end number when the \edindex command is called in critical notes.

**\theendpageline**

```
7787 \newcommand{\thestartpageline}{%
7788   \l@dparsedstartpage%
7789   \pagelinesep%
7790   \l@dparsedstartline%
7791 }
7792 \newcommand{\theendpageline}{%
7793   \l@dparsedendpage%
7794   \pagelinesep%
7795   \l@dparsedendline%
7796 }
7797 %
```

### 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 sidenote), and decremented at the end of each note. If its value is greater than 0, that means we are inside a note.

```
7798 \newcount\@edindex@fornote@
7799 %
```

**\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 |.

```
7800 \newcommand{\prepare@edindex@fornote}[1]{%
7801   \l@dp@rsefootspec#1| %
7802   \advance\@edindex@fornote@\@ne%
7803 }
7804 %
```

**\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.

```
7805 \newcommand{\get@edindex@ledinnote@command}{%
7806   \ifxindy@%
7807     \gdef\@ledinnote@command{%
7808       ledinnote\thelabidx%
7809     }%
7810   \ifxindyhyperref@%
```

```

7811 \immediate\write\eledmac@xindy@out{%
7812 (define-attributes ("ledinnote\thelabidx"))^^J
7813 \space\space(markup-locref^^J
7814 \eledmacmarkuplocdepth^^J
7815 :open "\string\ledinnote[\edindexlab\thelabidx]{\@index@command
}{"^^J
7816 :close "}"^^J
7817 :attr "ledinnote\thelabidx"^^J
7818 )
7819 }%
7820 \else%
7821 \immediate\write\eledmac@xindy@out{%
7822 (define-attributes ("ledinnote\thelabidx"))^^J
7823 \space\space(markup-locref^^J
7824 \eledmacmarkuplocdepth^^J
7825 :open "\string\ledinnote{\@index@command}{"}^^J
7826 :close "}"^^J
7827 :attr "ledinnote\thelabidx"^^J
7828 )
7829 }%
7830 \fi%
7831 %

```

If we do not use xindy option, \@ledinnote@command will produce something like ledinnote{formattingcommand}.

```

7832 \else%
7833 \gdef\@ledinnote@command{%
7834 ledinnote[\edindexlab\thelabidx]{\@index@command}%
7835 }%
7836 \fi%
7837 }
7838 %

```

## 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 |.

```

7839 \def\get@index@command#1|#2+{%
7840 \gdef\@index@txt{#1}%
7841 \gdef\@index@command{#2}%
7842 \xdef\@index@parenthesis{%
7843 \IfBeginWith{\@index@command}{(}{%
7844 \StrGobbleLeft{\@index@command}{1}[\@index@command]%
7845 \global\let\@index@command\@index@command@%
7846 \xdef\@index@parenthesis{(%
7847 }%
7848 \IfBeginWith{\@index@command}{)}{%
7849 \StrGobbleLeft{\@index@command}{1}[\@index@command]%
7850 \global\let\@index@command\@index@command@%

```

```

7851 \xdef\@index@parenthesis{}}%
7852 }{}%
7853 }
7854 %

```

## 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 number itself.

`\ledinnotehyperpage`

`\ledinnotemark`

```

7855 \newcommand{\ledinnote}[3][1,usedefault]{%
7856 \ifboolexpr{%
7857 test{\ifdefequal{\iftrue}{\ifHy@hyperindex}}%
7858 or%
7859 bool {xindyhyperref@}%
7860 }%
7861 {%
7862 \csuse{#2}{\hyperlink{#1}{\ledinnotemark{#3}}}%
7863 }%
7864 {%
7865 \csuse{#2}{\ledinnotemark{#3}}%
7866 }%
7867 }%
7868 \newcommand{\ledinnotehyperpage}[2]{\csuse{#1}{\ledinnotemark{\hyperpage
7869 {#2}}}}%
7870 \newcommand{\ledinnotemark}[1]{#1\emph{n}}%
7871 %

```

## 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 of these two packages was loaded. This system forced to maintain 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`

```

7871 \newcommand{\@wredindex}[2][1=\expandonce\jobname,usedefault]{%#1 = the
index name, #2 = the text
7872 \begingroup%
7873 \let\emph\@firstofone%
7874 \let\textbf\@firstofone%
7875 \let\textit\@firstofone%
7876 \let\textmd\@firstofone%

```

```

7877 \let\textnormal\@firstofone%
7878 \let\textrm\@firstofone%
7879 \let\textsc\@firstofone%
7880 \let\textsf\@firstofone%
7881 \let\textsl\@firstofone%
7882 \let\texttt\@firstofone%
7883 \let\textup\@firstofone%
7884 \xdef\@tmp{#2}%To be used in IfSubStr instead of #2 directly. Avoid
some expansion bugs (for example with \edindex{textsc{something}})
7885 \endgroup%
7886 \ifl@imakeidx%
7887 \ifnum\@edindex@fornote@>\z@%
7888 \IfSubStr[1]{\@tmp}{|}{\get@index@command#2+}{\get@index@command#2|+}
%
7889 \get@edindex@ledinnote@command%
7890 \expandafter\imki@wrindexentry{#1}{\@index@txt|(\@ledinnote@command
}{\thestartpageline}%
7891 \expandafter\imki@wrindexentry{#1}{\@index@txt|)\@ledinnote@command
}{\theendpageline}%
7892 \else%
7893 \get@edindex@hyperref{#2}%
7894 \imki@wrindexentry{#1}{\@index@txt\@edindex@hyperref}{\thepageline}%
7895 \fi%
7896 \else%
7897 \ifnum\@edindex@fornote@>\z@%
7898 \IfSubStr[1]{\@tmp}{|}{\get@index@command#2+}{\get@index@command#2|+}
%
7899 \get@edindex@ledinnote@command%
7900 \expandafter\protected@write\@indexfile{}%
7901 {\string\indexentry{\@index@txt|(\@ledinnote@command}{\thestartpageline}
}%
7902 }%
7903 \expandafter\protected@write\@indexfile{}%
7904 {\string\indexentry{\@index@txt|)\@ledinnote@command}{\theendpageline}
}%
7905 }%
7906 \else%
7907 \protected@write\@indexfile{}%
7908 {\string\indexentry{#2}{\thepageline}
}%
7909 }%
7910 \fi%
7911 \fi%
7912 \endgroup
7913 \@esphack%
7914 }
7915 %

```

Need to add the definition of `\edindex` to `\makeindex`, and initialise `\edindex` to do nothing.

```

7916 \pretocmd{\makeindex}{%
7917 \def\edindex{%
7918 \ifboolexpr{bool{numbering} or bool{numberingR} or bool{

```

```

l@dprintingpages} or bool{l@dprintingcolumns}}{%
7919   \@bsphack%
7920   \doedindexlabel%
7921   \begingroup%
7922   \@sanitize%
7923   \@wredindex%
7924   }%
7925   {%
7926   \led@warn@edinde@outsidenumbering%
7927   \index%
7928   }%
7929   }%
7930   }%
7931   {}%
7932   {\led@error@fail@patch@makeindex}%
7933   \newcommand{\edindex}[1]{\@bsphack\@esphack}
7934   \newcommandx{\dummy@edindex}[2][1=\expandonce\jobname,usedefault]{}%
7935   %

```

## XXVIII.7 Hyperlink

**\hyperlinkformat** \hyperlinkformat command is to be used to have both a internal hyperlink and a format, when indexing.

```

7936 \newcommand{\hyperlinkformat}[3]{%
7937   \ifstrempy{#1}%
7938   {\hyperlink{#2}{#3}}%
7939   {\csuse{#1}{\hyperlink{#2}{#3}}%
7940   }}
7941 %

```

**\hyperlinkR** \hyperlinkR command is to be used to create a internal hyperlink and \ledRflag, when indexing.

```

7942 \newcommand{\hyperlinkR}[2]{%
7943   \hyperlink{#1}{#2\@Rlineflag}%
7944   }%
7945 %
7946 %

```

**\hyperlinkformatR** \hyperlinkformatR command is to be used to create a internal hyperlink, a format and a \@Rlineflag, when indexing.

```

7947 \newcommand{\hyperlinkformatR}[3]{%
7948   \hyperlinkformat{#1}{#2}{#3\@Rlineflag}%
7949   }%
7950 %
7951 %

```

`\get@edindex@hyperref` `\@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`).

```
7952 \newcommand{\get@edindex@hyperref}[1]{%
7953 %
```

We have to disable temporary spaces to work through a `xstring` bug (or feature?)

```
7954 \edef\temp@{%
7955 \catcode\ =9 %space need for catcode
7956 \detokenize{#1}%For active character in unicode
7957 \catcode\ =10 % space need for catcode
7958 }%
7959 %
```

Now, we define `\@edindex@hyperref` if the `hyperindex` of `hyperref` is enabled.

```
7960 \ifdefequal{\iftrue}{\ifHy@hyperindex}{%
7961 \IfSubStr{\temp@}{|}%
7962 {\get@index@command#1+%
7963 \ifledRcol%
7964 \gdef\@edindex@hyperref{|\@index@parenthesis %space kept
7965 hyperlinkformatR{\@index@command}%
7966 {\edindexlab\thelabidx}}%
7967 \else%
7968 \gdef\@edindex@hyperref{|\@index@parenthesis %space kept
7969 hyperlinkformat{\@index@command}%
7970 {\edindexlab\thelabidx}}%
7971 \fi%
7972 }%
7973 {\get@index@command#1+%
7974 \ifledRcol%
7975 \gdef\@edindex@hyperref{|\hyperlinkR{\edindexlab\thelabidx}}%
7976 \else%
7977 \gdef\@edindex@hyperref{|\hyperlink{\edindexlab\thelabidx}}%
7978 \fi%
7979 }%
7980 }%
7981 %
```

```
7982 % If we use both xindy and hyperref, first get the \protect\cs{
index@command} command.
7983 % Then define \protect\cs{@edindex@hyperref} in the form \verb+eledmacXXX+
7984 % \begin{macrocode}
7985 {\ifxindyhyperref%
7986 \IfSubStr{\temp@}{|}%
7987 {\get@index@command#1+%
7988 {\get@index@command#1+%
7989 \gdef\@edindex@hyperref{|\eledmac\thelabidx}%
7990 %
```

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`.



```

7991 \IfStrEq{\@index@parenthesis}{(}{%
7992   {%
7993   \csxdef{xindy@parenthesis@\@index@txt}{\thelabidx}%
7994   \gdef\@edindex@hyperref{|(eledmac\thelabidx}%
7995   }%
7996   {}%
7997 %

```

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`.

```

7998 \IfStrEq{\@index@parenthesis}{)}{%
7999   {%
8000   \xdef\@edindex@hyperref{)|eledmac\csuse{xindy@parenthesis@\@index@txt}}%
8001   \global\csundef{xindy@parenthesis@\@index@txt}%
8002   }%
8003 %

```

Write in the `.xdy` file the attributes of the location.

```

8004   {%
8005   \immediate\write\eledmac@xindy@out{%
8006     (define-attributes ("eledmac\thelabidx"))^^J
8007     \space\space(markup-locref^^J
8008     \eledmacmarkuplocdepth^^J
8009     :open "\string\hyperlink%
8010           \ifledRcol R\fi%
8011           {\edindexlab\thelabidx}%
8012           {\ifdefempty{\@index@command}%
8013            }%
8014            {\@backslashchar\@index@command}%
8015           {"^^J
8016     :close "})"^^J
8017     :attr "eledmac\thelabidx"^^J
8018     )
8019   }%
8020   }%
8021 %

```

And now, in any other case.

```

8022 \else%
8023   \gdef\@index@txt{#1}%
8024   \gdef\@edindex@hyperref{}%
8025 \fi%
8026 }%
8027 }
8028 %

```

## 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.

```

8029 \newcommand{\led@set@index@fornote}[1]{%
8030   \ifbool{indtl@innote}%
8031     {\let\index\nindex}%
8032     {}%
8033   \ifbool{indtl@notenumber}%
8034     {%
8035     \renewcommand{\index}[2][\indtl@jobname]{%
8036       \orig@@index[##1]{%
8037         ##2|innotenumber{\this@footnoteX@reading}%
8038       }%
8039     }%
8040     }%
8041     {}%
8042   }%
8043   %

```

`\led@reinit@index@fornote` The `\led@reinit@index@fornote` just reset the default value of `\index`.

```

8044 \newcommand{\led@reinit@index@fornote}{%
8045   \ifbool{indtl@innote}%
8046     {\let\index\orig@@index}%
8047     {}%
8048   \ifbool{indtl@notenumber}%
8049     {\let\index\orig@@index}%
8050     {}%
8051   }%
8052   %

```

## XXIX Glossaries

Here, we define the `\gls`-like commands prefixed by `ed`, only if the package `glossaries` is loaded.

```

8053 \AtBeginDocument{%
8054   \ifpackageloaded{glossaries}{%
8055     %

```

First those which arguments are `[<options>]{<label>}[<insert>]`.

```

8056   \renewcommand{\do}[1]{%
8057     \expandafter\DeclareRobustCommand\csname ed#1\endcsname[3][1,3,
usedefault]{%
8058       \doedindexlabel%
8059       \csname#1\endcsname[counter=pageline,##1]{##2}[##3]%

```

```

8060 }%
8061 \expandafter\WithSuffix\expandafter\DeclareRobustCommandx\csname ed
#1\endcsname*[3][1,3,usedefault]{%
8062 \doedindexlabel%
8063 \csname#1\endcsname*[counter=pageline,##1]{##2}{##3}%
8064 }%
8065 }%
8066 \docsvlist{%
8067 gls,%
8068 GLs,%
8069 GLS,%
8070 glspl,%
8071 Glspl,%
8072 GLSpl,%
8073 glstext,%
8074 Glstext,%
8075 GLStext,%
8076 Glsfirst,%
8077 GLSfirst,%
8078 glsplural,%
8079 Glsplural,%
8080 GLSplural,%
8081 glsfirstplural,%
8082 Glsfirstplural,%
8083 GLSfirstplural,%
8084 glsname,%
8085 Glname,%
8086 GLSname,%
8087 glssymbol,%
8088 Glssymbol,%
8089 GLSsymbol,%
8090 glsdesc,%
8091 Glsdesc,%
8092 GLSdesc,%
8093 glsuseri,%
8094 Glsuseri,%
8095 GLSuseri,%
8096 glsuserii,%
8097 Glsuserii,%
8098 GLSuserii,%
8099 glsuseriii,%
8100 Glsuseriii,%
8101 GLSuseriii,%
8102 glsuseriv,%
8103 Glsuseriv,%
8104 GLSuseriv,%
8105 glsuserv,%
8106 Glsuserv,%
8107 GLSuserv,%
8108 glsuservi,%

```

```

8109     Glsuservi,%
8110     GLSuservi%
8111     }%
8112 %

```

Then those which arguments are [*options*]{*label*}{*link text*}.

```

8113     \renewcommand{\do}[1]{%
8114         \expandafter\DeclareRobustCommandx\csname ed#1\endcsname[3][1,
usedefault]{%
8115             \doedindexlabel%
8116             \csname#1\endcsname[counter=pageline,##1]{##2}{##3}%
8117         }%
8118         \expandafter\WithSuffix\expandafter\DeclareRobustCommandx\csname ed
#1\endcsname*[3][1,usedefault]{%
8119             \doedindexlabel%
8120             \csname#1\endcsname*[counter=pageline,##1]{##2}{##3}%
8121         }%
8122     }%
8123     \docsvlist{glsdisp,glslink}%
8124 %

```

Then those which arguments are [*options*]{*label*}.

```

8125     \renewcommand{\do}[1]{%
8126         \expandafter\DeclareRobustCommandx\csname ed#1\endcsname[2][1,
usedefault]{%
8127             \doedindexlabel%
8128             \csname#1\endcsname[counter=pageline,##1]{##2}%
8129         }%
8130         \expandafter\WithSuffix\expandafter\DeclareRobustCommandx\csname ed
#1\endcsname*[2][1,usedefault]{%
8131             \doedindexlabel%
8132             \csname#1\endcsname*[counter=pageline,##1]{##2}%
8133         }%
8134     }%
8135     \docsvlist{glsadd}%
8136 }{}%
8137 }%
8138 %

```

## 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` 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.

```
8139 \def\@hangingsymbol{}
8140 \newcommand*\sethangingsymbol[1]{%
8141   \gdef\@hangingsymbol{#1}%
8142 }%
8143 \newif\ifinstanza
8144 %
```

`\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.

```
8145 \newif\ifinserthangingsymbol
8146 \newcommand*\inserthangingsymbol{%
8147   \ifinserthangingsymbol%
8148     \ifinstanza%
8149       \@hangingsymbol%
8150     \fi%
8151   \fi%
8152 }
8153 %
```

### 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.

```
8154 \newcommand*\ampersand{\char` \&}
8155
8156 %
```

### 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.

```

8157 \chardef\body=\catcode`\@
8158 \catcode`\@=11
8159 \chardef\next=\catcode`\&
8160 \catcode`\&=\active
8161
8162 %

```

### 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.

```

8163 \newcount\stanza@count
8164 \newlength{\stanzaindentbase}
8165 \setlength{\stanzaindentbase}{20pt}
8166
8167 %

```

`\strip@szacnt`  
`\setstanzavalues`

The indentations of stanza lines are non-negative integer multiples of the unit called `\stanzaindentbase`. To make it easier for the user to specify these numbers, some list macros are defined. These take numerical values in a list separated by commas and assign the values to special control sequences using `\mathchardef`. Though this does limit the range from 0 to 32767, it should suffice for most applications, including *penalties*, which will be discussed below.

```

8168 \def\strip@szacnt#1,#2|{\def\@tempb{#1}\def\@tempa{#2|}}
8169 \newcommand*\setstanzavalues[2]{\def\@tempa{#2,|}%
8170   \stanza@count\z@
8171   \def\next{\expandafter\strip@szacnt\@tempa
8172     \ifx\@tempb\empty\let\next\relax\else
8173       \expandafter\mathchardef\csname #1@\number\stanza@count
8174       \@endcsname\@tempb\relax
8175       \advance\stanza@count\@one\fi\next}%
8176   \next}
8177
8178 %

```

`\setstanzaindents`  
`\setstanzapenalties`

In the original edmac, `\setstanzavalues{sza}{\langle... \rangle}` had to be called to set the indents, and similarly `\setstanzavalues{szp}{\langle... \rangle}` to set the penalties. `\setstanzaindents` and `\setstanzapenalties` macros are a convenience to give the user one less thing to worry about (misspelling the first argument).

```

8179 \newcommand*\setstanzaindents[1]{\setstanzavalues{sza}{#1}}
8180 \newcommand*\setstanzapenalties[1]{\setstanzavalues{szp}{#1}}
8181 %
8182 %

```

`\managestanza@modulo` Since version 0.13, the `stanzaindentsrepetition` counter can be used when the indentation is repeated every  $n$  verses. The `\managestanza@modulo` is a command which modifies the counter `stanza@modulo`. The command adds 1 to `stanza@modulo`, but if `stanza@modulo` is equal to the `stanzaindentsrepetition` counter, the command restarts it.

```

8183 \newcounter{stanzaindentsrepetition}
8184 \newcount\stanza@modulo
8185
8186 \newcommand*{\managestanza@modulo}[0]{%
8187   \advance\stanza@modulo\@ne%
8188   \ifnum\stanza@modulo>\value{stanzaindentsrepetition}%
8189     \stanza@modulo\@ne%
8190   \fi%
8191 }
8192 %

```

`\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.

```

8193 \newcommand{\stanzaindent}[1]{%
8194   \hspace{\dimexpr#1\stanzaindentbase-\parindent\relax}%
8195   \ignorespaces%
8196 }%
8197 \WithSuffix\newcommand\stanzaindent*[1]{%
8198   \stanzaindent{#1}%
8199   \global\advance\stanza@modulo-\@ne%
8200   \ifnum\stanza@modulo=0%
8201     \global\stanza@modulo=\value{stanzaindentsrepetition}%
8202   \fi%
8203   \ignorespaces%
8204 }%
8205 %

```

## XXX.5 Numbering stanza

Here, macro for numbering stanza. First, the stanza counter.

```

\thestanza \newcounter{stanza}
8206
8207 \renewcommand{\thestanza}{%
8208   \textbf{\arabic{stanza}}%
8209 }
8210 %

```

`\ifnumberstanza` Then, macro to activate automatically numbering of stanza.

```

8211 \newif\ifnumberstanza%
8212 %

```

`\@insertstanzanumber` Now, macro called at the first line of of verse of a stanza.

```

8213 \newcommand{\@insertstanzanumber}[0]{%
8214   \ifnumberstanza%
8215   \ifl@dpairing%
8216     \ifledRcol%
8217       \stanzanumwrapper{\thestanzaR}%
8218     \else%
8219       \stanzanumwrapper{\thestanzaL}%
8220     \fi%
8221   \else%
8222     \stanzanumwrapper{\thestanza}%
8223   \fi%
8224   \setline{1}%
8225 \fi%
8226 }%
8227 %

```

`\@advancestanzanumber` Also a command to advance the counter of stanza.

```

8228 \newcommand{\@advancestanzanumber}[0]{%
8229   \ifnumberstanza%
8230   \ifl@dpairing%
8231     \ifledRcol%
8232       \addtocounter{stanzaR}{1}%
8233     \else%
8234       \addtocounter{stanzaL}{1}%
8235     \fi%
8236   \else%
8237     \addtocounter{stanza}{1}%
8238   \fi%
8239 \fi%
8240 }%
8241 %

```

`\stanzanumwrapper` And finally, the wrapper for stanza number

```

8242 \newcommand{\stanzanumwrapper}[1]{%
8243   \flagstanza{#1}%
8244 }%
8245 %

```

## XXX.6 Stanza number in note

Here, the command called when printing stanza number in notes.

```

8246 \newcommand{\printstanza}[0]{%
8247   \ifboolexpr{bool{l@dpairing} or bool{l@dprintingpages} or bool{
8248     l@dprintingcolumns}}{%

```



```

8249         \thestanzaR%
8250     \else%
8251         \thestanzaL%
8252     \fi%
8253 }{%
8254     \thestanza%
8255 }%
8256 }
8257 %

```

### 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.

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.

```

8258 \newcommand{\stanza@line}[2][1,2,usedefault]{%
8259     \ifnum\value{stanzaindentsrepetition}=0
8260         \ifcsdef{sza@number\stanza@count @}{%
8261             {%
8262                 \parindent=\csname sza@number\stanza@count @\endcsname\
stanzaindentbase%
8263             }{%
8264                 \led@err@StanzaIndentNotDefined%
8265             }%
8266         \else
8267             \ifcsdef{sza@number\stanza@modulo @}{%
8268                 \parindent=\csname sza@number\stanza@modulo @\endcsname\
stanzaindentbase%
8269                 \managestanza@modulo%
8270             }%
8271             {%
8272                 \led@err@StanzaIndentNotDefined%
8273             }%
8274         \fi
8275         \pstart[#1][#2]\stanza@hang\ignorespaces%
8276     }%
8277 \xdef\stanza@hang{\noexpand\leavevmode\noexpand\startlock
8278     \hangindent\expandafter
8279     \noexpand\csname sza@00\endcsname\stanzaindentbase
8280     \hangafter\@ne}
8281 \def\sza@penalty{\count@\csname szp@number\stanza@count @\endcsname
8282     \ifnum\count@>\@M\advance\count@-\@M\penalty-\else
8283     \penalty\fi\count@}

```

```

8284 %

\@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 \&.

8285 \xdef\@startstanza[#1][#2]{%
8286 \noexpand\instanzatrue\expandafter
8287 \begingroup%
8288 \catcode`\noexpand\&\active%
8289 \global\stanza@count\@ne\stanza@modulo\@ne
8290 \noexpand\ifnum\expandafter\noexpand
8291 \csname sza@00\endcsname=\z@\let\noexpand\stanza@hang\relax
8292 \let\noexpand\endlock\relax\noexpand\else\interlinepenalty
8293 \@M\rightskip\z@ plus 1fil\relax\noexpand\fi\noexpand\ifnum
8294 \expandafter\noexpand\csname szp@00\endcsname=\z@
8295 \let\noexpand\sza@penalty\relax\noexpand\fi%
8296 \def\noexpand&{%
8297 \noexpand\newverse[] []}%
8298 \def\noexpand\&{\noexpand\@stopstanza}%
8299 \noexpand\@advancestanzanumber%
8300 \noexpand\stanza@line[#1][#2]%
8301 \noexpand\@insertstanzanumber%
8302 \let\par\relax\ignorespaces%No paragraph in verses
8303 }
8304
8305 \newcommandx{\stanza}[2][1,2,usedefault]{%
8306 \ifboolexpr{%
8307 not test{\ifdefvoid{\@at@every@stanza}}%
8308 and test{\ifstrempy{#1}}%
8309 and test{\ifstrempy{#2}}}%
8310 {\@startstanza[] [\@at@every@stanza]\@at@start@every@stanza}%
8311 {\@startstanza[#1][#2]\@at@start@every@stanza}%
8312 }%
8313
8314 \newcommandx{\@stopstanza}[2][1,2,usedefault]{%
8315 \unskip%
8316 \endlock%
8317 \ifboolexpr{%
8318 not test{\ifdefvoid{\@at@every@stop@stanza}}%
8319 and test{\ifstrempy{#1}}%
8320 and test{\ifstrempy{#2}}}%
8321 {\before@every@stop@stanza\pend[] [\@at@every@stop@stanza]}%
8322 {\before@every@stop@stanza\pend[#1][#2]}%

```

```

8323 \endgroup%
8324 \instanzafalse%
8325 }
8326
8327 \newcommand{\AtEveryStopStanza}[1]{%
8328 \ifstrepty{#1}%
8329 {\gdef\at@every@stop@stanza{}}%
8330 {\gdef\at@every@stop@stanza{\noindent#1}}%
8331 }%
8332 \WithSuffix\newcommand\AtEveryStopStanza*[1]{%
8333 \ifstrepty{#1}%
8334 {\gdef\at@every@stop@stanza{}}%
8335 {\gdef\at@every@stop@stanza{#1}}%
8336 }%
8337 \def\at@every@stop@stanza{}%
8338
8339 \newcommand{\AtEveryStanza}[1]{%
8340 \ifstrepty{#1}%
8341 {\gdef\at@every@stanza{}}%
8342 {\gdef\at@every@stanza{\noindent#1}}%
8343 }%
8344 \WithSuffix\newcommand\AtEveryStanza*[1]{%
8345 \ifstrepty{#1}%
8346 {\gdef\at@every@stanza{}}%
8347 {\gdef\at@every@stanza{#1}}%
8348 }%
8349
8350
8351
8352 \newcommand{\AtStartEveryStanza}[1]{%
8353 \ifstrepty{#1}%
8354 {\gdef\at@start@every@stanza{}}%
8355 {\gdef\at@start@every@stanza{#1}}%
8356 }%
8357 \def\at@start@every@stanza{}%
8358
8359 \newcommand{\BeforeEveryStopStanza}[1]{%
8360 \ifstrepty{#1}%
8361 {\gdef\before@every@stop@stanza{}}%
8362 {\gdef\before@every@stop@stanza{#1}}%
8363 }%
8364 \def\before@every@stop@stanza{}%
8365
8366 \newcommandx*\newverse}[4][1,2,3,4,usedefault]{%
8367 \unskip%
8368 \endlock\pend[#1][#3]\sza@penalty\global%
8369 \advance\stanza@count\@ne\stanza@line[#2][#4]%
8370 }
8371
8372 %

```

**\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`.

```
8373 \newcommand*{\flagstanza}[2][\stanzaindentbase]{%
8374   \hskip -#1\llap{#2}\hskip #1\ignorespaces}
8375
8376 %
```

## 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.

```
8377 \catcode`\&=\next
8378 \catcode`\@=\body
8379 \setstanzavalues{szp}{0}
8380
8381 %
```

# 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
\msdata@cR
8382 \def\msdata@c{}%
8383 \def\msdata@cR{}%
8384 \newcommand{\msdata}[1]{%
8385   \leavevmode%
8386   \unless\ifstopmsdata@inserted@%
8387     \stopmsdata%
8388     \led@warning@msdatawithoutstop%
8389   \fi%
8390   \global\stopmsdata@inserted@false%
8391   \unless\ifledRcol%
8392     \numgdef{\msdata@c}{\msdata@c+1}%
8393     \ifdef{\hypertarget}{%
8394       \edlabel{\msdata@c:start:msdata}%
8395     }{}%
8396     \protected@write\linenum@out{}{%
8397       \string\@msd{#1}%
8398     }%
8399   \else%
```

```

8400 \numdef{\msdata@cR}{\msdata@cR+1}%
8401 \ifdef{\hypertarget}{%
8402   \edlabel{\msdata@cR:start:msdata}%
8403 }{}%
8404 \protected@write\linenum@outR{%
8405   \string\@msd{#1}%
8406 }%
8407 \fi%
8408 }%
8409 %

```

**\stopmsdata** The user level `\stopmsdata` command only writes information about the end of manuscripts data in numbered auxiliary file.

```

8410 \newcommand{\stopmsdata}[0]{%
8411   \leavevmode%
8412   \unless\ifledRcol%
8413     \protected@write\linenum@out{%
8414       \string\@stopmsd%
8415     }%
8416     \ifdef{\hypertarget}{%
8417       \edlabel{\msdata@c:end:msdata}%
8418     }{}%
8419   \else%
8420     \protected@write\linenum@outR{%
8421       \string\@stopmsd%
8422     }%
8423     \ifdef{\hypertarget}{%
8424       \edlabel{\msdata@cR:end:msdata}%
8425     }{}%
8426   \fi%
8427   \global\stopmsdata@inserted@true%
8428 }%
8429 %

```

**\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`

```

8430 \newif\ifstopmsdata@inserted@%
8431 %

```

## 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?

```

8432 \newcommand{\setmsdataseries}[1]{%
8433   \gdef\@msdata@series{#1}%
8434 }%
8435 \def\@msdata@series{A}%
8436 %

```

**\setmsdataposition** The label for the manuscripts data.

```

8437 \def\ms@data@position{msdata-regular}%
8438 \newcommand{\setmsdataposition}[1]{%
8439   \gdef\ms@data@position{#1}%
8440 }%
8441 %

```

**\setmsdatalabel** The label for the manuscripts data.

```

8442 \def\ms@data@label{Ms.}%
8443 \newcommand{\setmsdatalabel}[1]{%
8444   \gdef\ms@data@label{#1}%
8445 }%
8446 %

```

### XXXI.3 Counters and lists

**\@msd@c** \@msd@c is a counter incremented at each \@msd read in auxiliary file.

```

8447 \numdef{\@msd@c}{0}
8448 \numdef{\@msd@cR}{0}
8449 %

```

**\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.

```

8450 \numdef{\add@msd@c}{0}%
8451 \numdef{\add@msd@cR}{0}%
8452 %

```

**\@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.

```

8453 \def\@msdata@list{}%
8454 %

```

## 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.

```

8455 \newcommand{\@msd}[1]{%
8456   \unless\ifledRcol%
8457     \global\newdef{\@msd@c}{\@msd@c+\@ne}%
8458     \csgdef{\@msdata@\@msd@c @data}{#1}%
8459     \csxdef{\@msdata@\@msd@c @linenumber}{\the\line@num}%
8460     \csxdef{\@msdata@\@msd@c @abslinenumber}{\the\absline@num}%
8461     \xright@appenditem{\the\absline@num}\to\actionlines@list%
8462     \xright@appenditem{-1010}\to\actions@list%
8463   \else%
8464     \global\newdef{\@msd@cR}{\@msd@cR+\@ne}%
8465     \csgdef{\@msdata@\@msd@cR @dataR}{#1}%
8466     \csxdef{\@msdata@\@msd@cR @linenumberR}{\the\line@numR}%
8467     \csxdef{\@msdata@\@msd@cR @abslinenumberR}{\the\absline@numR}%
8468     \xright@appenditem{\the\absline@numR}\to\actionlines@listR%
8469     \xright@appenditem{-1010}\to\actions@listR%
8470   \fi%
8471 }%
8472 %

```

**\@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.

```

8473 \newcommand{\@stopmsd}[0]{%
8474   \unless\ifledRcol%
8475     \ifcsundef{\@msdata@\@msd@c @stoplinenumber}{%
8476       \csxdef{\@msdata@\@msd@c @stopabslinenumber}{\the\absline@num}%
8477       \csxdef{\@msdata@\@msd@c @stoplinenumber}{\the\line@num}%
8478     }{}%
8479   \else%
8480     \ifcsundef{\@msdata@\@msd@cR @stoplinenumberR}{%
8481       \csxdef{\@msdata@\@msd@cR @stopabslinenumberR}{\the\absline@numR}%
8482       \csxdef{\@msdata@\@msd@cR @stoplinenumberR}{\the\line@numR}%
8483     }%
8484     {}%
8485   \fi%
8486 }%
8487 %

```

## 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.

```

8488 \newcommand{\add@msdata}{%
8489   \bgroup%
8490   \normalfont%
8491   \unless\ifledRcol{%
8492     \numgdef{\add@msd@c}{\add@msd@c+\@ne}%
8493     \ifcsdef{@msdata@\add@msd@c @data}{%
8494       \letcs{\@data}{@msdata@\add@msd@c @data}%
8495       \edef\l@d@nums{%
8496         000| % Start page = we don't print it
8497         \csuse{@msdata@\add@msd@c @linenumber}| % Start line number
8498         000| % Start subline number, for now, not used
8499         000| % End page number, we don't print it
8500         \ifnumless{\csuse{@msdata@\add@msd@c @stopabslinenumber}}{\csuse{
@lastabsline@forpage@the\page@num}}}%
8501         {\csuse{@msdata@\add@msd@c @stoplinenumber}}}%End line number if
in the same page
8502         {\csuse{@lastline@forpage@the\page@num}}}%Otherwiser, last
number of the page
8503         |%
8504         000| % End sub line number, for now, not used
8505         \edfont@info%Font
8506       }%
8507       \@msd@options@fullpagefalse%
8508       \if@firstlineofpage%Try if the data are for the full page. If yes
, will add options to the list.
8509       \unless\if@msdata@insertedfrompreviouspage%
8510       \ifnumless{\csuse{@lastabsline@forpage@the\page@num}}{\csuse{
@msdata@\add@msd@c @stopabslinenumber}+\@ne}%
8511       {%
8512       \numdef{\@tmp}{\add@msd@c+\@ne}%
8513       \ifcsdef{@msdata@\@tmp @abslinenumber}%
8514       {\ifnumequal{\csuse{@msdata@\@tmp @abslinenumber}}{\csuse{
@lastabsline@forpage@the\page@num}}}%
8515       {}%
8516       {\@msd@options@fullpagetrue}%
8517       }%
8518       {\@msd@options@fullpagetrue}%
8519     }%
8520   }%
8521   \fi%
8522 \fi%
8523 \listxadd{\@msdata@list}{%
8524   \@msd@options@iffullpage%
8525   \ifluatex%
8526     \csxdef{footnote@luatextextdir}{\the\textdir}%
8527     \csxdef{footnote@luatexpardir}{\the\pardir}%
8528   \fi%

```



```

8529     \csdef{@this@crossref@start}{\add@msd@c:start:msdata}%
8530     \csdef{@this@crossref@end}{\add@msd@c:end:msdata}%
8531     \unexpanded{%
8532       \def\annot@start{%
8533         \def\annot@end{%
8534           }%
8535         \noexpand\csuse{v\@msdata@series footnote}{\@msdata@series}{\
expandonce\l@d@nums}{\ms@data@label}{\expandonce\@data}}%
8536       \reset@msd@options@iffullpage%
8537     }%
8538   }%
8539   {%
8540   \else%
8541     \numgdef{\add@msd@cR}{\add@msd@cR+\@ne}%
8542     \ifcsdef{@msdata@\add@msd@cR @dataR}{%
8543       \letcs{\@data}{@msdata@\add@msd@cR @dataR}%
8544       \edef\l@d@nums{%
8545         000|% Start page = we don't print it
8546         \csuse{@msdata@\add@msd@cR @linenumberR}|% Start line number
8547         000|% Start subline number, for now, not used
8548         000|% End page number, we don't print it
8549         \ifnumless{\csuse{@msdata@\add@msd@cR @stopabslinenumberR}}{\
csuse{@lastline@forpageR@the\page@numR}}%
8550         {\csuse{@msdata@\add@msd@cR @stoplinenumberR}}%End line number
if in the same page
8551         {\csuse{@lastline@forpageR@the\page@numR}}%Otherwiser, last
number of the page
8552         |%
8553         000|% End sub line number, for now, not used
8554         \edfont@info%Font
8555       }%
8556       \@msd@options@fullpagefalse%
8557       \if@firstlineofpageR%
8558         \unless@if@msdata@insertedfrompreviouspage%
8559         \ifnumless{\csuse{@lastabsline@forpageR@the\page@numR}}{\
csuse{@msdata@\add@msd@c @stopabslinenumberR}+\@ne}%
8560         {%
8561           \numdef{\@tmp}{\add@msd@cR+\@ne}%
8562           \ifcsdef{@msdata@\@tmp @abslinenumberR}%
8563           {\ifnumequal{\csuse{@msdata@\@tmp @abslinenumberR}}{\csuse{
@lastabsline@forpageR@the\page@numR}}}%
8564           {}%
8565           {\@msd@options@fullpagetrue}%
8566         }%
8567         {\@msd@options@fullpagetrue}%
8568       }%
8569     }%
8570   \fi%
8571   \fi%
8572   \listxadd{\@msdata@list}{%

```

```

8573 \msd@options@iffullpage%
8574 \ifluatex%
8575 \csxdef{footnote@luatextextdir}{\the\textdir}%
8576 \csxdef{footnote@luatexpardir}{\the\pardir}%
8577 \fi%
8578 \csdef{@this@crossref@start}{\add@msd@cR:start:msdata}%
8579 \csdef{@this@crossref@end}{\add@msd@cR:end:msdata}%
8580 \unexpanded{%
8581 \def\annot@start{}%
8582 \def\annot@end{}%
8583 }%
8584 \noexpand\csuse{v\@msdata@series footnote}{\@msdata@series}{\@
expandonce\l@d@nums}{\ms@data@label}{\expandonce\data}}%
8585 \reset@msd@options@iffullpage%
8586 }%
8587 }%
8588 {}%
8589 \fi%
8590 \egroup%
8591 }%
8592 %

```

`\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.

```

8593 \newif\if@msdata@insertedfrompreviouspage%
8594 %

```

`\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.

```

8595 \newcommand{\add@msdata@firstlineofpage}{%
8596 \bgroup%
8597 \normalfont%
8598 \unless\ifledRcol{%
8599 \ifcsdef{@msdata@\add@msd@c @data}{%
8600 \ifnumless{\the\absline@num-\@one}{\csuse{@msdata@\add@msd@c
@stopabslinenumber}}%
8601 {%
8602 \global\@msdata@insertedfrompreviouspagetrue%
8603 \letcs{@data}{@msdata@\add@msd@c @data}%
8604 \edef\l@d@nums{%
8605 000|% Start page = we don't print it
8606 \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
8607 000|% Start subline number, for now, not used
8608 000|% End page number, we don't print it
8609 \ifnumless{\csuse{@msdata@\add@msd@c @stopabslinenumber}}{\csuse{
@lastabsline@forpage@\the\page@num}}%

```

```

8610         {\csuse{@msdata@\add@msd@c @stoplinenumber}}}%End line number if
      in the same page
8611         {\csuse{@lastline@forpage@\the\page@num}}}%Otherwise, last
number of the page
8612         |%
8613         000|}% End sub line number, for now, not used
8614         \edfont@info%Font
8615         }%
8616         \msd@options@fullpagefalse%
8617         \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
8618         {%
8619         \numdef{\@tmp}{\add@msd@c+\@one}%
8620         \ifcsdef{@msdata@\@tmp @abslinenumber}%
8621         {\ifnumequal{\csuse{@msdata@\@tmp @abslinenumber}}{\csuse{
@lastabsline@forpage@\the\page@num}}}%
8622         }%
8623         {\msd@options@fullpagetrue}%
8624         }%
8625         {\msd@options@fullpagetrue}%
8626         }%
8627         {}%
8628         \listxadd{\@msdata@list}{%
8629         \msd@options@iffullpage%
8630         \ifluatex%
8631         \csxdef{footnote@luatextextdir}{\the\textdir}%
8632         \csxdef{footnote@luatexpardir}{\the\pardir}%
8633         \fi%
8634         \csdef{@this@crossref@start}{\add@msd@c:start:msdata}%
8635         \csdef{@this@crossref@end}{\add@msd@c:end:msdata}%
8636         \unexpanded{%
8637         \def\annot@start{}%
8638         \def\annot@end{}}%
8639         }%
8640         \noexpand\csuse{v\@msdata@series footnote}{\@msdata@series}{\{
expandonce\l@d@nums}{\ms@data@label}{\expandonce\@data}}%
8641         \reset@msd@options@iffullpage%
8642         }%
8643         }%
8644         {\global\@msdata@insertedfrompreviouspagefalse}%
8645         }{}%
8646         \else%
8647         \ifcsdef{@msdata@\add@msd@cR @dataR}{%
8648         \ifnumless{\the\absline@numR-\@one}}{\csuse{@msdata@\add@msd@cR
@stopabslinenumberR}}}%
8649         {%
8650         \global\@msdata@insertedfrompreviouspagetrue%
8651         \letcs{\@data}{\@msdata@\add@msd@cR @dataR}%
8652         \edef\l@d@nums{%

```

```

8653         000| % Start page = we don't print it
8654         \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
8655         000| % Start subline number, for now, not used
8656         000| % End page number, we don't print it
8657         \ifnumless{\csuse{@msdata@\add@msd@cR @stopabslinenumberR}}{\csuse{@lastline@forpageR@\the\page@numR}}%
\csuse{@lastline@forpageR@\the\page@numR}}%
8658         {\csuse{@msdata@\add@msd@cR @stoplinenumberR}}%End line number
if in the same page
8659         {\csuse{@lastline@forpageR@\the\page@numR}}%Otherwise, last
number of the page
8660         |%
8661         000| % End sub line number, for now, not used
8662         \edfont@info%Font
8663         }%
8664         \@msd@options@fullpagefalse%
8665         \ifnumless{\csuse{@lastabsline@forpageR@\the\page@numR}}{\csuse{
@msdata@\add@msd@cR @stopabslinenumberR}+\@one}%
8666         {%
8667         \numdef{\@tmp}{\add@msd@cR+\@one}%
8668         \ifcsdef{@msdata@\@tmp @abslinenumberR}%
8669         {\ifnumequal{\csuse{@msdata@\@tmp @abslinenumberR}}{\csuse{
@lastabsline@forpageR@\the\page@numR}}}%
8670         }%
8671         {\@msd@options@fullpagetrue}%
8672         }%
8673         {\@msd@options@fullpagetrue}%
8674         }%
8675         {}%
8676         \listxadd{\@msdata@list}{%
8677         \@msd@options@iffullpage%
8678         \ifluatex%
8679         \csxdef{footnote@luatextextdir}{\the\textdir}%
8680         \csxdef{footnote@luatexpardir}{\the\pardir}%
8681         \fi%
8682         \csdef{@this@crossref@start}{\add@msd@cR:start:msdata}%
8683         \csdef{@this@crossref@end}{\add@msd@cR:end:msdata}%
8684         \unexpanded{%
8685         \def\annot@start{%
8686         \def\annot@end{%
8687         }%
8688         \noexpand\csuse{v\@msdata@series footnote}{\@msdata@series}{\{
expandonce\l@d@nums}{\ms@data@label}{\expandonce\data}}%
8689         \reset@msd@options@iffullpage%
8690         }%
8691         }%
8692         {\global\@msdata@insertedfrompreviouspagefalse}%
8693         }{}%
8694         \fi%

```

```

8695 \egroup%
8696 }%
8697 %

```

## 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%8 \newcommand{\insert@msdata}{%
8699 \def\do##1{##1}%
8700 \dolistloop{\@msdata@list}%
8701 \global\let\@msdata@list\relax%
8702 }%
8703 %

```

## 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.

```

8704 \newif\if@msd@options@fullpage%
8705 \newcommand{\@msd@options@iffullpage}[0]{%
8706 \if@msd@options@fullpage%
8707 \noexpand\toggletrue{nonum@}%
8708 \ifdefvoid{\ms@data@label}%
8709 {\noexpand\toggletrue{nosep@}}%
8710 }%
8711 \fi%
8712 }%
8713 \newcommand{\reset@msd@options@iffullpage}[0]{%
8714 \noexpand\togglefalse{nonum@}%
8715 \noexpand\togglefalse{nosep@}%
8716 }%
8717 %

```

## 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 `amsgen` package.

```
8718 \newtoks\@emptytoks
8719
8720 %
```

The rest is from `amsmath`.

`\l@denbody` A token register to contain the body.

```
8721 \newtoks\l@denbody
8722
8723 %
```

`\addtol@denbody` `\addtol@denbody{arg}` adds `arg` to the token register `\l@denbody`.

```
8724 \newcommand{\addtol@denbody}[1]{%
8725   \global\l@denbody\expandafter{\the\l@denbody#1}}
8726
8727 %
```

`\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`.

```
8728 \newcommand{\l@dcollect@body}[1]{%
8729   \l@denbody{\expandafter#1\expandafter{\the\l@denbody}}%
8730   \edef\processl@denbody{\the\l@denbody\noexpand\end{\@currenvir}}%
8731   \l@denbody\@emptytoks \def\l@dbegin@stack{b}%
8732   \begingroup
8733     \expandafter\let\csname\@currenvir\endcsname\l@dcollect@@body
8734     \edef\processl@denbody{\expandafter\noexpand\csname\@currenvir\endcsname}%
8735     \processl@denbody%
8736   }%
8737
8738 %
```

`\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.

```
8739 \def\l@dpush@begins#1\begin#2{%
8740   \ifx\end#2\else b\expandafter\l@dpush@begins\fi}
8741
8742 %
```

`\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.

```

8743 \def\l@dcollect@@body#1\end#2{%
8744   \edef\l@dbegin@stack{\l@dpush@begins#1\begin\end
8745                       \expandafter\@gobble\l@dbegin@stack}%
8746   \ifx\@empty\l@dbegin@stack
8747     \endgroup
8748     \@checkend{#2}%
8749     \addtol@denbody{#1}%
8750   \else
8751     \addtol@denbody{#1\end{#2}}%
8752   \fi
8753   \processl@denbody % A little tricky! Note the grouping
8754 }
8755
8756 %

```

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... 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.

```

8757 \newcommand*{\l@dtabnoexpands}{%
8758   \let\rtab=0%
8759   \let\ctab=0%
8760   \let\ltab=0%
8761   \let\rtabtext=0%
8762   \let\ltabtext=0%
8763   \let\ctabtext=0%
8764   \let\edbeforetab=0%
8765   \let\edaftertab=0%
8766   \let\edatleft=0%
8767   \let\edatright=0%
8768   \let\edvertline=0%
8769   \let\edvertdots=0%
8770   \let\edrowfill=0%
8771 }
8772 %
8773 %

```

`\disable@familiarnotes`    Macros to disable and restore familiar notes, to prevent them from printing multiple times in edtabularx and edarrayx environments.

`\restore@familiarnotes`

```

8774 \newcommand{\disable@familiarnotes}{%
8775   \unless\ifnofamiliar%
8776   \def\do##1{%
8777     \csletcs{footnote@@##1}{footnote##1}%
8778     \expandafter\renewcommand \csname footnote##1\endcsname[1]{%
8779       \protected@csxdef{@thefnmark##1}{\csuse{thefootnote##1}}%
8780       \csuse{@footnotemark##1}%
8781     }%
8782   }%
8783   \dolistloop{\@series}%
8784 \fi%
8785 }%
8786 \newcommand{\restore@familiarnotes}{%
8787   \unless\ifnofamiliar%
8788   \def\do##1{%
8789     \csletcs{footnote##1}{footnote@@##1}%
8790   }%
8791   \dolistloop{\@series}%
8792 \fi%
8793 }%
8794 %
8795 %

```

`\disable@sidenotes`    The same for sidenotes.

`\restore@sidenotes`

```

8796 \newcommand{\disable@sidenotes}{%
8797   \let\@@ledrightnote\ledrightnote%
8798   \let\@@ledleftnote\ledleftnote%
8799   \let\@@ledsidenote\ledsidenote%
8800   \let\ledrightnote\@gobble%
8801   \let\ledleftnote\@gobble%
8802   \let\ledsidenote\@gobble%
8803 }%
8804 \newcommand{\restore@sidenotes}{%
8805   \let\ledrightnote\@@ledrightnote%
8806   \let\ledleftnote\@@ledleftnote%
8807   \let\ledsidenote\@@ledsidenote%
8808 }%
8809 %

```

`\disable@endnotes`    The same for endnotes.

`\restore@endnotes`

```

8810 \newcommand{\disable@endnotes}{%
8811   \unless\ifnoend%
8812   \def\do##1{%
8813     \csletcs{##1@@endnote}{##1endnote}%
8814     \expandafter\renewcommand \csname ##1endnote\endcsname[1]{}%

```

```

8815 }%
8816 \dolistloop{\@series}%
8817 \fi%
8818 }%
8819 \newcommand{\restore@endnotes}{%
8820 \unless\ifnofamiliar@%
8821 \def\do##1{%
8822 \csletcs{##1endnote}{##1@endnote}%
8823 }%
8824 \dolistloop{\@series}%
8825 \fi%
8826 }%
8827 %

```

**\disable@notes** Disable/restore side, familiar and end notes.

```

\restore@notes
8828 \newcommand{\disable@notes}{%
8829 \disable@sidenotes%
8830 \disable@familiarnotes%
8831 \disable@endnotes%
8832 }%
8833 \newcommand{\restore@notes}{%
8834 \restore@sidenotes%
8835 \restore@familiarnotes%
8836 \restore@endnotes%
8837 }%
8838 %

```

**\EDTEXT** We need to be able to modify the `\edtext` macros and also restore their original definitions.

```

\edtext
8839 \let\EDTEXT=\edtext
8840 \newcommand{\xedtext}[2]{\EDTEXT{#1}{#2}}
8841 %

```

**\EDLABEL** We need to be able to modify and restore the `\edlabel` macro.

```

\xedlabel
8842 \let\EDLABEL=\edlabel
8843 \newcommand*\xedlabel[1]{\EDLABEL{#1}}
8844 %

```

**\xedindex** Macros supporting modification and restoration of `\edindex`.

```

\nulledindex
8845 \AtBeginDocument{\let\xedindex\edindex}%
8846 \newcommand{\nulledindex}[2][\jobname]{\@bsphack\@esphack}
8847
8848 %

```

**\@line@num** Macro supporting restoration of `\linenum`.

```
8849 \let\@line@num=\linenum
8850 %
```

**\l@dgobbleoptarg** \l@dgobbleoptarg[*arg*]{*arg*} replaces these two arguments (first is optional) by \relax.

```
8851 \newcommand*\l@dgobbleoptarg[2][\relax]%
8852 %
8853 %
```

**\l@secondmandarg** \l@secondoptarg[*arg*]{*arg*} gobble the first (optional) argument, and expand to the second (mandatory) argument.

```
8854 \NewExpandableDocumentCommand{\l@secondmandarg}{om}{#2}%
8855 %
```

```
\Relax56 \let\Relax=\relax
```

```
\NEXT57 \let\NEXT=\next
```

```
8858 %
8859 %
```

**\l@dmodforedtext** Modify and restore various macros for when \edtext is used.

**\l@drestoreforedtext**

```
8860 \newcommand{\l@dmodforedtext}{%
8861   \let\edtext\relax
8862   \def\do##1{\global\csletcs{##1footnote}{\l@dgobbleoptarg}}%
8863   \dolistloop{\@series}%
8864   \let\edindex\nulledindex
8865   \let\linenum@gobble}
8866 \newcommand{\l@drestoreforedtext}{%
8867   \def\do##1{\global\csletcs{##1footnote}{##1@footnote}}
8868   \dolistloop{\@series}%
8869   \let\edindex\xedindex}
8870 %
```

**\l@dnullfills** Nullify and restore some column fillers, etc.

**\l@drestorefills**

```
8871 \newcommand{\l@dnullfills}{%
8872   \def\edlabel##1{%
8873     \def\edrowfill##1##2##3{%
8874     }
8875   \newcommand{\l@drestorefills}{%
8876     \def\edrowfill##1##2##3{\@EDROWFILL@{##1}{##2}{##3}}%
8877   }
8878 %
8879 %
```

**\letsforverteilen** Gathers some lets and other code that is common to the \*verteilen\* macros.

```

8880 \newcommand{\letsforverteilen}{%
8881   \let\edtext\xedtext
8882   \let\edindex\xedindex
8883   \def\do##1{\global\csletcs{##1footnote}{##1@footnote}}
8884   \dolistloop{\@series}%
8885   \let\linenum\@line@num
8886   \hilfe skip=\l@dcwidth%
8887   \advance\hilfe skip by -\wd\hilfe box
8888   \def\edlabel##1{\xedlabel{##1}}
8889
8890 %

```

`\disablel@dtabfeet` Declarations for using or using `\edtext` inside tabulars. The default at this point is for  
`\enablel@dtabfeet` `\edtext`.

```

8891 \newcommand\disablel@dtabfeet{\l@dmodforedtext}%
8892 \newcommand\enablel@dtabfeet{\l@drestoreforedtext}%
8893 %

```

### XXXII.2.2 Counters, boxes and lengths

`\l@dampcount` `\l@dampcount` is a counter for the & column dividers and `\l@dcwidth` is a counter  
`\l@dcwidth` for the columns.

```

8894 \newcount\l@dampcount
8895   \l@dampcount=1\relax
8896 \newcount\l@dcwidth
8897   \l@dcwidth=0\relax
8898
8899 %

```

`\hilfe box` Some (temporary) helper items.

```

8900 \hilfe skip \newbox\hilfe box
8901 \Hilfe box \newskip\hilfe skip
8902 \hilfe count \newbox\Hilfe box
8903 \hilfe count \newcount\hilfe count
8904
8905 %

```

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).

```

8906 \newdimen\dcwidth
8907 \newdimen\dcwidthi
8908 \newdimen\dcwidthii
8909 \newdimen\dcwidthiii
8910 \newdimen\dcwidthiv
8911 \newdimen\dcwidthv
8912 \newdimen\dcwidthvi

```

```

8913 \newdimen\dcolviii
8914 \newdimen\dcolix
8915 \newdimen\dcolx
8916 \newdimen\dcolxi
8917 \newdimen\dcolxii
8918 \newdimen\dcolxiii
8919 \newdimen\dcolxiv
8920 \newdimen\dcolxv
8921 \newdimen\dcolxvi
8922 \newdimen\dcolxvii
8923 \newdimen\dcolxviii
8924 \newdimen\dcolxix
8925 \newdimen\dcolxx
8926 \newdimen\dcolxxi
8927 \newdimen\dcolxxii
8928 \newdimen\dcolxxiii
8929 \newdimen\dcolxxiv
8930 \newdimen\dcolxxv
8931 \newdimen\dcolxxvi
8932 \newdimen\dcolxxvii
8933 \newdimen\dcolxxviii
8934 \newdimen\dcolxxix
8935 \newdimen\dcolxxx
8936 \newdimen\dcolerr % added for error handling
8937
8938 %

```

**\l@dcolwidth** This is a cunning way of storing the columnwidths indexed by the column number \l@dcolcount, like an array. (was \Dimenzuordnung)

```

8939 \newcommand{\l@dcolwidth}{\ifcase \the\l@dcolcount \dcoli %???
8940 \or \dcoli \or \dcolii \or \dcoliii
8941 \or \dcoliv \or \dcolv \or \dcolvi
8942 \or \dcolvii \or \dcolviii \or \dcolix \or \dcolx
8943 \or \dcolxi \or \dcolxii \or \dcolxiii
8944 \or \dcolxiv \or \dcolxv \or \dcolxvi
8945 \or \dcolxvii \or \dcolxviii \or \dcolxix \or \dcolxx
8946 \or \dcolxxi \or \dcolxxii \or \dcolxxiii
8947 \or \dcolxxiv \or \dcolxxv \or \dcolxxvi
8948 \or \dcolxxvii \or \dcolxxviii \or \dcolxxix \or \dcolxxx
8949 \else \dcolerr \fi}
8950
8951 %

```

**\stepl@dcolcount** This increments the column counter, and issues an error message if it is too large.

```

8952 \newcommand*{\stepl@dcolcount}{\advance\l@dcolcount\@ne
8953 \ifnum\l@dcolcount>30\relax
8954 \led@err@TooManyColumns
8955 \fi}

```

```
8956
8957 %
```

**\l@setmaxcolwidth** Sets the column width to the maximum value seen so far.

```
8958 \newcommand{\l@setmaxcolwidth}{%
8959   \ifdim\l@dcolwidth < \wd\hilfsbox
8960     \l@dcolwidth = \wd\hilfsbox
8961   \else \relax \fi}
8962
8963 %
```

**\measuremcell** Measure (recursively) the width required for a math cell.

```
8964 \def\measuremcell #1{%
8965   \ifx #1\ \ifnum\l@dcolcount=0\let\NEXT\relax%
8966     \else\l@dccheckcols%
8967       \l@dcolcount=0%
8968       \let\NEXT\measuremcell%
8969     \fi%
8970   \else\setbox\hilfsbox=\hbox{$\displaystyle{#1}$}%
8971     \step\l@dcolcount%
8972     \l@setmaxcolwidth%
8973     \let\NEXT\measuremcell%
8974   \fi\NEXT}
8975
8976 %
```

**\measuretcell** Measure (recursively) the width required for a text cell.

```
8977 \def\measuretcell #1{%
8978   \ifx #1\ \ifnum\l@dcolcount=0\let\NEXT\relax%
8979     \else\l@dccheckcols%
8980       \l@dcolcount=0%
8981       \let\NEXT\measuretcell%
8982     \fi%
8983   \else\setbox\hilfsbox=\hbox{#1}%
8984     \step\l@dcolcount%
8985     \l@setmaxcolwidth%
8986     \let\NEXT\measuretcell%
8987   \fi\NEXT}
8988
8989 %
```

**\measuremrow** Measure (recursively) the width required for a math row.

```
8990 \def\measuremrow #1{%
8991   \ifx #1\ \let\NEXT\relax%
8992   \else\measuremcell #1\&\&\&%
8993     \let\NEXT\measuremrow%
```

```

8994 \fi\NEXT}
8995 %

```

**\measuretrow** Measure (recursively) the width required for a text row.

```

8996 \def\measuretrow #1\{\%
8997 \ifx #1&\let\NEXT\relax%
8998 \else\measuretcell #1&\&\&\&%
8999 \let\NEXT\measuretrow%
9000 \fi\NEXT}
9001
9002 %

```

**\edtabcolsep** The length \edtabcolsep controls the distance between columns.

```

9003 \newskip\edtabcolsep
9004 \global\edtabcolsep=10pt
9005
9006 %

```

```

\variab07 \newcommand{\variab}{\relax}
9008
9009 %

```

**\l@dcheckcols** Check that the number of columns is consistent.

```

9010 \newcommand*{\l@dcheckcols}{\%
9011 \ifnum\l@dcolcount=1\relax
9012 \else
9013 \ifnum\l@dampcount=1\relax
9014 \else
9015 \ifnum\l@dcolcount=\l@dampcount\relax
9016 \else
9017 \l@d@err@UnequalColumns
9018 \fi
9019 \fi
9020 \l@dampcount=\l@dcolcount
9021 \fi}
9022
9023 %

```

**\edfilldimen** A length.

```

9024 \newdimen\edfilldimen
9025 \edfilldimen=0pt
9026
9027 %

```



`\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`

```

9028 \newcounter{addcolcount}
9029 \renewcommand{\theadcolcount}{\romannumeral \c@addcolcount}
9030 %

```

### XXXII.2.3 Tabular typesetting

`\setmcellright` Typeset (recursively) cells of display math right justified.

```

9031 \def\setmcellright #1{\def\edlabel##1{}%
9032   \let\edindex\nulledindex
9033   \ifx #1\ \ifnum\l@dcolcount=0%\removelastskip
9034     \let\Next\relax%
9035   \else\l@dcolcount=0%
9036     \let\Next=\setmcellright%
9037   \fi%
9038   \else%
9039     \disablel@dtabfeet%
9040     \step1@dcolcount%
9041     \disable@notes%
9042     \setbox\hilfsbox=\hbox{$\displaystyle{#1}$}%
9043     \restore@notes%
9044     \letsforverteilen%
9045     \hskip\hilfsskip$\displaystyle{#1}$%
9046     \hskip\edtabcolsep%
9047     \let\Next=\setmcellright%
9048   \fi\Next}
9049 %
9050 %

```

`\settcclright` Typeset (recursively) cells of text right justified.

```

9051 \def\settcclright #1{\def\edlabel##1{}%
9052   \let\edindex\nulledindex
9053   \ifx #1\ \ifnum\l@dcolcount=0%\removelastskip
9054     \let\Next\relax%
9055   \else\l@dcolcount=0%
9056     \let\Next=\settcclright%
9057   \fi%
9058   \else%
9059     \disablel@dtabfeet%
9060     \step1@dcolcount%
9061     \disable@notes%
9062     \setbox\hilfsbox=\hbox{#1}%
9063     \restore@notes%
9064     \letsforverteilen%
9065     \hskip\hilfsskip#1%

```

```

9066         \hskip\edtabcolsep%
9067         \let\Next=\settccllright%
9068     \fi\Next}
9069 %

```

**\setmcellleft** Typeset (recursively) cells of display math left justified.

```

9070 \def\setmcellleft #1&{\def\edlabel##1{}%
9071     \let\edindex\nulledindex
9072     \ifx #1\\ \ifnum\l@dc@colcount=0 \let\Next\relax%
9073         \else\l@dc@colcount=0%
9074         \let\Next=\setmcellleft%
9075     \fi%
9076 \else \disablel@dtabfeet%
9077     \stepl@dc@colcount%
9078     \disable@notes%
9079     \setbox\hilfsbox=\hbox{$\displaystyle{#1}$}%
9080     \restore@notes%
9081     \letsforverteilen%
9082     $\displaystyle{#1}$\hskip\hilfsskip\hskip\edtabcolsep%
9083     \let\Next=\setmcellleft%
9084 \fi\Next}
9085 %
9086 %

```

**\settcclleft** Typeset (recursively) cells of text left justified.

```

9087 \def\settcclleft #1&{\def\edlabel##1{}%
9088     \let\edindex\nulledindex
9089     \ifx #1\\ \ifnum\l@dc@colcount=0 \let\Next\relax%
9090         \else\l@dc@colcount=0%
9091         \let\Next=\settcclleft%
9092     \fi%
9093 \else \disablel@dtabfeet%
9094     \stepl@dc@colcount%
9095     \disable@notes%
9096     \setbox\hilfsbox=\hbox{#1}%
9097     \restore@notes%
9098     \letsforverteilen%
9099     #1\hskip\hilfsskip\hskip\edtabcolsep%
9100     \let\Next=\settcclleft%
9101 \fi\Next}
9102 %

```

**\setmcellcenter** Typeset (recursively) cells of display math centered.

```

9103 \def\setmcellcenter #1&{\def\edlabel##1{}%
9104     \let\edindex\nulledindex
9105     \ifx #1\\ \ifnum\l@dc@colcount=0\let\Next\relax%
9106         \else\l@dc@colcount=0%

```

```

9107         \let\Next=\setmcellcenter%
9108     \fi%
9109 \else    \disablel@dtabfeet%
9110         \stepl@dcolcount%
9111         \disable@notes%
9112         \setbox\hilfsbox=\hbox{$\displaystyle{#1}$}%
9113         \restore@notes%
9114         \letsforverteilen%
9115         \hskip 0.5\hilfsskip$\displaystyle{#1}$\hskip0.5\hilfsskip%
9116         \hskip\edtabcolsep%
9117         \let\Next=\setmcellcenter%
9118     \fi\Next}
9119
9120 %

```

**\settcellcenter** Typeset (recursively) cells of text centered.

```

9121 \def\settcellcenter #1&{\def\edlabel##1{}%
9122     \let\edindex\nulledindex
9123     \ifx #1\\ \ifnum\l@dcolcount=0 \let\Next\relax%
9124         \else\l@dcolcount=0%
9125             \let\Next=\settcellcenter%
9126         \fi%
9127     \else    \disablel@dtabfeet%
9128             \stepl@dcolcount%
9129             \disable@notes%
9130             \setbox\hilfsbox=\hbox{#1}%
9131             \restore@notes%
9132             \letsforverteilen%
9133             \hskip 0.5\hilfsskip #1\hskip 0.5\hilfsskip%
9134             \hskip\edtabcolsep%
9135             \let\Next=\settcellcenter%
9136         \fi\Next}
9137
9138 %

```

**\NEXT** <sub>39</sub> \let\NEXT=\relax

```

9140
9141 %

```

**\setmrowright** Typeset (recursively) rows of right justified math.

```

9142 \def\setmrowright #1\\{%
9143     \ifx #1& \let\NEXT\relax
9144     \else \centerline{\setmcellright #1&\\&\\&}
9145         \let\NEXT=\setmrowright
9146     \fi\NEXT}
9147 %

```

`\settroright` Typeset (recursively) rows of right justified text.

```

9148 \def\settroright #1\\{%
9149     \ifx #1& \let\NEXT\relax
9150     \else \centerline{\settcellright #1&\\&\\&}
9151         \let\NEXT=\settroright
9152     \fi\NEXT}
9153
9154 %

```

`\setmrowleft` Typeset (recursively) rows of left justified math.

```

9155 \def\setmrowleft #1\\{%
9156     \ifx #1& \let\NEXT\relax
9157     \else \centerline{\setmcellleft #1&\\&\\&}
9158         \let\NEXT=\setmrowleft
9159     \fi\NEXT}
9160 %

```

`\settrorleft` Typeset (recursively) rows of left justified text.

```

9161 \def\settrorleft #1\\{%
9162     \ifx #1& \let\NEXT\relax
9163     \else \centerline{\settcellleft #1&\\&\\&}
9164         \let\NEXT=\settrorleft
9165     \fi\NEXT}
9166
9167 %

```

`\setmrowcenter` Typeset (recursively) rows of centered math.

```

9168 \def\setmrowcenter #1\\{%
9169     \ifx #1& \let\NEXT\relax%
9170     \else \centerline{\setmcellcenter #1&\\&\\&}
9171         \let\NEXT=\setmrowcenter
9172     \fi\NEXT}
9173 %

```

`\settrorcenter` Typeset (recursively) rows of centered text.

```

9174 \def\settrorcenter #1\\{%
9175     \ifx #1& \let\NEXT\relax
9176     \else \centerline{\settcellcenter #1&\\&\\&}
9177         \let\NEXT=\settrorcenter
9178     \fi\NEXT}
9179
9180 %

```

```

\newcommand{\nullsetzen}{%
  \step1@dcolcount%
  \l@dcolwidth=0pt%
  \ifnum\l@dcolcount=30\let\NEXT\relax%
    \l@dcolcount=0\relax
  \else\let\NEXT\nullsetzen%
  \fi\NEXT}
%
```

**\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.

```

\newcommand{\edatleft}[3][\@empty]{%
  \ifx#1\@empty
    \vbox to 10pt{\vss\hbox{$\left#2\vrule width0pt height #3
      depth 0pt \right. $\hss}\vfil}
  \else
    \vbox to 4pt{\vss\hbox{#1\left#2\vrule width0pt height #3
      depth 0pt \right. $\vfil}
  \fi}
%
```

**\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.

```

\newcommand{\edatright}[3][\@empty]{%
  \ifx#1\@empty
    \vbox to 10pt{\vss\hbox{$\left.\vrule width0pt height #3
      depth 0pt \right#2 $\hss}\vfil}
  \else
    \vbox to 4pt{\vss\hbox{$\left.\vrule width0pt height #3
      depth 0pt \right#2 #1 $\vfil}
  \fi}
%
```

**\edvertline** `\edvertline{ $\langle len \rangle$ }` vertical line  $\langle len \rangle$  high.

```

\newcommand{\edvertline}[1]{\vbox to 8pt{\vss\hbox{\vrule height #1}\vfil}}
%
```

**\edvertdots** `\edvertdots{ $\langle len \rangle$ }` vertical dotted line  $\langle len \rangle$  high.

```

\newcommand{\edvertdots}[1]{\vbox to 1pt{\vss\vbox to #1%
  {\cleaders\hbox{$\math\hbox{.}\vbox to 0.5em{ }\vfil}}}
%
```

`\l@dtabaddcols` `\l@dtabaddcols{<startcol>}{<endcol>}` adds the widths of the columns `<startcol>` through `<endcol>` to `\edfilldimen`. It is a  $\text{\TeX}$  style reimplementation of the original `\@add@`.

```

9216 \newcommand{\l@dtabaddcols}[2]{%
9217   \l@dcheckstartend{#1}{#2}%
9218   \ifl@dstartendok
9219   \setcounter{addcolcount}{#1}%
9220   \@whilenum \value{addcolcount}<#2\relax \do
9221   {\advance\edfilldimen by \the\csname dcol\theaddcolcount\endcsname
9222    \advance\edfilldimen by \edtabcolsep
9223    \stepcounter{addcolcount}}%
9224   \advance\edfilldimen by \the\csname dcol\theaddcolcount\endcsname
9225   \fi
9226 }
9227
9228 %

```

`\ifl@dstartendok` `\l@dcheckstartend{<startcol>}{<endcol>}` checks that the values of `<startcol>` and `<endcol>` are sensible. If they are then `\ifl@dstartendok` is set TRUE, otherwise it is set FALSE.

```

9229 \newif\ifl@dstartendok
9230 \newcommand{\l@dcheckstartend}[2]{%
9231   \l@dstartendoktrue
9232   \ifnum #1<\@ne
9233     \l@dstartendokfalse
9234     \led@err@LowStartColumn
9235   \fi
9236   \ifnum #2>30\relax
9237     \l@dstartendokfalse
9238     \led@err@HighEndColumn
9239   \fi
9240   \ifnum #1>#2\relax
9241     \l@dstartendokfalse
9242     \led@err@ReverseColumns
9243   \fi
9244 }
9245
9246 %

```

`\edrowfill` `\edrowfill{<startcol>}{<endcol>}` fill fills columns `<startcol>` to `<endcol>` inclusive with `<fill>` (e.g. `\hrulefill`, `\upbracefill`). This is a  $\text{\TeX}$  style reimplementation and generalization of the original `\waklam`, `\Waklam`, `\waklamec`, `\wastricht` and `\wapunktel` macros.

```

9247 \newcommand*{\edrowfill}[3]{%
9248   \l@dtabaddcols{#1}{#2}%
9249   \hb@xt@ \the\l@dcolwidth{\hb@xt@ \the\edfilldimen{#3}\hss}}
9250 \let\@edrowfill@=\edrowfill
9251 \def\@EDROWFILL@#1#2#3{\@edrowfill@{#1}{#2}{#3}}

```

9252  
9253 %

**\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`.

```
9254 \newcommand{\leftltab}[1]{%
9255   \hb@xt@z@{\vbox{\edtabindent%
9256     \moveleft\Hilfsskip\hbox{\ #1}}\hss}}
9257
9258 %
```

**\leftrtab**     `\leftrtab{⟨text⟩}{⟨math⟩}` for `\edbeforetab` in `\rtab`.

```
9259 \newcommand{\leftrtab}[2]{%
9260   #2\hb@xt@z@{\vbox{\edtabindent%
9261     \advance\Hilfsskip by\dcoli%
9262     \moveleft\Hilfsskip\hbox{\ #1}}\hss}}
9263
9264 %
```

**\leftctab**     `\leftctab{⟨text⟩}{⟨math⟩}` for `\edbeforetab` in `\ctab`.

```
9265 \newcommand{\leftctab}[2]{%
9266   \hb@xt@z@{\vbox{\edtabindent\l@dcolcount=\l@dampcount%
9267     \advance\Hilfsskip by 0.5\dcoli%
9268     \setbox\hilfsbox=\hbox{\def\edlabel##1}%
9269     \disablel@dtabfeet$\displaystyle{#2}$}%
9270     \advance\Hilfsskip by -0.5\wd\hilfsbox%
9271     \moveleft\Hilfsskip\hbox{\ #1}}\hss}%
9272   #2}
9273
9274 %
```

**\rightctab**     `\rightctab{⟨math⟩}{⟨text⟩}` for `\edaftertab` in `\ctab`.

```
9275 \newcommand{\rightctab}[2]{%
9276   \setbox\hilfsbox=\hbox{\def\edlabel##1}%
9277   \disablel@dtabfeet#2}\l@dampcount=\l@dcolcount%
9278   #1\hb@xt@z@{\vbox{\edtabindent\l@dcolcount=\l@dampcount%
9279     \advance\Hilfsskip by 0.5\l@dcolwidth%
9280     \advance\Hilfsskip by -\wd\hilfsbox%
9281     \setbox\hilfsbox=\hbox{\def\edlabel##1}%
9282     \disablel@dtabfeet$\displaystyle{#1}$}%
9283     \advance\Hilfsskip by -0.5\wd\hilfsbox%
```

```

9284 \advance\Hilfsskip by \edtabcolsep%
9285 \moveright\Hilfsskip\hbox{ #2}}\hss}%
9286 }
9287
9288 %

```

**\rightltab** `\rightltab{<math>}<{<text>}` for `\edaftertab` in `\ltab`.

```

9289 \newcommand{\rightltab}[2]{%
9290 \setbox\hilfsbox=\hbox{\def\edlabel##1{%
9291 \disablel@dtabfeet#2}\l@dampcount=\l@dcolcount%
9292 #1\hb@xt@{\z@{\vbox{\edtabindent\l@dcolcount=\l@dampcount%
9293 \advance\Hilfsskip by\l@dcolwidth%
9294 \advance\Hilfsskip by-\wd\hilfsbox%
9295 \setbox\hilfsbox=\hbox{\def\edlabel##1{%
9296 \disablel@dtabfeet$\displaystyle{#1}$}%
9297 \advance\Hilfsskip by-\wd\hilfsbox%
9298 \advance\Hilfsskip by\edtabcolsep%
9299 \moveright\Hilfsskip\hbox{ #2}}\hss}%
9300 }
9301
9302 %

```

**\rightrtab** `\rightrtab{<math>}<{<text>}` for `\edaftertab` in `\rtab`.

```

9303 \newcommand{\rightrtab}[2]{%
9304 \setbox\hilfsbox=\hbox{\def\edlabel##1{%
9305 \disablel@dtabfeet#2}%
9306 #1\hb@xt@{\z@{\vbox{\edtabindent%
9307 \advance\Hilfsskip by-\wd\hilfsbox%
9308 \advance\Hilfsskip by\edtabcolsep%
9309 \moveright\Hilfsskip\hbox{ #2}}\hss}%
9310 }
9311
9312 %

```

**\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.

```

9313 \newcommand{\rtab}[1]{%
9314 \l@dnnullfills
9315 \def\edbeforetab##1##2{\lefttab{##1}{##2}}%
9316 \def\edaftertab##1##2{\righttab{##1}{##2}}%
9317 \measurebody{#1}%
9318 \l@dretofills
9319 \variab
9320 \setmrowright #1\\&\\%
9321 \enablel@dtabfeet}
9322
9323 %

```



`\measurebody` `\measurebody{⟨body⟩}` measures the array `⟨body⟩`.

```

9324 \newcommand{\measurebody}[1]{%
9325   \disablel@dtabfeet%
9326   \l@dc@count=0%
9327   \nullsetzer%
9328   \l@dc@count=0
9329   \measuremrow #1\\&\\%
9330   \global\l@dampcount=1}
9331
9332 %

```

`\rtabtext` `\rtabtext{⟨body⟩}` typesets `⟨body⟩` as a tabular with the entries right justified.

```

9333 \newcommand{\rtabtext}[1]{%
9334   \l@dn@fills
9335   \measuretbody{#1}%
9336   \l@dre@fills
9337   \variab
9338   \settr@right #1\\&\\%
9339   \enablel@dtabfeet}
9340
9341 %

```

`\measuretbody` `\measuretbody{⟨body⟩}` measures the tabular `⟨body⟩`.

```

9342 \newcommand{\measuretbody}[1]{%
9343   \disable@notes%
9344   \disablel@dtabfeet%
9345   \l@dc@count=0%
9346   \nullsetzer%
9347   \l@dc@count=0
9348   \measuretbody #1\\&\\%
9349   \restore@notes%
9350   \global\l@dampcount=1}
9351
9352 %

```

`\ltab` Array with entries left justified.

```

\edbeforetab \newcommand{\ltab}[1]{%
\edaftertab  \l@dn@fills
9353
9354   \def\edbeforetab##1##2{\leftltab{##1}{##2}}%
9355   \def\edaftertab##1##2{\rightltab{##1}{##2}}%
9356   \measuretbody{#1}%
9357   \l@dre@fills
9358   \variab
9359   \setmrowleft #1\\&\\%
9360   \enablel@dtabfeet}
9361
9362 %
9363

```

**\ltabtext** Tabular with entries left justified.

```

9364 \newcommand{\ltabtext}[1]{%
9365   \l@dnnullfills
9366   \measuretbody{#1}%
9367   \l@drestorefills
9368   \variab
9369   \settrorleft #1\&\%
9370   \enablel@dtabfeet}
9371
9372 %

```

**\ctab** Array with centered entries.

```

\edbeforetab
\edaftertab
9373 \newcommand{\ctab}[1]{%
9374   \l@dnnullfills
9375   \def\edbeforetab##1##2{\leftctab{##1}{##2}}%
9376   \def\edaftertab##1##2{\rightctab{##1}{##2}}%
9377   \measuretbody{#1}%
9378   \l@drestorefills
9379   \variab
9380   \setmrowcenter #1\&\%
9381   \enablel@dtabfeet}
9382
9383 %

```

**\ctabtext** Tabular with entries centered.

```

9384 \newcommand{\ctabtext}[1]{%
9385   \l@dnnullfills
9386   \measuretbody{#1}%
9387   \l@drestorefills
9388   \variab
9389   \settrorcenter #1\&\%
9390   \enablel@dtabfeet}
9391
9392 %

```

```

\spreadtext
9393 \newcommand{\spreadtext}[1]{%\l@dcolcount=\l@dampcount%
9394   \hb@xt@ \the\l@dcolwidth{\hbox{#1}\hss}}
9395 %

```

```

\spreadmath
9396 \newcommand{\spreadmath}[1]{%
9397   \hb@xt@ \the\l@dcolwidth{\hbox{$\displaystyle{#1}$}\hss}}
9398
9399 %

```

**\HILFSskip** More helpers.

**\Hilfsskip**

```

9400 \newskip\HILFSskip
9401 \newskip\Hilfsskip
9402
9403 %

```

```

\EDTABINDENT \newcommand{\EDTABINDENT}{%
9404
9405   \ifnum\l@dc@lcount=30\let\NEXT\relax\l@dc@lcount=0%
9406   \else\step\l@dc@lcount%
9407     \advance\Hilfsskip by\l@dc@lwidth%
9408     \ifdim\l@dc@lwidth=0pt\advance\hilfsc@unt\@ne
9409     \else\advance\Hilfsskip by \the\hilfsc@unt\edtabcolsep%
9410     \hilfsc@unt=1\fi%
9411     \let\NEXT=\EDTABINDENT%
9412   \fi\NEXT}%
9413 %

```

`\edtabindent` (was `\tabindent`)

```

9414 \newcommand{\edtabindent}{%
9415   \l@dc@lcount=0\relax
9416   \Hilfsskip=0pt%
9417   \hilfsc@unt=1\relax
9418   \EDTABINDENT%
9419   \hilfsskip=\hsize%
9420   \advance\hilfsskip -\Hilfsskip%
9421   \Hilfsskip=0.5\hilfsskip%
9422 }%
9423
9424 %

```

`\EDTAB` (was `\TAB`)

```

9425 \def\EDTAB #1|#2|{%
9426   \setbox\tabhilfbox=\hbox{$\displaystyle{#1}$}%
9427   \setbox\tabHilfbox=\hbox{$\displaystyle{#2}$}%
9428   \advance\tabelskip -\wd\tabhilfbox%
9429   \advance\tabelskip -\wd\tabHilfbox%
9430   \unhbox\tabhilfbox\hskip\tabelskip%
9431   \unhbox\tabHilfbox}%
9432
9433 %

```

`\EDTABtext` (was `\TABtext`)

```

9434 \def\EDTABtext #1|#2|{%
9435   \setbox\tabhilfbox=\hbox{#1}%
9436   \setbox\tabHilfbox=\hbox{#2}%
9437   \advance\tabelskip -\wd\tabhilfbox%
9438   \advance\tabelskip -\wd\tabHilfbox%

```

```

9439 \unhbox\tabhilfbox\hskip\tabelskip%
9440 \unhbox\tabHilfbox}%
9441 %

```

`\tabhilfbox` Further helpers.

```

\tabHilfbox
9442 \newbox\tabhilfbox
9443 \newbox\tabHilfbox
9444
9445 %

```

### XXXII.2.4 Environments

`edarrayl edarrayc edarrayr` The ‘environment’ forms for `\ltab`, `\ctab` and `\rtab`.

```

9446 \newenvironment{edarrayl}{\l@collect@body\ltab}{\}
9447 \newenvironment{edarrayc}{\l@collect@body\ctab}{\}
9448 \newenvironment{edarrayr}{\l@collect@body\rtab}{\}
9449
9450 %

```

`edtabularl edtabularc edtabularr` The ‘environment’ forms for `\ltabtext`, `\ctabtext` and `\rtabtext`.

```

9451 \newenvironment{edtabularl}{\l@collect@body\ltabtext}{\}
9452 \newenvironment{edtabularc}{\l@collect@body\ctabtext}{\}
9453 \newenvironment{edtabularr}{\l@collect@body\rtabtext}{\}
9454
9455 %

```

## 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
9456 \newcommand{\initnumbering@quote}{
9457 \ifnoquotation@else
9458 \renewcommand{\quotation}{\par\leavevmode%
9459 \parindent=1.5em%
9460 \skipnumbering%
9461 \ifautopar%
9462 \vskip-\parskip%
9463 \else%
9464 \vskip\topsep%
9465 \fi%
9466 \global\leftskip=\leftmargin%

```

```

9467         \global\rightskip=\leftmargin%
9468     }
9469     \renewcommand{\endquotation}{\par%
9470         \global\leftskip=0pt%
9471         \global\rightskip=0pt%
9472         \leavevmode%
9473         \skipnumbering%
9474         \ifautopar%
9475             \vskip-\parskip%
9476         \else%
9477             \vskip\topsep%
9478         \fi%
9479     }
9480     \renewcommand{\quote}{\par\leavevmode%
9481         \parindent=0pt%
9482         \skipnumbering%
9483         \ifautopar%
9484             \vskip-\parskip%
9485         \else%
9486             \vskip\topsep%
9487         \fi%
9488         \global\leftskip=\leftmargin%
9489         \global\rightskip=\leftmargin%
9490     }
9491     \renewcommand{\endquote}{\par%
9492         \global\leftskip=0pt%
9493         \global\rightskip=0pt%
9494         \leavevmode%
9495         \skipnumbering%
9496         \ifautopar%
9497             \vskip-\parskip%
9498         \else%
9499             \vskip\topsep%
9500         \fi%
9501     }
9502     \fi
9503 }
9504 %

```

## XXXIV Section's title commands

### XXXIV.1 Commands to disable some feature

`\ledsectnotoc` The `\ledsectnotoc` only disables the `\addcontentsline` macro.

```

9505 \newcommand{\ledsectnotoc}{\let\addcontentsline\@gobblethree}
9506 %

```

`\ledsectnomark` The `\ledsectnomark` only disables the `\chaptermark`, `\sectionmark` and `\subsectionmark` macros.

```

9507 \newcommand{\ledsectnomark}{%
9508   \let\chaptermark\@gobble%
9509   \let\sectionmark\@gobble%
9510   \let\subsectionmark\@gobble%
9511 }
9512 %

```

## 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  $\text{\TeX}$  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  $\text{\TeX}$  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).

```

9513 \catcode`\#=12
9514 \notbool{@noeled@sec}{%
9515 %

```

`\beforeeledchapter` For technical reasons, not yet solved, page-breaking before chapters can't be made automatically by `eledmac`. Users have to use `\beforeeledchapter`.

```

9516 \ifl@dmemoir
9517   \newcommand\beforeeledchapter{%
9518     \clearforchapter%
9519   }
9520 \else
9521   \newcommand\beforeeledchapter{%
9522     \if@openright%
9523       \cleardoublepage%
9524     \else%
9525       \clearpage%
9526     \fi%
9527   }
9528 \fi
9529 %

```

## 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 line numbers. They include tests for RTL languages.

```

9530 \def\print@rightmargin@eledsection{%
9531   \if@eled@sectioning%
9532     \begingroup%
9533     \if@RTL%
9534       \let\llap\rlap%
9535       \let\leftlinenum\rightlinenum%
9536       \let\leftlinenumR\rightlinenumR%
9537       \let\l@drd@ta\l@dld@ta%
9538       \let\l@drsn@te\l@dlsn@te%
9539     \fi%
9540     \hfill\l@drd@ta \csuse{LR}{\l@drsn@te}%
9541     \endgroup%
9542   \fi%
9543 }%
9544
9545 \def\print@leftmargin@eledsection{%
9546   \if@eled@sectioning%
9547     \leavevmode%
9548     \begingroup%
9549     \if@RTL%
9550       \let\rlap\llap%
9551       \let\rightlinenum\leftlinenum%
9552       \let\rightlinenumR\leftlinenumR%
9553       \let\l@dld@ta\l@drd@ta%
9554       \let\l@dlsn@te\l@drsn@te%
9555     \fi%

```

```

9556 \l@ddld@ta\csuse{LR}{\l@dlsn@te}%
9557 \endgroup%
9558 \fi%
9559 }%
9560
9561 %

```

### XXXIV.5 Patching standard commands

`\M@sect` We have to patch `LATEX`, book and memoir sectioning commands in order to:

- Disable `\edtext` inside.
- Disable page breaking (for `\chapter`).
- Add line numbers and sidenotes.

Unfortunately, Maïeul Rouquette was not able to try if memoir is loaded. That is why `eledmac` tries to define for both standard class and memoir class.

```

9562 \AtBeginDocument{%
9563
9564
9565 \pretocmd{\M@sect}
9566   {\let\old@edtext=\edtext%
9567    \let\edtext=\dummy@edtext@showlemma%
9568   }
9569 {}
9570 {}
9571
9572 \apptocmd{\M@sect}
9573   {\let\edtext=\old@edtext}
9574 {}
9575 {}
9576
9577 \patchcmd{\M@sect}
9578   { #9}
9579   { #9%
9580    \print@rightmargin@eledsection%
9581   }
9582 {}
9583 {}
9584
9585 \patchcmd{\M@sect}
9586   {\hskip #3\relax}
9587   {\hskip #3\relax%
9588    \print@leftmargin@eledsection%
9589   }
9590 {}
9591 {}

```



```

9592 \patchcmd{\@mem@old@ssect}
9593   {\#5}
9594   {\#5%
9595   \print@leftmargin@eledsection%
9596   }
9597   {}
9598   {}
9599   {}
9600
9601 \patchcmd{\@mem@old@ssect}
9602   {\hskip #1}
9603   {\hskip #1%
9604   \print@rightmargin@eledsection%
9605   }
9606   {}
9607   {}
9608
9609
9610
9611 \patchcmd{\scr@startchapter}{\if@openright\cleardoublepage\else\clearpage\fi}{%
9612   \if@eled@sectioning\else%
9613   \ifl@dprintingpages\else%
9614   \if@openright\cleardoublepage\else\clearpage\fi}%No clearpage inside a
9615   \Pages: will keep critical notes from printing on the title page. Here for
9616   scrbook.
9617   \fi%
9618   \fi%
9619   }
9620   {}
9621   {}
9622
9623 \patchcmd{\@makechapterhead}
9624   {\#1}
9625   {\print@leftmargin@eledsection%
9626   #1%
9627   \print@rightmargin@eledsection%
9628   }
9629   {}
9630   {}
9631
9632 \patchcmd{\@makechapterhead}% For BIDI
9633   {\if@RTL\raggedleft\else\raggedright\fi}%
9634   {\if@eled@sectioning\else%
9635   \if@RTL\raggedleft\else\raggedright\fi%
9636   \fi%
9637   }%
9638   {}%
9639   {}%

```

```

9639 \patchcmd{\@makeschapterhead}
9640   {#1}
9641   {\print@leftmargin@eledsection%
9642     #1%
9643     \print@rightmargin@eledsection%
9644   }
9645   {}
9646   {}
9647
9648 \pretocmd{\@sect}
9649   {\let\old@edtext=\edtext
9650     \let\edtext=\dummy@edtext@showlemma%
9651   }
9652   {}
9653   {}
9654
9655 \apptocmd{\@sect}
9656   {\let\edtext=\old@edtext}
9657   {}
9658   {}
9659
9660 \pretocmd{\@ssect}
9661   {\let\old@edtext=\edtext%
9662     \let\edtext=\dummy@edtext@showlemma%
9663   }
9664   {}
9665   {}
9666
9667 \apptocmd{\@ssect}
9668   {\let\edtext=\old@edtext}
9669   {}
9670   {}
9671
9672 %

```

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.

```

9673 \@ifpackageloaded{nameref}{
9674
9675   \patchcmd{\NR@sect}
9676     {#8}
9677     {#8%
9678       \print@rightmargin@eledsection%
9679     }
9680     {}
9681     {}
9682
9683   \patchcmd{\NR@sect}
9684     {\hskip #3\relax}
9685     {\hskip #3\relax%

```

```

9686 \print@leftmargin@eledsection%
9687 }
9688 {}
9689 {}
9690
9691 \patchcmd{\NR@ssect}
9692   {#5}
9693   {#5%
9694   \print@rightmargin@eledsection%
9695   }
9696   {}
9697   {}
9698
9699 \patchcmd{\NR@ssect}
9700   {\hskip #1}
9701   {\hskip #1%
9702   \print@leftmargin@eledsection%
9703   }
9704   {}
9705   {}
9706 }%
9707 {
9708 \patchcmd{\@sect}
9709   {#8}
9710   {#8%
9711   \print@rightmargin@eledsection%
9712   }
9713   {}
9714   {}
9715
9716 \patchcmd{\@sect}
9717   {\hskip #3\relax}
9718   {\hskip #3\relax%
9719   \print@leftmargin@eledsection%
9720   }
9721   {}
9722   {}
9723
9724 \patchcmd{\@ssect}
9725   {#5}
9726   {#5%
9727   \print@rightmargin@eledsection%
9728   }
9729   {}
9730   {}
9731
9732 \patchcmd{\@ssect}
9733   {\hskip #1}
9734   {\hskip #1%
9735   \print@leftmargin@eledsection%

```

```

9736 }
9737 {}
9738 {}
9739 }%
9740 }%
9741 %

```

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 `#`.

```

9742 {}}%
9743 \protect\catcode`\#=6 %Space NEEDS by \catcode
9744 %

```

**\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.

```

9745 \AtBeginDocument{%
9746 \patchcmd{\chapter}{\clearforchapter}{%
9747   \if@eled@sectioning\else%
9748     \ifl@dprintingpages\else%
9749       \clearforchapter%
9750     \fi%
9751   \fi%
9752 }%
9753 {}%
9754 {}%
9755
9756 \patchcmd{\chapter}{\if@openright\cleardoublepage\else\clearpage\fi}{%
9757   \if@eled@sectioning\else%
9758     \ifl@dprintingpages%
9759       \endgraf%
9760     \else%
9761       \if@openright\cleardoublepage\else\clearpage\fi%No clearpage inside a
9762       \Pages: will keep critical notes from printing on the title page. Here for
9763       classical classes
9764     \fi%
9765   \fi%
9766 }%
9767 {}%
9768 {}%
9769 }%
9770 %

```

**\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.

```

9769 \newif\if@eled@sectioning%
9770 %

```

We reopen a new \notbool{@noeled@sec} statement, as we will define the \elesection commands.

```
9771 \notbool{@noeled@sec}{%
9772 %
```

## 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.

```
9773 \newwrite\eled@sectioning@out
9774 %
```

`\eledchapter` And now, the user sectioning commands, which write to the file, and also add content as a “normal” line.

```

\eledsection
\eledsubsection
\eledsubsubsection
\eledchapter*
\eledsection*
\eledsubsection*
\eledsubsubsection*
9775 \newcommand{\eledchapter}[2][]{%
9776   \disable@familiarnotes%
9777   #2%
9778   \restore@familiarnotes%
9779   \ifledRcol%
9780     \immediate\write\eled@sectioningR@out{%
9781       \string\eled@chapter{#1}{\unexpanded{#2}}{\the\l@dumpstartsR}{R}
9782     }%
9783   \else%
9784     \immediate\write\eled@sectioning@out{%
9785       \string\eled@chapter{#1}{\unexpanded{#2}}{\the\pstarts@read@L}{R}
9786     }%
9787   \fi%
9788 }
9789
9790 \newcommand{\eledsection}[2][]{%
9791   \disable@familiarnotes%
9792   #2%
9793   \restore@familiarnotes%
9794   \ifledRcol%
9795     \immediate\write\eled@sectioningR@out{%
9796       \string\eled@section{#1}{\unexpanded{#2}}{\the\l@dumpstartsR}{R}
9797     }%
9798   \else%
9799     \immediate\write\eled@sectioning@out{%
9800       \string\eled@section{#1}{\unexpanded{#2}}{\the\pstarts@read@L}{R}
9801     }%
9802   \fi%
9803 }
9804
9805 \newcommand{\eledsubsection}[2][]{%
9806   \disable@familiarnotes%
9807   #2%
```

```

9808 \restore@familiarnotes%
9809 \ifledRcol%
9810 \immediate\write\eled@sectioningR@out{%
9811 \string\eled@subsection{#1}{\unexpanded{#2}}{\the\l@dumpstartsR}{R
9812 }%
9813 \else%
9814 \immediate\write\eled@sectioning@out{%
9815 \string\eled@subsection{#1}{\unexpanded{#2}}{\the\pstarts@read@L}{R
9816 }%
9817 \fi%
9818 }
9819 \newcommand{\eledsubsubsection}[2][]{%
9820 \disable@familiarnotes%
9821 #2%
9822 \restore@familiarnotes%
9823 \ifledRcol%
9824 \immediate\write\eled@sectioningR@out{%
9825 \string\eled@subsubsection{#1}{\unexpanded{#2}}{\the\l@dumpstartsR
9826 }{R}
9827 }%
9828 \else%
9829 \immediate\write\eled@sectioning@out{%
9830 \string\eled@subsubsection{#1}{\unexpanded{#2}}{\the\pstarts@read@L
9831 }{R}%
9832 }%
9833 \fi%
9834 }
9835 \WithSuffix\newcommand\eledchapter*[2][]{%
9836 \disable@familiarnotes%
9837 #2%
9838 \restore@familiarnotes%
9839 \ifledRcol%
9840 \immediate\write\eled@sectioningR@out{%
9841 \string\eled@chapter{#1}{\unexpanded{#2}}{\the\l@dumpstartsR}{*}{R}
9842 }%
9843 \else%
9844 \immediate\write\eled@sectioning@out{%
9845 \string\eled@chapter{#1}{\unexpanded{#2}}{\the\pstarts@read@L}{*}{R}%
9846 }%
9847 \fi%
9848 }
9849 \WithSuffix\newcommand\eledsection*[2][]{%
9850 \disable@familiarnotes%
9851 #2%
9852 \restore@familiarnotes%

```

```

9854 \ifledRcol%
9855 \immediate\write\eled@sectioningR@out{%
9856 \string\eled@section{#1}{\unexpanded{#2}}{\the\l@dumpstartsR}{*}{R}
9857 }%
9858 \else%
9859 \immediate\write\eled@sectioning@out{%
9860 \string\eled@section{#1}{\unexpanded{#2}}{\the\pstarts@read@L}{*}{}%
9861 }%
9862 \fi%
9863 }
9864
9865 \WithSuffix\newcommand\eledsubsection*[2] [] {%
9866 \disable@familiarnotes%
9867 #2%
9868 \restore@familiarnotes%
9869 \ifledRcol%
9870 \immediate\write\eled@sectioningR@out{%
9871 \string\eled@subsection{#1}{\unexpanded{#2}}{\the\l@dumpstartsR}{*}{
R}
9872 }%
9873 \else%
9874 \immediate\write\eled@sectioning@out{%
9875 \string\eled@subsection{#1}{\unexpanded{#2}}{\the\pstarts@read@L
}{*}{}%
9876 }%
9877 \fi%
9878 }
9879
9880 \WithSuffix\newcommand\eledsubsubsection*[2] [] {%
9881 \disable@familiarnotes%
9882 #2%
9883 \restore@familiarnotes%
9884 \ifledRcol%
9885 \immediate\write\eled@sectioningR@out{%
9886 \string\eled@subsubsection{#1}{\unexpanded{#2}}{\the\l@dumpstartsR
}{*}{R}
9887 }%
9888 \else%
9889 \immediate\write\eled@sectioning@out{%
9890 \string\eled@subsubsection{#1}{\unexpanded{#2}}{\the\pstarts@read@L
}{*}{}%
9891 }%
9892 \fi%
9893 }
9894 %

```

### XXXIV.7 Macros written in the auxiliary file

`\eled@chapter` The sectioning macros, called in the auxiliary file. They have five arguments:  
`\eled@section`  
`\eled@subsection`  
`\eled@subsubsection`

1. Optional arguments of  $\LaTeX$  sectioning command.
2. Mandatory arguments of  $\LaTeX$  sectioning command.
3. Pstart number.
4. Side: R if right, nothing if left.
5. Starred or not.

```

9895 \def\eled@chapter#1#2#3#4#5{%
9896     \ifstrempy{#4}%
9897     {%
9898         \ifstrempy{#1}%
9899         {%
9900             \csgdef{eled@sectioning@#3#5}{\let\edtext=\dummy@edtext@showlemma\
chapter{#2}}%
9901             \csgdef{eled@sectmark@#3#5}{\let\edtext=\dummy@edtext{}\chaptermark
{#2}}}%
9902             }%Need for \pairs, because of using parbox.
9903             {%
9904                 \csgdef{eled@sectioning@#3#5}{\let\edtext=\dummy@edtext@showlemma\
chapter[#1]{#2}}%
9905                 \csgdef{eled@sectmark@#3#5}{\let\edtext=\dummy@edtext{}\chaptermark
{#2}}}%Need for \pairs, because of using parbox.
9906             }%
9907         }%
9908     }%
9909     \ifstrempy{#1}%
9910     {\csgdef{eled@sectioning@#3#5}{\let\edtext=\dummy@edtext@showlemma\
chapter*{#2}}}%
9911     {\csgdef{eled@sectioning@#3#5}{\let\edtext=\dummy@edtext@showlemma\
chapter*[#1]{#2}}}%Bug in LaTeX!
9912     }%
9913     \listcsgadd{eled@sections#5@@}{#3}%
9914     }
9915 \def\eled@section#1#2#3#4#5{%
9916     \ifstrempy{#4}%
9917     {\ifstrempy{#1}%
9918     {%
9919         \csgdef{eled@sectioning@#3#5}{\section{#2}}%
9920         \csgdef{eled@sectmark@#3#5}{\let\edtext=\dummy@edtext{}\sectionmark
{#2}}}%Need for \pairs, because of using parbox.
9921         }%
9922         {%
9923             \csgdef{eled@sectioning@#3#5}{\section[#1]{#2}}%
9924             \csgdef{eled@sectmark@#3#5}{\let\edtext=\dummy@edtext{}\sectionmark
{#1}}}%Need for \pairs, because of using parbox.
9925             }%
9926         }%

```



```

9927     {\ifstrempy{#1}%
9928       {\csgdef{eled@sectioning@#3#5}{\section*{#2}}}%
9929       {\csgdef{eled@sectioning@#3#5}{\section*{#1}{#2}}}%Bug in LaTeX!
9930     }
9931     \listcsgadd{eled@sections#5@@}{#3}%
9932   }
9933   \def\eled@subsection#1#2#3#4#5{%
9934     \ifstrempy{#4}%
9935       {\ifstrempy{#1}%
9936         {%
9937           \csgdef{eled@sectioning@#3#5}{\subsection{#2}}%
9938           \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)
9939         }%
9940         {%
9941           \csgdef{eled@sectioning@#3#5}{\subsection[#1]{#2}}%
9942           \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)
9943         }%
9944       }%
9945       {\ifstrempy{#1}%
9946         {\csgdef{eled@sectioning@#3#5}{\subsection*{#2}}}%
9947         {\csgdef{eled@sectioning@#3#5}{\subsection*{#1}{#2}}}%Bug in LaTeX!
9948       }
9949       \listcsgadd{eled@sections#5@@}{#3}%
9950     }
9951   \def\eled@subsubsection#1#2#3#4#5{%
9952     \ifstrempy{#4}%
9953       {\ifstrempy{#1}%
9954         {\csgdef{eled@sectioning@#3#5}{\subsubsection{#2}}}%
9955         {\csgdef{eled@sectioning@#3#5}{\subsubsection[#1]{#2}}}%
9956       }%
9957       {\ifstrempy{#1}%
9958         {\csgdef{eled@sectioning@#3#5}{\subsubsection*{#2}}}%
9959         {\csgdef{eled@sectioning@#3#5}{\subsubsection*{#1}{#2}}}%Bug in
LaTeX!
9960       }
9961       \listcsgadd{eled@sections#5@@}{#3}%
9962     }
9963   %
9964   %

```

End of the conditional test about noeledsec option.

```

9965   }{}
9966   %

```

## 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.

```
9967 \def\normal@page@break{}
9968 %
```

**\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.

```
9969 \def\l@prev@pb{}
9970 \def\l@prev@nopb{}
9971 %
```

**\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.

```
9972 \newcommand{\ledpb}{\write\linenum@out{\string\led@pb}}
9973 \newcommand{\ledpbnum}[1]{\write\linenum@out{\string\led@pbnum{#1}}}
9974 \newcommand{\lednopb}{\write\linenum@out{\string\led@nopb}}
9975 \newcommand{\lednopbnum}[1]{\write\linenum@out{\string\led@nopbnum{#1}}}
9976 %
```

**\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.

```
9977 \newcommand{\led@pb}{\listxadd{\l@prev@pb}{\the\absline@num}}
9978 \newcommand{\led@pbnum}[1]{\listxadd{\l@prev@pb}{#1}}
9979 \newcommand{\led@nopb}{\listxadd{\l@prev@nopb}{\the\absline@num}}
9980 \newcommand{\led@nopbnum}[1]{\listxadd{\l@prev@nopb}{#1}}
9981 %
```

**\ledpbsetting** The \ledpbsetting macro only changes the value of \led@pb@macro, for which the default value is before.

```
9982 \def\led@pb@setting{before}
9983 \newcommand{\ledpbsetting}[1]{\gdef\led@pb@setting{#1}}
9984 %
```

`\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`.

```

9985 \newcommand{\led@check@pb}{\xifinlist{\the\absline@num}{\l@prev@pb}{\
pagebreak[4]}}{}
9986 \newcommand{\led@check@nopb}{%
9987   \IfStrEq{\led@pb@setting}{before}{%
9988     \xifinlist{\the\absline@num}{\l@prev@nopb}%
9989     {\numdef{\abs@prevline}{\the\absline@num-1}%
9990     \xifinlist{\abs@prevline}{\normal@page@break}%
9991     {\nopagebreak[4]\enlargethispage{\baselineskip}}%
9992     {}}%
9993   {}}%
9994   {}%
9995   {}%
9996 \IfStrEq{\led@pb@setting}{after}{%
9997   \xifinlist{\the\absline@num}{\l@prev@nopb}{%
9998     \xifinlist{\the\absline@num}{\normal@page@break}%
9999     {\nopagebreak[4]\enlargethispage{\baselineskip}}%
10000     {}}%
10001 }%
10002   {}%
10003   {}%
10004   {}%
10005 }
10006 %

```

## 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.

```

10007 \newcommand{\check@pb@in@verse}{%
10008   \ifinstanza\iflednopbinverse\ifinserthangingsymbol% Using stanzas and
enabling page breaks in verse control, while on a hanging verse.
10009   \ifnum\page@num=\last@page@num\else%If we have change page
10010   \IfStrEq{\led@pb@setting}{before}{%

```

```

10011         \numgdef{\abs@line@verse}{\the\absline@num-1}%
10012         \ledpbnum{\abs@line@verse}%
10013     }{}%
10014     \IfStrEq{\led@pb@setting}{after}{%
10015         \numgdef{\abs@line@verse}{\the\absline@num-1}%
10016         \lednopbnum{\abs@line@verse}%
10017     }{}%
10018     \fi%
10019 \fi\fi\fi%
10020 }
10021 %

```

## 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>38</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>39</sup>

```

10022 \def\Hy@raisedlink@left#1{%
10023     \ifvmode
10024         #1%
10025     \else
10026         \Hy@SaveSpaceFactor
10027         \llap{\smash{%
10028             \begingroup
10029                 \let\HyperRaiseLinkLength\@tempdima
10030                 \setlength\HyperRaiseLinkLength\HyperRaiseLinkDefault
10031                 \HyperRaiseLinkHook
10032             \expandafter\endgroup
10033             \expandafter\raise\the\HyperRaiseLinkLength\hbox{%
10034                 \Hy@RestoreSpaceFactor
10035                 #1%
10036                 \Hy@SaveSpaceFactor
10037             }}%
10038         \Hy@RestoreSpaceFactor
10039         \penalty\@M\hskip\z@\relax
10040     \fi
10041 }
10042 %
10043

```

<sup>38</sup><http://tex.stackexchange.com/a/17138/7712>.

<sup>39</sup>The code is inspired by an answer given by @unbonpetit. Thanks to him. <http://texnique.fr:80/osqa/questions/781/hyraisedlink-perturbe-la-maniere-dont-se-fait-la-coupure-de-ligne/> 801.

## XXXVIII Compatibility with eledmac

Here, we define some commands for the eledmac-compat option.

```

10044 \ifeledmaccompat@%
10045
10046 \newcommand{\footnormalX}[1]{\arrangementX[#1]{normal}}%
10047 \newcommand{\footparagraphX}[1]{\arrangementX[#1]{paragraph}}%
10048 \newcommand{\foottwocolX}[1]{\arrangementX[#1]{twocol}}%
10049 \newcommand{\footthreecolX}[1]{\XarrangementX[#1]{threecol}}%
10050
10051 \unless\ifnocritical@
10052   \newcommand{\footnormal}[1]{\Xarrangement[#1]{normal}}%
10053   \newcommand{\footparagraph}[1]{\Xarrangement[#1]{paragraph}}%
10054   \newcommand{\foottwocol}[1]{\Xarrangement[#1]{twocol}}%
10055   \newcommand{\footthreecol}[1]{\Xarrangement[#1]{threecol}}%
10056   \let\hsizetwocol\Xhsizetwocol
10057   \let\hsizethreecol\Xhsizethreecol
10058   \let\bhookXnote\Xbhooknote
10059   \let\boxsymlinenum\Xboxsymlinenum
10060   \let\symlinenum\Xsymlinenum
10061   \let\beforenumberinfootnote\Xbeforenumber
10062   \let\afternumberinfootnote\Xafternumber
10063   \let\beforeXsymlinenum\XbeforeXsymlinenum
10064   \let\afterXsymlinenum\XafterXsymlinenum
10065   \let\inplaceofnumber\Xinplaceofnumber
10066   \let\Xlemmaseparator\lemmaseparator
10067   \let\afterlemmaseparator\Xafterlemmaseparator
10068   \let\beforelemmaseparator\Xbeforelemmaseparator
10069   \let\inplaceoflemmaseparator\Xinplaceoflemmaseparator
10070   \let\txbeforeXnotes\Xtxbeforenotes
10071   \let\afterXrule\Xafterrule
10072   \let\numberonlyfirstinline\Xnumberonlyfirstinline
10073   \let\numberonlyfirstintwoline\Xnumberonlyfirstintwoline
10074   \let\nonumberinfootnote\Xnonumberinfootnote
10075   \let\pstartinfootnote\Xpstart
10076   \let\pstartinfootnoteeverytime\Xpstarteverytime
10077   \let\onlyXpstart\Xonlypstart
10078   \let\Xnonumberinfootnote\Xnonumber
10079   \let\nonbreakableafternumber\Xnonbreakableafternumber
10080   \let\maxhXnotes\Xmaxhnotes
10081   \let\beforeXnotes\Xbeforenotes
10082   \let\boxlinenum\Xboxlinenum
10083   \let\boxlinenumalign\Xboxlinenumalign
10084   \let\boxstartlinenum\Xboxstartlinenum
10085   \let\boxendlinenum\Xboxendlinenum
10086   \let\twoline\Xtwoline
10087   \let\morethantwoline\Xmorethantwoline
10088   \let\twolinebutnotmore\Xtwolinebutnotmore
10089   \let\twolineonlyinsamepage\Xtwolineonlyinsamepage

```

```

10090 \fi
10091
10092 \unless\ifnofamiliar@
10093   \let\notesXwidthliketwocolumns\noteswidthliketwocolumnsX
10094 \fi
10095 \newcommandx{\parafootsep}[2][1,usedefault]{%
10096   \Xparafootsep[#1]{#2}%
10097   \parafootsepX[#1]{#2}
10098 }%
10099
10100 \newcommandx{\afternote}[2][1,usedefault]{%
10101   \Xafternote[#1]{#2}%
10102   \afternoteX[#1]{#2}%
10103 }%
10104
10105 \unless\ifnoend@
10106   \let\XendXtwolines\Xendtwolines
10107   \let\XendXmorethantwolines\Xendmorethantwolines
10108   \let\XhookXendnote\Xendhooknote
10109   \let\XboxXendlinenum\Xendboxlinenum%
10110   \let\XboxXendlinenumalign\Xendboxlinenumalign%
10111   \let\XboxXendstartlinenum\Xendboxstartlinenum%
10112   \let\XboxXendendlinenum\Xendboxendlinenum%
10113   \let\XendXlemmaseparator\Xendlemmaseparator
10114   \let\XendXbeforelemmaseparator\Xendbeforelemmaseparator
10115   \let\XendXafterlemmaseparator\Xendafterlemmaseparator
10116   \let\XendXinplaceoflemmaseparator\Xendinplaceoflemmaseparator
10117 \fi
10118
10119 \AtBeginDocument{%
10120   \ifdef\lineref{}\let\lineref\edlineref}%
10121 }%
10122
10123 \fi%
10124 %
10125 %

```

&lt;/code&gt;

## 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>40</sup> to `\critext`; use the `\edtext` macro instead. However, if you do use `\critext` (the new name for `\text`), the following is a reminder.

`\critext` Within numbered paragraphs, footnotes and endnotes are generated by forms of the `\critext` macro:

```
\critext{⟨lemma⟩}{⟨commands⟩/}
```

The `⟨lemma⟩` argument is the lemma in the main text: `\critext` both prints this as part of the text, and makes it available to the `⟨commands⟩` you specify to generate notes. The `/` at the end terminates the command; it is part of the macro's definition so that spaces after the macro will be treated as significant.

For example:

<pre>I saw my friend \critext{Smith} \Afootnote{Jones C, D.}/ on Tuesday.</pre>	<pre>1 I saw my friend 2 Smith on Tuesday. 2 Smith] Jones C, D.</pre>
---	---

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

The `⟨lemma⟩` may contain further `\critext` commands. Nesting makes it possible to print an explanatory note on a long passage together with notes on variants for individual words within the passage. For example:

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

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

The second argument of the `\critext` macro, `⟨commands⟩`, is the same as the second argument to the `\edtext` macro.

It is possible to define aliases for `\critext`, which can be easier to type. You can make a single character substitute for `\critext` by saying this:

```
\catcode'\<=\active
```

<sup>40</sup>A name like `\text` is likely to be defined by other  $\TeX$  packages (it certainly is by the AMS packages) and it seems sensible to try and avoid clashes with other definitions.

```
\let<=\critext
```

Then you might say `<{Smith}\variant{Jones}/`. This of course destroys the ability to use `<` in any new macro definitions, so long as it remains in effect; hence it should be used with care.

Changing the character at the end of the command requires more work:

```
\catcode'\<=\active
\def\xtext#1#2>{\critext{#1}{#2}/}
\let<=\xtext
```

This allows you to say `<{Smith}\Afootnote{Jones}>`.

Aliases for `\critext` of the first kind shown here also can't be nested—that is, you can't use the alias in the text that forms the first argument to `\critext`. (See VI p. 136 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. 42) 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>41</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>41</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 `stanzaindentsrepetition` (cf. 9.3 p. 57). This bug had two consequences:

1. `stanzaindentsrepetition` did not work when its value was greater than 2.
2. `stanzaindentsrepetition` worked wrong when its value was equal to 2.

So, if you used `stanzaindentsrepetition` with a value equal to 2, you had to change your `\setstanzaindents`. Explanation:

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

This code, in versions prior to 1.5.1, made the first line have an indentation of 0, the second line of 1, the third verse of 0, the fourth verse of 1 and so forth.

But this code should have instead achieved quite the contrary: the first line would have an indentation of 1, the second line of 0, the third line of 1, the fourth line of 0 and so forth.

So version 1.5.1 corrected this bug. If you want to keep the former presentation, you must change:

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

to:

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

### 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. 75.

## 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 `\printnnum`. This spurious space has been deleted. However, if you want to keep the previous spurious space, you may load the package with the `oldprintnnumspace` 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. 77 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 `\Xlinerrangeseparator` or `/` and `\Xendlinerrangeseparator` (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. 45.

`\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.

## 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 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>\@hangingsymbol</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	77
CLASSbook	77, 376
CLASSmemoir	222, 223, 287–290, 325, 376, 395, 458, 462
CLASSscrbook	462
COMMAND\*footnote	79
COMMAND\...\@footnotemark...	225
COMMAND\...d@ta	165
COMMAND\<hook	
@<series	274
COMMAND\<hookname	
<pseudoserries	276
COMMAND\<type	
footfmt	209
COMMAND\@@line	200
COMMAND\@MM	186, 459
COMMAND\@Rlineflag	327, 459
COMMAND\@Serefprefix	304
COMMAND\@Serefprefixmore	304
COMMAND\@add@	366
COMMAND\@adv	119
COMMAND\@annot	171
COMMAND\@annot@end@print	213
COMMAND\@annot@start@print	213, 217, 255
COMMAND\@apprefprefixmore	304
COMMAND\@apprefprefixsingle	304
COMMAND\@beforeinsertofthisedtext	142, 143
COMMAND\@bsphack	292
COMMAND\@docclearpage	288, 289, 452, 462
COMMAND\@doreinfeetX	462
COMMAND\@dprintingcolumns	459
COMMAND\@edindex@hyperref	328, 329
COMMAND\@edtext@level	139
COMMAND\@esphack	292
COMMAND\@firstofone	37
COMMAND\@fnpos	244, 284
COMMAND\@footnotemark	223, 452, 462
COMMAND\@footnotetext	223, 224, 452
COMMAND\@gobble	27, 37, 137, 138, 171, 268
COMMAND\@gobblefive	461
COMMAND\@gobblefour	458

COMMAND\@gobbleseven	270
COMMAND\@gobblethree	451
COMMAND\@h	203
COMMAND\@hangingsymbol	333
COMMAND\@iiiminipage	316, 319, 451, 462
COMMAND\@iiiminpage	317
COMMAND\@l	456
COMMAND\@l@tempcnta	168, 172, 180
COMMAND\@l@tempcntb	172
COMMAND\@l@reg	456
COMMAND\@lab	115, 292, 295, 300, 451
COMMAND\@ldunboxmpfoot	319
COMMAND\@led@extranofeet	396
COMMAND\@ledinnote@command	323, 324
COMMAND\@lemma	142, 145
COMMAND\@lock	109, 333
COMMAND\@lopL	452
COMMAND\@lopR	452
COMMAND\@makecol	284, 285, 287, 288, 462
COMMAND\@mempnum	290
COMMAND\@mpfnpos	244
COMMAND\@msd	342, 343
COMMAND\@msd@c	342
COMMAND\@msd@options@iffullpage	349
COMMAND\@msdata@list	342, 344
COMMAND\@nl	115–120, 130, 171, 295, 451, 452
COMMAND\@nl@reg	116, 393, 452, 456
COMMAND\@opXfeet	452
COMMAND\@opfeetX	462
COMMAND\@opxtrafeeti	462
COMMAND\@outputpage	290, 291
COMMAND\@page	115, 117, 295
COMMAND\@pend	452
COMMAND\@pendR	452
COMMAND\@ref	115, 125–127, 131, 132, 137
COMMAND\@ref@later	127, 133
COMMAND\@ref@reg	126, 171, 452
COMMAND\@ref@reg@parsearg	127
COMMAND\@reinserts	284, 286, 287, 462
COMMAND\@secondoftwo	79
COMMAND\@sect	378
COMMAND\@series	272
COMMAND\@set	120
COMMAND\@sidenotesep	315
COMMAND\@stopmsd	343
COMMAND\@sw	127, 147, 150, 151
COMMAND\@tag	139, 140, 144
COMMAND\@tempcnta	93
COMMAND\@tempcntb	93
COMMAND\@this@crossref@start	221

COMMAND\@toksa	100
COMMAND\@toksb	100
COMMAND\@xloop	181
COMMAND\@xympar	309, 462
COMMAND\Aendnote	17, 30
COMMAND\Afootfmt	186
COMMAND\Afootgroup	186
COMMAND\Afootnote	9, 17, 29, 30, 33, 141, 194, 222, 245, 263, 461
COMMAND\Afootstart	186
COMMAND\AtBeginDocument	288
COMMAND\AtEndEveryPend	19, 468
COMMAND\AtEveryPend	19, 60, 158, 458, 460, 462, 468
COMMAND\AtEveryPend*	19
COMMAND\AtEveryPstart	19, 60, 458, 460, 462, 465, 468
COMMAND\AtEveryPstart*	19
COMMAND\AtEveryStanza	59, 466, 468
COMMAND\AtEveryStopStanza	60, 466, 468
COMMAND\AtStartEveryPstart	19, 468
COMMAND\AtStartEveryStanza	59, 468
COMMAND\BeforeEveryStopStanza	60, 468
COMMAND\Bendnote	17, 29
COMMAND\Bfootnote	9, 17, 222, 245, 263
COMMAND\Centering	53
COMMAND\Cfootnote	222
COMMAND\Columns	94, 154, 191
COMMAND\Dfootnote	222
COMMAND\Efootnote	222
COMMAND\Gls	71
COMMAND\Hy@raisedlink	388
COMMAND\Hy@raisedlink@left	388
COMMAND\LTR	52
COMMAND\NR	378
COMMAND\Pages	94, 154, 285, 286, 290
COMMAND\ProcessOptionsX	84
COMMAND\RL	51
COMMAND\RaggedLeft	53
COMMAND\RaggedRight	52
COMMAND\RenewExpandableDocumentCommand	38, 149
COMMAND\SEonlypage	301, 464
COMMAND\SEref	64–66, 301, 305, 465, 467, 471
COMMAND\SErefonlypage	64–66, 399, 464
COMMAND\SErefwithpage	64, 66, 301, 305, 399, 464, 466
COMMAND\Stanza	457
COMMAND\Waklam	366
COMMAND\X@doreinfeet	286, 462
COMMAND\XR@prefix	308
COMMAND\XR@test	308
COMMAND\XR@test@mac	308
COMMAND\XR@test@mac@test	308
COMMAND\XXXXXXfmt	392

COMMAND\XXXXXfmt	392
COMMAND\Xafterenumber	48
COMMAND\Xafterlemmaseparator	49, 396
COMMAND\Xafternote	53, 54, 395, 396
COMMAND\Xafternumber	47, 48, 396
COMMAND\Xafterrule	55, 245, 396, 457, 460
COMMAND\Xaftersymlinenum	48, 396
COMMAND\Xarrangement	43, 53, 79, 187, 274, 396
COMMAND\Xarrangement@footparagraph	193
COMMAND\Xarrangement@normal	187
COMMAND\Xarrangement@paragraph	192
COMMAND\Xbeforeinserting	52
COMMAND\Xbeforelemmaseparator	49, 396
COMMAND\Xbeforenotes	54, 244, 396, 457, 460
COMMAND\Xbeforenumber	45, 47, 48, 396
COMMAND\Xbeforesymlinenum	48, 396
COMMAND\Xbhookendnote	397
COMMAND\Xbhookgroup	54, 464, 465
COMMAND\Xbhooknote	52, 397, 398, 462, 463
COMMAND\Xboxendlinum	49, 397, 461
COMMAND\Xboxlinenum	48, 49, 397
COMMAND\Xboxlinenumalign	48, 49, 397, 461
COMMAND\Xboxstartlinenum	49, 397, 461
COMMAND\Xboxsymlinenum	48, 397
COMMAND\Xcolalign	52, 460
COMMAND\Xdo@feet	462, 467
COMMAND\Xemdnoidenticallinenumannotation	66
COMMAND\Xend	270
COMMAND\XendXafterlemmaseparator	397
COMMAND\XendXbeforelemmaseparator	397
COMMAND\XendXinplaceoflemmaseparator	397
COMMAND\XendXlemmaseparator	397
COMMAND\XendXmorethantwolines	397
COMMAND\XendXtwolines	397
COMMAND\Xendafterenumber	47, 463
COMMAND\Xendafterlemmaseparator	50, 397
COMMAND\Xendafternote	56, 465
COMMAND\Xendafternumber	49
COMMAND\Xendafterpagenumbe	466
COMMAND\Xendafterpagenumber	45, 66
COMMAND\XendafterpagenumberSErefonlypage	399
COMMAND\Xendaftersymlinenum	48, 49, 463
COMMAND\Xendahookinplaceofnumber	49, 463
COMMAND\Xendahooklinenum	49, 463
COMMAND\Xendbeforelemmaseparator	50, 397
COMMAND\Xendbeforelinenum	49
COMMAND\Xendbeforenumber	47, 463
COMMAND\Xendbeforepagenumber	45, 65, 66
COMMAND\XendbeforepagenumberSErefonlypage	65
COMMAND\Xendbeforesymlinenum	48, 49, 463

COMMAND\Xendbhookinplaceofnumber	49, 463
COMMAND\Xendbhooklinenumber	49, 463
COMMAND\Xendbhooknote	52
COMMAND\Xendboxendlinenum	49, 461
COMMAND\Xendboxlinenum	49, 397, 459
COMMAND\Xendboxlinenumalign	49, 397, 461
COMMAND\Xendboxstartlinenum	49, 461
COMMAND\Xendboxsymlinenum	48, 463
COMMAND\Xendhangindent	52, 463, 465
COMMAND\Xendinplaceoflemmaseparator	30, 50, 397
COMMAND\Xendinplaceofnumber	48, 462
COMMAND\Xendinplaceofpagenumber	45, 467
COMMAND\Xendinsertsep@	251
COMMAND\Xendlemmadisablefontselection	51
COMMAND\Xendlemmafont	51, 464
COMMAND\Xendlemmaseparator	31, 50, 397
COMMAND\Xendletboxendlinenum	397
COMMAND\Xendlineflag	66
COMMAND\Xendlinenumannotationonlyfirst	44, 255, 471
COMMAND\Xendlinenumannotationonlyfirstintwo	44, 255, 471
COMMAND\Xendlinenumannotationposition	26, 66
COMMAND\Xendlineprefixmore	45, 66
COMMAND\Xendlineprefixsingle	45, 66
COMMAND\Xendlinerangeseparator	46, 66, 208, 399, 463
COMMAND\Xendmorethantwolines	30, 46, 66, 397, 460, 461
COMMAND\Xendnoidenticallinenumannotation	27, 255, 470
COMMAND\Xendnonumber	46, 462
COMMAND\Xendnote	248, 269, 270, 460
COMMAND\Xendnotefontsize	51, 396
COMMAND\Xendnotenumfont	49, 50, 396
COMMAND\Xendnotes	251
COMMAND\Xendnumberonlyfirstinline	44, 463
COMMAND\Xendnumberonlyfirstintwolines	44, 463
COMMAND\Xendpagenumberonlyfirst	45, 467
COMMAND\Xendpagenumberonlyfirstifsingle	45, 467
COMMAND\Xendpagenumberonlyfirstintwo	45, 467
COMMAND\Xendparagraph	56, 457
COMMAND\Xendsep	56
COMMAND\Xendsublinesep	47, 66, 208
COMMAND\Xendsymlinenum	44, 463
COMMAND\Xendsymlinenumannotation	44, 471
COMMAND\Xendsympagenum	45, 467
COMMAND\Xendtwolines	30, 46, 66, 397, 460, 461
COMMAND\Xendtwolinesbutnotmore	46, 66, 460, 461
COMMAND\Xendtwolinesonlyinsamepage	46, 66, 460, 461
COMMAND\Xendtxtbeforenotes	54, 469
COMMAND\Xendwrapcontent	51, 466
COMMAND\Xendwraplemma	51, 466
COMMAND\Xendwraplinenumannotation	27, 66
COMMAND\Xfootfmt	398



COMMAND\Xfootgroup	192
COMMAND\Xfootins	191
COMMAND\Xfootnote	62, 69, 139, 392, 454, 458–460, 464, 466
COMMAND\Xfootstarts	192
COMMAND\Xgroupbyline	54, 179, 220, 282, 401, 469–471
COMMAND\Xgroupbylines	467
COMMAND\Xgroupbylineseparetwolines	54
COMMAND\Xhangindent	51, 463, 471
COMMAND\Xhsize	399, 464, 465
COMMAND\Xhsizethreecol	53, 55, 397
COMMAND\Xhsizetwocol	53, 55, 276, 397
COMMAND\Xinplaceoflemmaseparator	30, 50, 397
COMMAND\Xinplaceofnumber	48, 397, 459, 461
COMMAND\Xinplaceofnnumber	30
COMMAND\Xinsertparafootsep	197, 199
COMMAND\Xledsetnormalparstuff	394, 395, 460
COMMAND\Xlemmadisablefontselection	50
COMMAND\Xlemmafont	51, 464
COMMAND\Xlemmaseparator	49, 50, 208, 277, 280, 284, 397
COMMAND\Xlineflag	65, 399, 400, 466
COMMAND\Xlinenumannotationonlyfirst	44, 217, 471
COMMAND\Xlinenumannotationonlyfirsttwo	44, 217, 471
COMMAND\Xlinenumannotationposition	26, 65, 170
COMMAND\Xlinenumannotationposition@side	170
COMMAND\Xlinenumannotationpositionside	26
COMMAND\Xlinrangeseparator	46, 65, 208, 399, 463
COMMAND\Xmaxhnotes	55, 77, 79, 396, 397, 457, 459
COMMAND\Xmorethantwolines	30, 46, 65, 397, 459
COMMAND\Xnoidenticallinenumannotation	27, 65, 216, 470
COMMAND\Xnoindent	463
COMMAND\Xnolemmaseparator	50, 284, 397
COMMAND\Xnonbreakableafternumber	48, 397, 455
COMMAND\Xnonumber	46, 397
COMMAND\Xnonumberinfootnote	397
COMMAND\Xnotefontsize	51, 396
COMMAND\Xnotefontsize@⟨s⟩	197, 202, 203
COMMAND\Xnotenumfont	50, 396
COMMAND\Xnoteswidthliketwocolumns	56, 458
COMMAND\Xnumberonlyfirstinline	44, 45, 54, 112, 209, 276, 277, 280, 397, 454, 459, 468
COMMAND\Xnumberonlyfirstintwolines	44, 54, 397, 454, 468
COMMAND\Xonlypstart	47, 397, 454, 459
COMMAND\Xpagelinesep	47, 467
COMMAND\Xparafootsep	53, 54, 112, 396, 397, 466, 468, 469
COMMAND\Xparafootsep@series	197
COMMAND\Xparindent	51, 399, 401, 460, 463, 465
COMMAND\Xprenotes	54, 245, 466
COMMAND\Xprenotes@	191, 245, 454
COMMAND\Xprintlines	111
COMMAND\Xpstart	46, 47, 394, 397, 454, 459
COMMAND\Xstarteverytime	47, 394, 397, 459

COMMAND\Xragged	53
COMMAND\Xrigidbalance	200, 399, 464
COMMAND\Xsethangindent	207
COMMAND\Xstanza	47, 60
COMMAND\Xstanzaseparator	47
COMMAND\Xstorelineinfo	209
COMMAND\Xsublinesep	23, 47, 65, 208, 399
COMMAND\Xsublinesepside	23, 47
COMMAND\Xsymlinenum	44, 53, 396, 397, 461, 468
COMMAND\Xsymlinenumannotation	44, 471
COMMAND\Xtextbeforenotes	182
COMMAND\Xtoendnotes	31, 270
COMMAND\Xtwolines	30, 46, 65, 216, 217, 276, 397, 459
COMMAND\Xtwolinesappref	276
COMMAND\Xtwolinesbutnotmore	46, 65, 397, 460
COMMAND\Xtwolinesbutnotmoreappref	276
COMMAND\Xtwolinesonlyinsamepage	46, 65, 397, 460
COMMAND\Xtxtbeforenotes	54, 397, 466, 467, 469
COMMAND\Xtxtbeforenotesonlyonce	54, 469
COMMAND\Xunvxh	195, 398
COMMAND\Xwidth	55, 399, 465
COMMAND\Xwrapcontent	51, 466
COMMAND\Xwraplemma	51, 52, 466
COMMAND\Xwraplinenumannotation	27, 65, 170
COMMAND\Xwraplinenumannotation@ref	170
COMMAND\Xwraplinenumannotation@side	170
COMMAND\Xwraplinenumannotationref	27
COMMAND\Xwraplinenumannotationside	27
COMMAND\&	396
COMMAND\absline@num	108, 109, 166
COMMAND\accent	138
COMMAND\actionlines@list	110, 168
COMMAND\actions@list	110
COMMAND\add@Xgroupbyline	179
COMMAND\add@hooktoggle@specific@to@cs	282
COMMAND\add@inserts	109, 178, 179
COMMAND\add@inserts@next	178, 179
COMMAND\add@msd@	342
COMMAND\add@msdata	342, 343
COMMAND\add@msdata@firstlineofpage	346
COMMAND\add@msddata	342
COMMAND\add@penalties	166, 180
COMMAND\addcontentsline	373
COMMAND\addfootins	392, 395
COMMAND\addfootinsX	392, 395
COMMAND\addtoendnotes	270
COMMAND\advancelabel@refs	294
COMMAND\advanceline	24, 111, 119, 134, 462
COMMAND\affixline@num	172, 175, 177, 315, 452
COMMAND\affixpstart@num	177

COMMAND\afterXrule	396
COMMAND\afterXsymlinenum	396
COMMAND\aftergroup	137, 141
COMMAND\afterlemmaseparator	396
COMMAND\afternote	396
COMMAND\afternoteX	53, 395, 396
COMMAND\afternumberinfootnote	396
COMMAND\afterruleX	55, 457, 460
COMMAND\annot@end@print	217, 255
COMMAND\annot@list	111, 172
COMMAND\annotationref	27, 62
COMMAND\applabel	64, 296, 297, 460, 466
COMMAND\appref	62, 64–66, 301, 305, 464, 465, 471
COMMAND\apprefprefixmore	65, 396
COMMAND\apprefprefixsingle	64, 396
COMMAND\apprefwithpage	64, 66, 301, 305, 461, 464
COMMAND\arrangementX	43, 79, 226, 274, 396
COMMAND\arrangementX@normal	231
COMMAND\article	15
COMMAND\at@every@pend	158
COMMAND\autopar	18, 19, 154, 159, 160, 242, 453, 455, 456, 460, 470
COMMAND\ballast	78
COMMAND\ballast@count	166, 180
COMMAND\baselineskip	43, 193, 197
COMMAND\beforeXnotes	396
COMMAND\beforeXsymlinenum	396
COMMAND\beforeeledchapter	11, 75, 374, 375
COMMAND\beforeinsertingX	52
COMMAND\beforelemmaseparator	396
COMMAND\beforenotesX	54, 398, 456, 457, 460
COMMAND\beforenumberinfootnote	396
COMMAND\begin	350, 351
COMMAND\beginnumbering	17, 18, 20, 94, 95, 98, 99, 108, 112, 113, 129, 159, 247, 341, 454, 457, 461, 462, 467, 468
COMMAND\beginnumberingR	94
COMMAND\bf	454
COMMAND\bfseries	50, 454
COMMAND\bhookXnote	397
COMMAND\bhookgroupX	54, 464
COMMAND\bhooknoteX	52, 398, 462, 463
COMMAND\body	333
COMMAND\bodyfootmarkA	40
COMMAND\book	15
COMMAND\boxXendlinenum	397
COMMAND\boxXendlinenumalign	397
COMMAND\boxXendstartlinenum	397
COMMAND\boxendlinenum	397
COMMAND\boxlinefootnote	211
COMMAND\boxlinenum	397
COMMAND\boxlinenumalign	397

COMMAND\boxstartlinenum	397
COMMAND\boxsymlinenum	397
COMMAND\break	43, 195
COMMAND\brokenpenalty	180
COMMAND\c@page	130
COMMAND\c@page@page	130
COMMAND\centering	53
COMMAND\ch@ck@l@ck	452
COMMAND\ch@cksub@l@ck	175, 452
COMMAND\chapter	74, 376, 380, 457, 460, 462, 466
COMMAND\chaptermark	374
COMMAND\check@pb@in@verse	387
COMMAND\colalignX	52, 460
COMMAND\collect@body	351
COMMAND\color	466
COMMAND\colorbox	79
COMMAND\columns	56
COMMAND\columnwidth	193, 458
COMMAND\command names	276
COMMAND\continuousnumberingwithcolumns@sync@linenumber@singletext	169
COMMAND\copyright	138
COMMAND\correct@Xfootins@box	459
COMMAND\correct@footinsX@box	459
COMMAND\count	201
COMMAND\critex	453
COMMAND\critext	145, 391, 392, 395
COMMAND\csname	84, 148, 149
COMMAND\csquotes	267
COMMAND\ctab	367, 372
COMMAND\ctabtext	372
COMMAND\current@annot	171
COMMAND\dcoll	361
COMMAND\def	81, 97
COMMAND\detokenize	149
COMMAND\dimen	201
COMMAND\dimexpr	55
COMMAND\discretionary	195
COMMAND\displaywidowpenalty	180
COMMAND\do@Xfeet	285, 452, 462, 467
COMMAND\do@actions	166–168, 452
COMMAND\do@actions@fixedcode	452
COMMAND\do@actions@next	167, 168
COMMAND\do@ballast	166, 180
COMMAND\do@feet@custom@order	284
COMMAND\do@insidelinehook	455
COMMAND\do@line	109, 136, 156, 160, 164, 165, 178, 180, 333, 452, 453, 455, 457
COMMAND\do@linehook	452
COMMAND\do@lockoff	110
COMMAND\do@lockon	110
COMMAND\dodoreinextrafeet	451

COMMAND\doendnotes	31, 251, 252, 398, 461, 468
COMMAND\doendnotesbysection	31, 252, 271, 461, 468
COMMAND\doennotes	468
COMMAND\doinsidelinehook	28, 457
COMMAND\doinsidethislinehook	28, 164, 470
COMMAND\dolinehook	28, 457
COMMAND\doreintrafeeti	462
COMMAND\doreintrafeetii	462
COMMAND\doxtrafeet	451
COMMAND\doxtrafeeti	462
COMMAND\doxtrafeetii	462
COMMAND\dummy@ref	137
COMMAND\edaftertab	73, 367, 368
COMMAND\edannotationref	299
COMMAND\edatleft	73, 365
COMMAND\edatright	73, 74, 365
COMMAND\edbeforetab	73, 367
COMMAND\edfilldimen	366
COMMAND\edfont@info	144
COMMAND\edgls	71, 322
COMMAND\edgls...	465
COMMAND\edglsadd	468
COMMAND\edindex	38, 69–71, 322, 323, 326, 328, 329, 355, 455, 457, 459, 462, 463, 467–469
COMMAND\edindexlab	70
COMMAND\edlabel	62–64, 67, 138, 291, 292, 294, 295, 298, 299, 307, 322, 355, 388, 400, 451, 454–456, 459, 464, 469
COMMAND\edlabelE	64, 297
COMMAND\edlabelS	64, 297
COMMAND\edlabelSE	64
COMMAND\edlineref	62, 292, 397, 459, 461, 464, 468
COMMAND\edmakelabel	64, 307, 308
COMMAND\edpageref	62, 291, 292, 298, 307
COMMAND\edrowfill	366
COMMAND\edsublineref	62
COMMAND\edtabcolsep	360
COMMAND\edtext	7, 25, 29, 32, 34, 35, 37, 56, 62–64, 68, 69, 71, 79, 97, 109, 111, 125, 127, 131, 132, 136–145, 147, 148, 150, 152, 170–172, 296, 297, 301, 355, 357, 376, 388, 391, 392, 395, 451, 453, 455, 457–461, 467–470
COMMAND\edtext@level	461
COMMAND\edtextlater	128
COMMAND\edvertdots	74, 365
COMMAND\edvertline	74, 365
COMMAND\elechapter	75
COMMAND\eled@sectioning@out	381
COMMAND\eledchapter	75, 395, 457, 462
COMMAND\eledchapter*	75
COMMAND\eledmac@error	451
COMMAND\eledsection	7, 17, 75, 137, 163, 374, 395, 459, 468, 470
COMMAND\eledsection*	75
COMMAND\eledsubsection	75, 395

COMMAND\eledsubsection*	75
COMMAND\eledsubsubsection	75, 395
COMMAND\eledsubsubsection*	75
COMMAND\eledxxx	11, 75, 380, 381, 457
COMMAND\eledxxxx	374
COMMAND\elesection	381
COMMAND\else	321, 374
COMMAND\emph	37
COMMAND\empty	93, 173, 292
COMMAND\end	350, 351
COMMAND\end@lemmas	137
COMMAND\endashchar	207
COMMAND\endgraf	156, 197, 242
COMMAND\endlock	23, 111, 135, 338
COMMAND\endminipage	316, 317, 319, 451, 462
COMMAND\endmsdata	39
COMMAND\endnotes	460, 464
COMMAND\endnumbering	17, 18, 20, 94, 95, 97, 98, 129, 452, 461, 467, 468
COMMAND\endnumberingR	94
COMMAND\endprint	247, 250, 270, 394
COMMAND\endstanzaextra	396
COMMAND\endsub	23, 110, 133
COMMAND\endverse	398
COMMAND\everypar	159
COMMAND\extensionchars	77, 95
COMMAND\externaldocument	67, 308
COMMAND\f@x@l@cks	452
COMMAND>falseverse	395, 455, 457
COMMAND\fi	374
COMMAND\firstlinenum	22, 172, 453
COMMAND\firstsublinenum	22, 453
COMMAND\flag@end	131, 132, 143, 457
COMMAND\flag@end@RTL	132
COMMAND\flag@end@later	133
COMMAND\flag@start	131, 132, 143, 457, 458
COMMAND\flag@start@RTL	132
COMMAND\flag@start@later	133
COMMAND\flagstanza	61
COMMAND\floatingpenalty	186, 459
COMMAND\flush@notes	181
COMMAND\fnpos	42, 244, 456, 467
COMMAND\footfmt	186, 188
COMMAND\footfmt...	226
COMMAND\footfootmarkA	40
COMMAND\footfudgefactor	195
COMMAND\footfudgefiddle	78, 193, 451
COMMAND\footgroup	186
COMMAND\footins	191
COMMAND\footnormal	275, 396, 451
COMMAND\footnormalX	396

COMMAND\footnote	40, 78, 222–224, 393, 452
COMMAND\footnote@lang	208
COMMAND\footnoteA	17, 40
COMMAND\footnoteB	17
COMMAND\footnoteC	29
COMMAND\footnoteE	40
COMMAND\footnoteX	9, 41, 266, 268, 467
COMMAND\footnoteX@reading	268
COMMAND\footnoteXmark	268, 468
COMMAND\footnoteXmk	283
COMMAND\footnoteXtext	268, 468
COMMAND\footnote⟨X⟩	138
COMMAND\footnote⟨X⟩mark	41
COMMAND\footnote⟨X⟩mk	41
COMMAND\footnote⟨X⟩nomk	41
COMMAND\footnote⟨X⟩text	41
COMMAND\footnotelang@lua	184
COMMAND\footnotelang@poly	185
COMMAND\footnotemark	41, 268
COMMAND\footnoteoption@	184, 463
COMMAND\footnoterule	201
COMMAND\footnotesize	51
COMMAND\footnotetext	41, 268
COMMAND\footparagraph	193, 275, 396, 457
COMMAND\footparagraphX	237, 396, 457
COMMAND\footplitskips	452, 459
COMMAND\footstart	186, 191, 201
COMMAND\footstrut	197
COMMAND\footthreecol	396
COMMAND\footthreecolX	396, 460
COMMAND\foottwocol	396
COMMAND\foottwocolX	396, 460
COMMAND\foreignlanguage	51
COMMAND\fullstop	399
COMMAND\get@edindex@hyperref	328
COMMAND\get@edindex@ledinnote@command	323
COMMAND\get@fnmark	224
COMMAND\get@index@command	456
COMMAND\get@linelistfile	452
COMMAND\get@thisfootnote	230
COMMAND\getline@num	166, 167
COMMAND\gl@p	101
COMMAND\global	115
COMMAND\globaldefs	115
COMMAND\gls	71, 330
COMMAND\hangindentX	52, 460, 463, 471
COMMAND\hangingsymbol	396, 453
COMMAND\hbox	195
COMMAND\hfill	455
COMMAND\hidenumbering	24, 124, 400, 460, 470

COMMAND\hidenumberingonleftpage	24, 124, 466
COMMAND\hidenumberingonrightpage	24, 124, 466
COMMAND\hline	71
COMMAND\hooktoggle@specific	282
COMMAND\hrulefill	366
COMMAND\hsize	43, 191, 193–195, 202, 205, 242, 452, 458
COMMAND\hsizeX	399, 464, 465
COMMAND\hsizethreecol	397
COMMAND\hsizethreecolX	53, 55
COMMAND\hsizetwocol	397
COMMAND\hsizetwocolX	53, 55
COMMAND\hyperlinkR	327
COMMAND\hyperlinkformat	327
COMMAND\hyperlinkformatR	327
COMMAND\if@RTL	85, 141
COMMAND\if@edtext@	458, 461
COMMAND\if@eled@sectioning	380
COMMAND\if@firstlineofpage	86
COMMAND\if@firstlineofpageR	86
COMMAND\if@msd@options@fullpage	349
COMMAND\if@msdata@insertedfrompreviouspage	346
COMMAND\if@nobreak	158
COMMAND\if@noneed@Footnote	131
COMMAND\ifXnote@	94
COMMAND\ifafterendnumberingR	94
COMMAND\ifautopar	159
COMMAND\ifby@autopar	159
COMMAND\ifbypage@	101
COMMAND\ifbypage@R	101
COMMAND\ifbypstart@	101
COMMAND\ifbypstart@R	101
COMMAND\iffirst@linenum@out@	128, 129
COMMAND\ifindtl@innote	94
COMMAND\ifindtl@notenumber	94
COMMAND\ifinserthangingsymbol	333
COMMAND\ifinstanza	333
COMMAND\ifistwofollowinglines	217, 218
COMMAND\ifl@d@Xmorethantwolines	213, 459
COMMAND\ifl@d@Xtwolines	213
COMMAND\ifl@d@dash	213
COMMAND\ifl@d@elin	212
COMMAND\ifl@d@esl	212
COMMAND\ifl@d@pnum	212
COMMAND\ifl@d@ssub	212
COMMAND\ifl@dend@X	269
COMMAND\ifl@dmemoir	451
COMMAND\ifl@dpaging	458
COMMAND\ifl@dpairing	93, 453
COMMAND\ifl@dprintingpages	459
COMMAND\ifl@dskipnumber	172



COMMAND\ifl@dstartendok	366
COMMAND\ifl@imakeidx	85
COMMAND\ifledRcol	94, 453
COMMAND\ifledRcol@	94, 457
COMMAND\iflemmacommand@	458
COMMAND\ifnoend@	252
COMMAND\ifnoledgroup@	321
COMMAND\ifnoteschanged@	112
COMMAND\ifnumberedpar@	154
COMMAND\ifnumbering	95, 98
COMMAND\ifnumberingR	94, 453
COMMAND\ifnumberline	143, 172
COMMAND\ifpst@rted	453
COMMAND\ifpst@rtedL	95
COMMAND\ifresumenummering@start	98
COMMAND\ifseriesbefore	273
COMMAND\ifstopmsdata@inserted@	341
COMMAND\ifsublines@	108, 121
COMMAND\iftrue	461
COMMAND\ifvmode	294
COMMAND\ifxxx	374
COMMAND\ignorespaces	141
COMMAND\imki@wrindexentry	85
COMMAND\immediate	128, 129, 247
COMMAND\indent	19, 159, 398
COMMAND\index	330, 467
COMMAND\indtl@wrindexentry	85
COMMAND\initnumbering@quote	372, 462
COMMAND\initnumbering@reg	452
COMMAND\initnumbering@sectcmd	462
COMMAND\inplaceoflemmaseparator	397
COMMAND\inplaceofnumber	397
COMMAND\insert	143, 178, 186, 188, 189, 203, 220, 226
COMMAND\insert@Xtxtbeforenotes	182, 203
COMMAND\insert@count	125, 131, 141
COMMAND\insert@countR	141
COMMAND\insert@msdata	342, 349
COMMAND\insert@txtbeforenotesX	182
COMMAND\inserthangingsymbol	455
COMMAND\insertlines@list	109, 125, 126
COMMAND\insertparafootsepX	240
COMMAND\inserts@list	136, 154, 178, 194
COMMAND\interAfootnotelinepenalty	453
COMMAND\interfootnotelinepenalty	453
COMMAND\interlinepenalty	186
COMMAND\interparanoteglue	395
COMMAND\justifying	52
COMMAND\keep@this@crossref@forinserting	221
COMMAND\l@advance@parledgroup@beforenormalnotes	462
COMMAND\l@d@@wrindexhyp	458

COMMAND\l@d@add	146
COMMAND\l@d@end	247, 269
COMMAND\l@d@nums	140, 143, 145, 146, 212, 213
COMMAND\l@d@section	247
COMMAND\l@d@set	120, 134
COMMAND\l@dampcount	357
COMMAND\l@dbfnote	224, 452
COMMAND\l@dcheckstartend	366
COMMAND\l@dchset@num	120
COMMAND\l@dcolcount	357, 358
COMMAND\l@dcollect@@body	351
COMMAND\l@dcollect@body	350
COMMAND\l@dcsnote	457
COMMAND\l@dcsnotetext	165, 313
COMMAND\l@dcsnotetext@l	165, 313
COMMAND\l@dcsnotetext@r	165, 313
COMMAND\l@ddodoreinxtrafeet	286, 451
COMMAND\l@ddoxtrafeet	284, 286, 451
COMMAND\l@demptyd@ta	452
COMMAND\l@dend@close	247
COMMAND\l@dend@open	247
COMMAND\l@dend@stuff	247
COMMAND\l@denvbody	350
COMMAND\l@dfeetbeginmini	451
COMMAND\l@dfeetendmini	451
COMMAND\l@dgetline@margin	453
COMMAND\l@dgetlock@disp	453
COMMAND\l@dgetref@num	300
COMMAND\l@dgetsidenote@margin	309, 453
COMMAND\l@dgobbeloptarg	458
COMMAND\l@dgonblearg	458
COMMAND\l@dgonbleoptarg	356
COMMAND\l@dlabel@parse	300
COMMAND\l@dld@ta	172, 174
COMMAND\l@dlp@rbox	314
COMMAND\l@dlsn@te	453
COMMAND\l@dlsnote	457
COMMAND\l@dmake@labels	294, 308
COMMAND\l@dmake@labelsR	308
COMMAND\l@dnumpstartsL	95, 453
COMMAND\l@dp@rsefootspec	213
COMMAND\l@dp@rsefootspec	213
COMMAND\l@dpush@begins	351
COMMAND\l@d@rd@ta	172, 174
COMMAND\l@dref@undefined	299
COMMAND\l@d@rsn@te	453
COMMAND\l@d@rsnote	457
COMMAND\l@dtabaddcols	366
COMMAND\l@dtabnoexpands	451
COMMAND\l@dumboxmpfoot	462

COMMAND\l@dunboxmpfoot	452
COMMAND\l@dunhbox@line	160
COMMAND\l@dzeropenalties	453, 458
COMMAND\l@pb	387
COMMAND\l@prev@nopb	386
COMMAND\l@prev@pb	386
COMMAND\l@reg	393
COMMAND\l@secondoptarg	356
COMMAND\l@wrap@ifnotemptybox	92
COMMAND\label	20, 64, 67, 70, 292, 293, 301
COMMAND\label@refs	292
COMMAND\labelstarttrue	20, 400, 454, 469
COMMAND\labelref@list	292, 295
COMMAND\language	195
COMMAND\lastbox	159
COMMAND\lastskip	133
COMMAND\latex@makecol	287
COMMAND\leavevmode	19, 159
COMMAND\led@check@nopb	387
COMMAND\led@check@pb	387
COMMAND\led@nopb	386, 387
COMMAND\led@nopbnum	386
COMMAND\led@pb	386, 387
COMMAND\led@pb@macro	386
COMMAND\led@pbnum	386
COMMAND\led@reinit@index@fornote	330
COMMAND\led@set@index@fornote	330
COMMAND\ledRflag	327
COMMAND\ledchapter	395, 454
COMMAND\ledfootinsdim	396
COMMAND\ledinnernote	67, 310, 457, 466
COMMAND\ledinnote	325, 394, 461
COMMAND\ledinnotemark	69, 394, 460
COMMAND\ledleftnote	67, 310
COMMAND\ledlinenum	107, 453
COMMAND\ledllfill	165
COMMAND\ledlsnotefontsetup	466
COMMAND\ledlsnotesep	68
COMMAND\ledlsnotewidth	67
COMMAND\lednopb	76, 386
COMMAND\lednopbinverse	387
COMMAND\lednopbinversetrue	59, 76
COMMAND\lednopbnum	386
COMMAND\ledouternote	67, 310, 457, 466
COMMAND\ledpb	76, 386, 395
COMMAND\ledpbnum	386
COMMAND\ledpbsetting	76, 386, 463
COMMAND\ledrightnote	67, 310
COMMAND\ledrsnotefontsetup	466
COMMAND\ledrsnotesep	68

COMMAND\ledrsnotewidth	67
COMMAND\ledsection	395
COMMAND\ledsectnomark	374
COMMAND\ledsectnotoc	373
COMMAND\ledsetnormalparstuff	394, 395, 460
COMMAND\ledsetnormalparstuff@common	242
COMMAND\ledsetnormalparstuffX	394, 395, 460
COMMAND\ledsidenote	67, 310, 313
COMMAND\ledsubsection	395
COMMAND\ledsubsubsection	395
COMMAND\ledxxx	457
COMMAND\left	73
COMMAND\leftctab	367
COMMAND\leftheadline	106
COMMAND\leftlinenum	23, 106, 451, 453
COMMAND\leftltab	367
COMMAND\leftnoteupfalse	68
COMMAND\leftpstartnum	177
COMMAND\leftrtab	367
COMMAND\leftsidenote	313
COMMAND\leftskip	191, 194, 195
COMMAND\lemma	3, 29, 32, 34, 35, 37, 136, 140, 141, 144, 145, 147, 391, 453, 454, 461, 462, 464
COMMAND\lemmaseparator	397
COMMAND\let	33, 57, 138, 143, 268, 287, 338, 451
COMMAND\letboxXendendlinenum	397
COMMAND\line	200, 203
COMMAND\line@list	109, 111, 126, 143
COMMAND\line@list@stuff	95, 112, 113, 129, 451, 453
COMMAND\line@list@version	115
COMMAND\line@margin	103, 174, 309
COMMAND\line@num	108–110, 172, 451
COMMAND\line@set	145, 146
COMMAND\lineannot	26, 63, 146
COMMAND\lineannot@set	146
COMMAND\lineation	22, 101, 469, 470
COMMAND\linebreak	43
COMMAND\lineneation	465
COMMAND\linenum	26, 29, 32, 63, 64, 136, 145, 146, 298, 300, 308, 391
COMMAND\linenum@out	128, 292, 295
COMMAND\linenumannotation	25, 170, 469, 470
COMMAND\linenumberlist	22, 93, 173, 451
COMMAND\linenumberstyle	24, 106, 451, 468
COMMAND\linenumincrement	22, 453
COMMAND\linenummargin	22, 103, 309
COMMAND\linenumr@p	106, 451, 453
COMMAND\linenumrep	106, 453
COMMAND\linenumsep	23, 68, 106, 310
COMMAND\linerrangesep@	283
COMMAND\lineref	291, 298, 307, 397, 459
COMMAND\list@clear	100

COMMAND\list@clearing@reg	453
COMMAND\list@create	100
COMMAND\lock@disp	105
COMMAND\lock@off	123
COMMAND\lock@on	122
COMMAND\lockdisp	23, 105
COMMAND\loop	181, 333
COMMAND\ltab	367, 368, 372
COMMAND\ltabtext	372
COMMAND\m@mmf@prepare	223
COMMAND\mainmatter	469, 470
COMMAND\makeatletter	164
COMMAND\makehboxoffhboxes	196, 197
COMMAND\makeindex	69, 326
COMMAND\makelabel	308
COMMAND\managestanza@modulo	335
COMMAND\marginpar	67, 78, 309, 452
COMMAND\marginparwidth	67, 310
COMMAND\markboth	28, 164
COMMAND\mathchardef	334
COMMAND\maxXnotes	397
COMMAND\maxhnotesX	55, 79, 396, 456, 457, 459, 460
COMMAND\maxlinesinpar@list	113
COMMAND\measurebody	369
COMMAND\measuretbody	369
COMMAND\memorybreak	21
COMMAND\metaXXXvfootnote	220
COMMAND\microtype	160
COMMAND\morenoexpands	79, 136, 138
COMMAND\morethantwolines	397
COMMAND\mpfnpos	42, 244, 456, 467
COMMAND\mpnormalfootgroup	451
COMMAND\mpnormalvfootnote	452
COMMAND\msdata	38, 39, 340, 341, 343, 468, 470
COMMAND\msdataposition	40
COMMAND\multfootsep	40, 222
COMMAND\multiplefootnotemarker	222
COMMAND\musixtex	456
COMMAND\n@num	453, 460
COMMAND\n@num@ref	460
COMMAND\new@line	130, 452
COMMAND\new@series	138
COMMAND\newcommand	33, 81, 97, 222, 294
COMMAND\newcommandx	33
COMMAND\newhookarg@specific	282
COMMAND\newhookcommand@series	276, 277, 460
COMMAND\newhookcommand@series@reload	277
COMMAND\newhookcommand@toggle@reload	277, 458
COMMAND\newhooktoggle@series	276, 277, 460
COMMAND\newhooktoggle@specific	282

COMMAND\newif	460
COMMAND\newline	43
COMMAND\newlinechar	269
COMMAND\newseries	41, 392, 395, 396
COMMAND\newseries@	260, 274
COMMAND\newverse	60, 61, 395, 457, 468
COMMAND\next	333
COMMAND\next@action	113
COMMAND\next@actionline	113
COMMAND\next@insert	178
COMMAND\next@line@list@stuff	129, 469
COMMAND\nl@regR	116
COMMAND\no@expands	79, 138, 144, 268, 451
COMMAND\noXlemmaseparator	397
COMMAND\nobreak	212
COMMAND\nocritical	261
COMMAND\noeledsec	75, 395
COMMAND\noendnotes	395
COMMAND\noexpand	393
COMMAND\nofamiliar	279
COMMAND\noindent	19, 59, 60, 159, 463
COMMAND\noindentX	463
COMMAND\nomk@	283
COMMAND\nonbreakableafternumber	397
COMMAND\nonnumberinfootnote	397
COMMAND\norelax	57
COMMAND\normal@footnotemarkX	226
COMMAND\normal@page@break	386
COMMAND\normal@pars	242
COMMAND\normalbfnoteX	453
COMMAND\normalbodyfootmarkX	226
COMMAND\normalfont	467
COMMAND\normalfootfmt	57, 190, 197, 207, 247, 466
COMMAND\normalfootfmtX	227, 228
COMMAND\normalfootfootmarkX	228
COMMAND\normalfootgroup	192
COMMAND\normalfootgroupX	229
COMMAND\normalfootnoterule	187
COMMAND\normalfootstart	190, 194
COMMAND\normalfootstartX	228
COMMAND\normalvfootnote	188, 189
COMMAND\normalvfootnote@inserted	189
COMMAND\normalvfootnoteX	226
COMMAND\notbool	374
COMMAND\notfontsetup	396
COMMAND\notfontsizeX	51, 396
COMMAND\notenumfont	396
COMMAND\notenumfontX	50, 396
COMMAND\notesXwidthliketwocolumns	397
COMMAND\noteswidthliketwocolumnsX	56, 397, 458, 460

COMMAND\num@lines	154, 180
COMMAND\numberlinefalse	21, 470
COMMAND\numberlinetrue	21, 470
COMMAND\numberonlyfirstinline	274, 397
COMMAND\numberonlyfirstintwolines	397
COMMAND\numberpstartfalse	20
COMMAND\numberpstarttrue	20, 47, 394, 453, 462, 468, 470
COMMAND\numberstanza	47
COMMAND\numberstanzafalse	60
COMMAND\numberstanzatrue	60
COMMAND\numlabfont	23, 56, 107
COMMAND\one@line	154
COMMAND\onehalfspacing	463
COMMAND\onlyXpstart	397
COMMAND\onlysideX	267
COMMAND\page@action	110, 120
COMMAND\page@start	110, 453
COMMAND\pagecontents	110
COMMAND\pagenumbering	290
COMMAND\pageparbreak	395
COMMAND\pageref	64, 298
COMMAND\par	32, 43, 159, 242
COMMAND\par@line	154, 180
COMMAND\par@patch@pagenumbering	290
COMMAND\par@patch@thepage	130, 290
COMMAND\para@footgroup	194
COMMAND\para@footgroupX	239
COMMAND\para@footsetup	193, 451
COMMAND\para@footsetupX	237, 451, 458
COMMAND\para@vfootnoteX	238
COMMAND\parafootfmt	196, 197, 466
COMMAND\parafootfmtX	239
COMMAND\parafootftm	199
COMMAND\parafootftmX	240
COMMAND\parafootftmsep	396
COMMAND\parafootsep	397, 456, 461
COMMAND\parafootsepX	53, 112, 396, 397, 466
COMMAND\parafootstart	194
COMMAND\parafootstartX	237
COMMAND\paravfootnote	194, 197
COMMAND\parfillskip	197
COMMAND\parindent	463
COMMAND\parindentX	51, 399, 463, 465
COMMAND\parse@annot	172
COMMAND\parshape	78
COMMAND\parskip	19, 59, 60, 159
COMMAND\pause@numbering	98
COMMAND\pausenumbering	20, 98, 113, 115, 160, 456, 458, 465
COMMAND\pausenumbering@page@num	98
COMMAND\penalty	197

COMMAND\pend	2, 7, 18, 19, 21, 22, 24, 75, 76, 134, 136, 139, 146, 154, 156–160, 177, 178, 393, 456, 457, 466–469
COMMAND\preXnotes	460, 466
COMMAND\prenotesX	54, 246, 460
COMMAND\prepare@Xgroupbyline	220, 221
COMMAND\prepare@Xprenotes	244
COMMAND\prev@nopb	386
COMMAND\prev@pb	386
COMMAND\prevlineX	112
COMMAND\prevpageX@num	112
COMMAND\print@Xfootnoterule	460
COMMAND\print@Xnotes	285, 286
COMMAND\print@Xnotes@forpages	459
COMMAND\print@eledsection	163
COMMAND\print@footnoteXrule	460
COMMAND\print@leftmargin@eledsection	375
COMMAND\print@lemma	208
COMMAND\print@line	161
COMMAND\print@notesX@forpages	459
COMMAND\print@rightmargin@eledsection	375
COMMAND\printendlines	252, 260, 305, 399, 451, 453
COMMAND\printlinefootnote	209, 211, 459
COMMAND\printlinefootnotearea	211, 212, 459
COMMAND\printlinefootnotenumbers	210
COMMAND\printlines	190, 207, 212, 214, 252, 305, 399, 451, 453, 459, 464
COMMAND\printnpnum	394, 399
COMMAND\printpstart	208
COMMAND\protect	138, 393
COMMAND\providecommand	222, 451
COMMAND\pstarL	154
COMMAND\pstart	2, 7, 18–22, 24, 74–76, 120, 134, 139, 143, 146, 154, 157–159, 163, 178, 380, 393, 394, 453–455, 457, 458, 460–462, 465–470
COMMAND\pstartinfootnote	397
COMMAND\pstartinfootnoteeverytime	397
COMMAND\pstartnum	177
COMMAND\pstartref	62, 291, 299, 456, 468
COMMAND\pstarts	453, 454
COMMAND\pstarts@readL	154
COMMAND\pstarts@typeset@L	154
COMMAND\raggedX	53
COMMAND\raggedleft	53
COMMAND\raggedright	52
COMMAND\raw@text	154
COMMAND\rbracket	49, 50
COMMAND\read@linelist	112, 114
COMMAND\ref	64, 67, 70
COMMAND\reformatted@	305
COMMAND\reformattedwithpage	305
COMMAND\relax	19, 120, 143, 167, 178, 338, 356, 393
COMMAND\renewcommand	78, 396, 399



COMMAND\reset@current@annot	171
COMMAND\reset@msd@options@iffullpage	349
COMMAND\resetprevline@	112
COMMAND\resetprevpage@	112
COMMAND\resumenummering	20, 94, 98, 99, 113, 115, 160, 453, 457, 458, 465
COMMAND\right	73
COMMAND\rightctab	367
COMMAND\rightlinenum	23, 106, 451, 453
COMMAND\rightltab	368
COMMAND\rightnoteupfalse	68
COMMAND\rightrtab	368
COMMAND\rightsidenote	313
COMMAND\rightright	191, 194, 195, 197
COMMAND\rightrightnum	177
COMMAND\rigidbalance	200, 201, 203, 399, 464
COMMAND\rigidbalanceX	200, 399, 464
COMMAND\robustify	44
COMMAND\roman	361, 464
COMMAND\rtab	367, 368, 372
COMMAND\rtabtext	369, 372
COMMAND\sameparallelnumber	130
COMMAND\sameword	16, 33–35, 37, 128, 146–150, 152, 401, 459, 461, 463, 466, 469
COMMAND\sameword@inedtext	148
COMMAND\saweword	147
COMMAND\scriptsize	107
COMMAND\section	74, 453, 467
COMMAND\section@num	94
COMMAND\sectionmark	374
COMMAND\select@lemmafont	56, 57, 184
COMMAND\series	260
COMMAND\series@	260
COMMAND\seriesatbegin	41, 273, 460
COMMAND\seriesatend	41, 273, 460
COMMAND\set@Xtxtbeforenotes	182
COMMAND\set@continuousnumberingforl	99
COMMAND\set@line	143
COMMAND\set@line@action	111, 121
COMMAND\set@txtbeforenotesX	182
COMMAND\setSErefonlypageprefixmore	65, 305, 465, 468
COMMAND\setSErefonlypageprefixsingle	65, 305, 465, 468
COMMAND\setSErefprefixmore	65
COMMAND\setSErefprefixsingle	65
COMMAND\setapprefprefixmore	65, 396
COMMAND\setapprefprefixsingle	65, 66, 396, 464
COMMAND\setcommand@series	275
COMMAND\sethangindentX	207
COMMAND\sethangingsymbol	59, 333, 396, 463
COMMAND\sethanginsymbol	57
COMMAND\setistwofollowinglines	217
COMMAND\setl@dlprbox	314

COMMAND\setline	24, 111, 115, 120, 134, 138, 157, 462
COMMAND\setlinenum	24, 115, 120, 134, 451, 469
COMMAND\setlinenumannotationsep	27, 400, 470
COMMAND\setmsdatalabel	39
COMMAND\setmsdataposition	468
COMMAND\setmsdataseries	39
COMMAND\setprintendlines	252, 256, 453
COMMAND\setprintlines	214, 217, 252, 453
COMMAND\setsidenoteseq	68
COMMAND\setsidenotsep	396
COMMAND\setstanzaindent	335
COMMAND\setstanzaindents	58, 334, 393
COMMAND\setstanzapenalties	334
COMMAND\setstanzavalues	334
COMMAND\settoggle@series	274, 454, 458
COMMAND\showlemma	137, 451, 452
COMMAND\showwordrank	37, 148
COMMAND\sidenote@margin	452
COMMAND\sidenotemargin	67, 452, 457
COMMAND\sidenoteseq	396
COMMAND\sidepstartnumtrue	20
COMMAND\skip	191
COMMAND\skipnumbering	24, 123, 124, 135, 453, 460, 461
COMMAND\skipnumbering@reg	460
COMMAND\small	51
COMMAND\somemacro	37
COMMAND\special	13
COMMAND\splitmaxdepth	186, 202
COMMAND\splitoff	200
COMMAND\splittopskip	186, 202, 203
COMMAND\stanza	23, 24, 59–61, 338, 396, 398, 463, 468
COMMAND\stanza@hang	337
COMMAND\stanza@line	337
COMMAND\stanzaindent	58, 335, 459
COMMAND\stanzaindent*	58
COMMAND\stanzaindentbase	334
COMMAND\stanzanumwrapper	60
COMMAND\startlock	21, 23, 111, 135, 338
COMMAND\startstanzahook	396
COMMAND\startsub	23, 110, 133
COMMAND\stopmsd	343
COMMAND\stopmsdata	39, 341
COMMAND\store@annot@to@absline	171
COMMAND\strip@pt	194
COMMAND\strutbox	202
COMMAND\sub@action	110, 121
COMMAND\sub@lock	109
COMMAND\sub@off	118, 295
COMMAND\sub@on	118, 295
COMMAND\subline@num	108–110

COMMAND\sublinenum@rep	451
COMMAND\sublinenumberstyle	24, 106, 451
COMMAND\sublinenumincrement	22
COMMAND\sublinenumr@p	106, 451, 453
COMMAND\sublinenumrep	106, 453
COMMAND\sublineref	62, 291, 298
COMMAND\subsectionmark	374
COMMAND\sw@inthisedtext	140
COMMAND\sw@list@inedtext	144, 152
COMMAND\sw@txt	149
COMMAND\swnoexpands	37, 469
COMMAND\symlinenum	397
COMMAND\symplinum	396
COMMAND\sza@penalty	337
COMMAND>tag	459
COMMAND\text	391
COMMAND\text{ <i>language</i> }	51
COMMAND\textbeforenotesX	182
COMMAND\textcolor	79
COMMAND\textheight	79
COMMAND\textsuperscript	27, 92, 170
COMMAND\the	130, 451
COMMAND\thefootnoteA	40
COMMAND\thefootnoteX	455
COMMAND\thelabidx	328, 329
COMMAND\thepage	115, 130, 290
COMMAND\thepstart	20, 400
COMMAND\thepstartL	454
COMMAND\thepstartR	454
COMMAND\thestanza	60
COMMAND\this@crossref@end	221
COMMAND\this@line@list@version	128
COMMAND\this@section@next@page@num	98
COMMAND\this@section@page@num	111
COMMAND\thisfootnote	230
COMMAND\threecol@begin@insert	203
COMMAND\threecolfootfmt	202, 203, 466
COMMAND\threecolfootfmtX	235
COMMAND\threecolfootgroup	201
COMMAND\threecolfootgroupX	236
COMMAND\threecolfootsetup	201
COMMAND\threecolfootsetupX	235
COMMAND\threecolvfootnote	202
COMMAND\threecolvfootnote@inserted	202
COMMAND\threecolvfootnoteX	235
COMMAND\toendnotes	31, 251, 466
COMMAND\twocolfootfmt	466
COMMAND\twocolfootfmtX	233
COMMAND\twocolfootgroupX	234
COMMAND\twocolfootsetupX	233

COMMAND\twocolvfootnoteX	233
COMMAND\twolines	274, 397
COMMAND\twolines@A	274
COMMAND\twolines@B	274
COMMAND\twolines@C	274
COMMAND\twolinesbutnotmore	397
COMMAND\twolinesonlyinsamepage	397
COMMAND\txtbeforeXnotes	397
COMMAND\txtbeforenotesX	54, 469
COMMAND\txtbeforenotesonlyonceX	54, 469
COMMAND\uline	41
COMMAND\unhbox	160, 195
COMMAND\unpenalty	196, 197
COMMAND\unskip	197
COMMAND\unvxh	197, 398
COMMAND\unvxhX	398
COMMAND\upbracefill	366
COMMAND\usingcritext	392, 395
COMMAND\usingdtext	392, 395
COMMAND\vAfootnote	186
COMMAND\vadjust	132
COMMAND\variant	33
COMMAND\vbox	156, 159, 195, 200, 244
COMMAND\vfootnote	186, 191, 194, 202
COMMAND\vl@dbfnote	224, 452
COMMAND\vl@disnote	313
COMMAND\vl@dlsnote	313
COMMAND\vl@dosnote	313
COMMAND\vl@drsnote	313
COMMAND\vnunfootnoteX	453
COMMAND\vsizer	55, 79
COMMAND\vsplit	180
COMMAND\waklam	366
COMMAND\waklamec	366
COMMAND\wapunktel	366
COMMAND\wastricht	366
COMMAND\widthX	55, 399, 465
COMMAND\wrap@edcrossref	297, 458
COMMAND\wrapcontentX	51, 466
COMMAND\wrapped@bodyfootmarkX	241
COMMAND\wrapped@footfootmarkX	241
COMMAND\x...	63
COMMAND\xabslineref	299
COMMAND\xannotationref	63
COMMAND\xdef	100, 338
COMMAND\xflagref	63, 299, 399, 464
COMMAND\xleft@appenditem	101, 137
COMMAND\xlineref	63, 399, 464
COMMAND\xpageref	63
COMMAND\xparse	38

COMMAND\xpstartref	63, 456, 469
COMMAND\xr	67
COMMAND\xright@appenditem	100, 101
COMMAND\xsublineref	63
COMMAND\xxref	63, 300, 301, 308, 456, 459, 466
COMMAND\xxxmatter	290
ENVIRONMENTastanza	464
ENVIRONMENTedarrayc	372
ENVIRONMENTedarrayl	372
ENVIRONMENTedarrayr	372
ENVIRONMENTedtabularc	372
ENVIRONMENTedtabularl	372
ENVIRONMENTedtabularr	372
ENVIRONMENTledgroup	85, 320, 399, 464
ENVIRONMENTledgroupsize	320
PACKAGE(r)(e)ledmac	41
PACKAGEEledmac	110, 325, 394, 395, 459, 460
PACKAGEEledpar	460
PACKAGEEtoolbox	84
PACKAGEParallel	402
PACKAGEReledmac	398, 399
PACKAGEamsgen	350
PACKAGEamsmath	349, 350
PACKAGEbabel	51, 80, 361, 464
PACKAGEbiblatex	77
PACKAGEbidi	51, 52, 85, 463
PACKAGEccaption	93
PACKAGEcolor	79
PACKAGEcsquotes	468
PACKAGEedmac	1, 6, 11, 13, 14, 81, 212, 222, 292, 334, 391, 402, 451
PACKAGEedstanza	1, 14, 332
PACKAGEeledmac	1, 11, 15, 16, 70, 222, 322, 325, 353, 376, 389, 392, 394–396, 455, 457, 459
PACKAGEeledpar	186, 402, 453, 457–459
PACKAGEetex	463
PACKAGEebox	100, 146, 260, 274, 284, 313, 374, 386
PACKAGEfancyhdr	287, 288, 467
PACKAGEfloatrow	77, 399
PACKAGEfootmisc	40, 80, 85, 222, 402, 467
PACKAGEgeometry	15
PACKAGEglossaries	70, 71, 330, 464
PACKAGEhandout	458
PACKAGEhyperlink	268
PACKAGEhyperref	63, 140, 241, 293, 328, 378, 388, 456–458, 465, 468, 471
PACKAGEifluatex	84
PACKAGEifxetex	84
PACKAGEimakeidx	68, 69, 78, 85, 322, 325, 395, 455–457, 459
PACKAGEindextols	330
PACKAGEindextool	395
PACKAGEindextools	68, 69, 78, 85, 94, 322, 325, 330, 395, 459, 464, 467
PACKAGEinputenc	148

PACKAGEledarab	80
PACKAGEledmac	1, 11, 14, 15, 80, 100, 325, 391, 392, 395, 398
PACKAGEledpar	80
PACKAGEMemoir	85, 325, 395, 402, 457
PACKAGEMorewrites	78
PACKAGEMusixtex	457
PACKAGEperpage	464
PACKAGEpolyglossia	50, 80, 185, 208, 464
PACKAGERagged2e	52, 53, 84
PACKAGEreledmac	1, 2, 6, 11–13, 15–18, 20–22, 25, 28, 30, 32–35, 38, 40, 41, 44–46, 48, 51, 52, 54–57, 59, 61–64, 67–71, 75–82, 101, 104, 110, 111, 114, 115, 118, 128–130, 138, 141, 143, 146, 147, 178, 187, 191, 195, 207, 222, 247, 261, 265, 266, 274, 284, 290, 297, 299, 302, 308, 325, 346, 353, 374, 375, 387, 388, 395, 396, 398–401, 462, 465, 467
PACKAGEreledpar	1, 4, 6, 9, 16, 19, 21, 41, 56, 63, 65, 66, 76, 77, 80, 82, 93, 101, 113, 118, 129, 130, 141, 143, 154, 188, 191, 242, 243, 260, 267, 283, 285, 286, 290, 321, 332, 374, 463, 464, 466, 467, 469, 470
PACKAGESuffix	84
PACKAGETabmac	1, 14, 402
PACKAGEulem	41
PACKAGEuninormalize	34
PACKAGExargs	33, 84
PACKAGEXkeyval	82, 283
PACKAGEXparse	84, 470
PACKAGEXr	5, 67, 308, 465
PACKAGEXref	308
PACKAGEXstring	84, 328

## 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
\add@penalties	1
\add@Xgroupbyline	1
\addtol@denbody	1
Adelard II	14
\advancelabel@refs	1
\advanceline	1, 23
\Aendnote	30
\affixline@num	1
\affixpstart@num	1
\affixside@note	1
\Afootnote	30

<code>\afternoteX</code>	53
<code>\afterruleX</code>	55
<code>\ampersand</code>	1, 61
<code>\annot@list</code>	1
<code>\annotationref</code>	1, 62
<code>\append@notesX</code>	1
<code>\append@Xnotes</code>	1
<code>\applabel</code>	1, 64
<code>\appref</code>	1, 64
<code>\apprefwithpage</code>	1, 64
<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, 19
<code>\AtEveryPend</code>	1, 19
<code>\AtEveryPstart</code>	1, 19
<code>\AtEveryStanza</code>	1, 59
<code>\AtEveryStopStanza</code>	1, 60
<code>\AtStartEveryPstart</code>	1, 19
<code>\AtStartEveryStanza</code>	1, 59
<code>\autopar</code>	1, 18

## B

<code>\ballast</code>	78
<code>\ballast@count</code>	1
Beeton, Barbara Ann Neuhaus Friend	19
<code>\beforeeledchapter</code>	1
<code>\BeforeEveryStopStanza</code>	1, 60
<code>\beforeinsertingX</code>	52
<code>\beforeinsertion@X</code>	1
<code>\beforenotesX</code>	54
<code>\beginnumbering</code>	1, 17
<code>\Bendnote</code>	30
<code>\Bfootnote</code>	30
<code>\bhookgroupX</code>	54
<code>\bhooknoteX</code>	52
<code>\bodyfootmarkA</code>	40
<code>\boxfootnotenumbers</code>	1
Bredon, Simon	14
Breger, Herbert	14, 353
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> .....	52
<code>\continuousnumberingwithcolumns@sync@linenumber@singletext</code> .....	1
Copernicus, Nicolaus .....	14
<code>\critext</code> .....	391
<code>\ctab</code> .....	1
<code>\ctabtext</code> .....	1
<code>\current@annot</code> .....	1

## D

Dekker, Dirk-Jan .....	79
<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 .....	78, 195, 197
<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, 73, 367
<code>edarrayc</code> (environment) .....	71
<code>edarrayl</code> (environment) .....	71
<code>edarrayr</code> (environment) .....	71
<code>\edatleft</code> .....	1, 73
<code>\edatright</code> .....	1, 73
<code>\edbforetab</code> .....	1, 73, 367
<code>\edfilldimen</code> .....	1
<code>\edfont@info</code> .....	1
<code>\edindex</code> .....	1, 68
<code>\edindexlab</code> .....	1, 70
<code>\EDLABEL</code> .....	1
<code>\edlabel</code> .....	1, 62
<code>\edlabelE</code> .....	1, 64
<code>\edlabelS</code> .....	1, 64
<code>\edlabelSE</code> .....	1, 64
<code>\edlineref</code> .....	1, 62
<code>\edmakelabel</code> .....	1, 63
<code>\edpageref</code> .....	1, 62
<code>\edrowfill</code> .....	1, 72
<code>\EDTAB</code> .....	1
<code>\edtabcolsep</code> .....	1, 72
<code>\EDTABINDENT</code> .....	1
<code>\edtabindent</code> .....	1
<code>\EDTABtext</code> .....	1
<code>edtabularc</code> (environment) .....	71
<code>edtabularl</code> (environment) .....	71
<code>edtabularr</code> (environment) .....	71
<code>\EDTEXT</code> .....	1
<code>\edtext</code> .....	1, 29
<code>\edvertdots</code> .....	1, 74
<code>\edvertline</code> .....	1, 74
<code>\Eendnote</code> .....	30
<code>\Efootnote</code> .....	30
<code>\eled@chapter</code> .....	1
<code>\eled@section</code> .....	1
<code>\eled@sectioning@out</code> .....	1
<code>\eled@subsection</code> .....	1
<code>\eled@subsubsection</code> .....	1

<code>\eledchapter</code>	1
<code>\eledchapter*</code>	1
<code>\eledsection</code>	1
<code>\eledsection*</code>	1
<code>\eledsubsection</code>	1
<code>\eledsubsection*</code>	1
<code>\eledsubsubsection</code>	1
<code>\eledsubsubsection*</code>	1
<code>\enablel@dtabfeet</code>	1
<code>\end@lemmas</code>	1
<code>\endashchar</code>	1
<code>\endline@num</code>	1
<code>\endlock</code>	1, 23
<code>\endminipage</code>	1
<code>\endnumbering</code>	1, 17
<code>\endpage@num</code>	1
<code>\endprint</code>	1
<code>\endquotation</code>	1
<code>\endquote</code>	1
<code>\endsub</code>	1, 23
<code>\endsubline@num</code>	1
environments:	
<code>edarrayc</code>	71
<code>edarrayl</code>	71
<code>edarrayr</code>	71
<code>edtabularc</code>	71
<code>edtabularl</code>	71
<code>edtabularr</code>	71
<code>ledgroup</code>	61
<code>ledgroupsize</code>	61
<code>minipage</code>	61
<code>Euclid</code>	14
<code>\extensionchars</code>	1, 77

## F

<code>\f@x@l@cks</code>	1
Fairbairns, Robin	40
<code>\first@linenum@out@false</code>	1
<code>\first@linenum@out@true</code>	1
<code>\firstlinenum</code>	1, 22
<code>\firstseriesX@</code>	1
<code>\firstsublinenum</code>	1, 22
<code>\firstXseries@</code>	1
<code>\flag@end</code>	1
<code>\flag@end@later</code>	1
<code>\flag@end@RTL</code>	1
<code>\flag@start</code>	1
<code>\flag@start@later</code>	1
<code>\flag@start@RTL</code>	1
<code>\flagstanza</code>	1, 61

\flush@notes	1
\fnpos	1, 41
Folkerts, Menso	14
\footfootmarkA	40
\footfudgefiddle	1, 78
\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@fnmark	1
\get@fnmarkX	1
\get@index@command	1
\get@linelistfile	1
\get@sw@txt	1
\get@thisfootnote	1
\get@thisfootnoteX	1
\getline@num	1
\gl@p	1

## H

\h@num	1
\hangindentX	52
\hidenumbering	1, 24
\hidenumberingonleftpage	1, 24
\hidenumberingonrightpage	1
\Hilfsbox	1
\hilfsbox	1
\hilfscount	1
\HILFSskip	1
\Hilfsskip	1
\hilfsskip	1
\hookarg@specific	1
\hooktoggle@specific	1
\hsizethreecolX	53
\hsizetwocolX	53
\Hy@raisedlink@left	1

\hyperlinkformat	1
\hyperlinkformatR	1
\hyperlinkR	1

## I

\if@addsw	1
\if@edtext@secondarg@	1
\if@eled@sectioning	1
\if@firstlineofpage	1
\if@led@nofoot	1
\if@ledgroup	1
\if@lemmacommand@	1
\if@msdata@insertedfrompreviouspage	1
\if@noeled@sec	1
\if@noneed@Footnote	1
\if@RTL	1
\ifafterendnumberingR	1
\ifat@every@pend@star@	1
\ifat@every@pstart@star@	1
\ifautopar@pause	1
\ifbypage@	1
\ifbypage@R	1
\ifbypstart@	1
\ifbypstart@R	1
\ifeledmacompat@	1
\iffirst@linenum@out@	1
\ifindtl@innote	1
\ifindtl@notenumber	1
\ifinserthangingsymbol	1
\ifistanza	1
\ifl@d@dash	1
\ifl@d@elin	1
\ifl@d@esl	1
\ifl@d@pnum	1
\ifl@d@ssub	1
\ifl@d@Xmorethantwolines	1
\ifl@d@Xtwolines	1
\ifl@dend@X	1
\ifl@dhidenumber	1
\ifl@dmemoir	1
\ifl@dpaging	1
\ifl@dpairing	1
\ifl@dprintingcolumns	1
\ifl@dprintingpages	1
\ifl@dskipnumber	1
\ifl@dskipversenumber	1
\ifl@dstartendok	1
\ifl@footmisc	1
\ifl@imakeidx	1
\ifl@indextools	1

\ifledfinal .....	1, 77
\ifledgroupnotesL@ .....	1
\ifledgroupnotesR@ .....	1
\iflednopbinverse .....	1
\ifledRcol .....	1
\ifledRcol@ .....	1
\ifnocritical@ .....	1
\ifnoend@ .....	1
\ifnofamiliar@ .....	1
\ifnoledgroup@ .....	1
\ifnoquotation@ .....	1
\ifnoresetlinenumannotation@ .....	1
\ifnoteschanged@ .....	1
\ifnumberedpar@ .....	1
\ifnumbering .....	1
\ifnumberingR .....	1
\ifnumberline .....	1
\ifnumberstanza .....	1
\ifparapparatus@ .....	1
\ifparledgroup .....	1
\ifprevpgnotnumbered .....	1
\ifpst@rtedL .....	1
\ifresumenumbering@start .....	1
\ifsameparallelpagenumber .....	1
\ifseriesbefore .....	1
\ifsidepstartnum .....	1
\ifstopmsdata@inserted@ .....	1
\ifsublines@ .....	1
\ifwidthliketwocolumns .....	1
\ifXendinsertsep@ .....	1
\ifxindy@ .....	1
\ifxindyhyperref@ .....	1
\initnumbering@quote .....	1
\initnumbering@reg .....	1
\insert@count .....	0, 1
\insert@msdata .....	1
\insert@txtbeforenotesX .....	1
\insert@Xtxtbeforenotes .....	1
\insertthangingymbol .....	1
\insertlines@list .....	1
\insertparafootsepX .....	1
\inserts@list .....	1

## J

Jayaditya .....	14
-----------------	----

## K

Kabelschacht, Alois .....	181
\keep@this@crossref@forinserting .....	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@demptyd@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@denvbody</code> .....	1
<code>\l@desnote</code> .....	1
<code>\l@dfambeginmini</code> .....	1
<code>\l@dfamendmini</code> .....	1
<code>\l@dfeetbeginmini</code> .....	1
<code>\l@dfeetendmini</code> .....	1
<code>\l@dgetline@margin</code> .....	1
<code>\l@dgetlock@disp</code> .....	1
<code>\l@dgetref@num</code> .....	1
<code>\l@dgetsidenote@margin</code> .....	1
<code>\l@dobbleoptarg</code> .....	1
<code>\l@disnote</code> .....	1
<code>\l@dlabel@parse</code> .....	1
<code>\l@dld@ta</code> .....	1
<code>\l@dlp@rbox</code> .....	1
<code>\l@dlsn@te</code> .....	1
<code>\l@dlsnote</code> .....	1
<code>\l@dmake@labels</code> .....	1
<code>\l@dmodforedtext</code> .....	1
<code>\l@dnullfills</code> .....	1

<code>\ldnumpstartsL</code>	1
<code>\ldoldold@footnotetext</code>	1
<code>\ldp@rsefootspec</code>	1
<code>\ldparsedendline</code>	1
<code>\ldparsedendpage</code>	1
<code>\ldparsedendsub</code>	1
<code>\ldparsedstartline</code>	1
<code>\ldparsedstartpage</code>	1
<code>\ldparsedstartsub</code>	1
<code>\ldparsefootspec</code>	1
<code>\ldpush@begins</code>	1
<code>\ldrd@ta</code>	1
<code>\ldref@undefined</code>	1
<code>\ldrestorefills</code>	1
<code>\ldrestoreforedtext</code>	1
<code>\ldrp@rbox</code>	1
<code>\ldrsn@te</code>	1
<code>\ldrsnote</code>	1
<code>\ldsetmaxcolwidth</code>	1
<code>\ldskipnumberfalse</code>	1
<code>\ldskipnumbertrue</code>	1
<code>\ldtabaddcols</code>	1
<code>\ldtabnoexpands</code>	1
<code>\ldunboxmpfoot</code>	1
<code>\ldunhbox@line</code>	1
<code>\ldzeropenalties</code>	1
<code>\lsecondmandarg</code>	1
<code>\lthe@c@page</code>	1
<code>\lwrap@ifnotemptybox</code>	1
<code>\lwrapcs@ifnotemptybox</code>	1
<code>\label</code>	64
<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@toendnotes@outsidenumbering</code>	1
<code>\led@err@TooManyColumns</code>	1
<code>\led@err@UnequalColumns</code>	1
<code>\led@error@fail@patch@@doclearpage</code>	1
<code>\led@error@fail@patch@@iiiminipage</code>	1
<code>\led@error@fail@patch@@makecol</code>	1
<code>\led@error@fail@patch@@reinserts</code>	1
<code>\led@error@fail@patch@endminipage</code>	1
<code>\led@error@PackageAfterEledmac</code>	1
<code>\led@mess@NotesChanged</code>	1
<code>\led@mess@SectionContinued</code>	1
<code>\led@nopb</code>	1
<code>\led@nopbnum</code>	1
<code>\led@pb</code>	1
<code>\led@pb@setting</code>	1
<code>\led@pbnum</code>	1
<code>\led@reinit@index@fornote</code>	1
<code>\led@set@index@fornote</code>	1
<code>\led@toksa</code>	1
<code>\led@toksb</code>	1
<code>\led@warn@AppLabelOutSecondArgEdtext</code>	1
<code>\led@warn@BadAction</code>	1
<code>\led@warn@BadAdvancelineLine</code>	1
<code>\led@warn@BadAdvancelineSubline</code>	1
<code>\led@warn@BadLineation</code>	1
<code>\led@warn@BadLinenummargin</code>	1
<code>\led@warn@BadLockdisp</code>	1
<code>\led@warn@BadSetline</code>	1
<code>\led@warn@BadSetlinenum</code>	1
<code>\led@warn@BadSidenotemargin</code>	1
<code>\led@warn@BadSublockdisp</code>	1
<code>\led@warn@DuplicateLabel</code>	1
<code>\led@warn@edinde@outsidenumbering</code>	1
<code>\led@warn@LineFileObsolete</code>	1
<code>\led@warn@NoFile</code>	1
<code>\led@warn@NoIndexFile</code>	1
<code>\led@warn@NoMarginpars</code>	1
<code>\led@warn@RefUndefined</code>	1



\led@warn@SeriesStillExist	1
\led@warning@hsizeX@deprecated	1
\led@warning@msdatawithoutstop	1
\led@warning@preXnotes@deprecated	1
\led@warning@Xhsize@deprecated	1
ledgroup (environment)	61
ledgroupsize (environment)	61
\ledinnernote	1, 67
\ledinnote	1
\ledinnotehyperpage	1
\ledinnotemark	1
\ledleftnote	1, 67
\ledlinenum	1
\ledllfill	1
\ledlsnotefontsetup	1, 68
\ledlsnotesep	1, 68
\ledlsnotewidth	1, 67
\lednopb	1, 76
\lednopbinversetrue	76
\lednopbnum	1
\ledouternote	1, 67
\ledpb	1, 76
\ledpbnum	1
\ledpbsetting	1, 76
\ledrightnote	1, 67
\ledrlfill	1
\ledrsnotefontsetup	1, 68
\ledrsnotesep	1, 68
\ledrsnotewidth	1, 67
\ledsectnomark	1
\ledsectnotoc	1
\ledsetnormalparstuff@common	1
\ledsetnormalparstuffX	1
\ledsidenote	1, 67
\leftctab	1
\leftlinenum	1, 23
\leftltab	1
\leftnoteupfalse	68
\leftpstartnum	1
\leftrtab	1
Leibniz	14
\lemma	1, 32
\letsforverteilen	1
\line@list	1
\line@list@stuff	1
\line@list@version	1
\line@margin	1
\line@num	1
\line@set	1
\lineannot	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, 22
<code>\linenumr@p</code> .....	1
<code>\linenumrep</code> .....	1
<code>\linenumsep</code> .....	1, 23
<code>\linerangesep@</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, 23
Lorch, Richard .....	14
<code>\ltab</code> .....	1
<code>\ltabtext</code> .....	1
Luecking, Dan .....	84

## M

<code>\m@mmf@check</code> .....	1
<code>\m@mmf@prepare</code> .....	1
<code>\M@sect</code> .....	1
<code>\makehboxofhboxes</code> .....	1
<code>\manage stanza@modulo</code> .....	1
<code>\maxhnotesX</code> .....	55
Mayer, Gyula .....	14
<code>\measurebody</code> .....	1
<code>\measuremcell</code> .....	1
<code>\measuremrow</code> .....	1
<code>\measuretbody</code> .....	1
<code>\measuretcell</code> .....	1
<code>\measuretrow</code> .....	1
Middleton, Thomas .....	14, 108
<code>minipage</code> (environment) .....	61
Mittelbach, Frank .....	13
<code>\morenoexpands</code> .....	1, 79
<code>\mp@append@notesX</code> .....	1
<code>\mp@append@Xnotes</code> .....	1
<code>\mpfnpos</code> .....	1, 41
<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, 38
<code>\msdata@c</code> .....	1
<code>\msdata@cR</code> .....	1
<code>\multfootsep</code> .....	1, 40
<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>\noeledsec</code> .....	75
<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>	50
<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>	60
<code>\numberstanzatrue</code>	60
<code>\numlabfont</code>	1, 56

## O

<code>\old@hsize</code>	1
<code>\one@line</code>	1
<code>optionafter</code>	26
<code>optionauxdir</code>	16, 466
<code>optionbefore</code>	26
<code>optioncontinuousnumberingwithcolumns</code>	94, 162, 169, 465, 470
<code>optioninnnote</code>	464
<code>optioninnote</code>	464
<code>optionlinerrangesep</code>	283
<code>optionnocritical</code>	464
<code>optionnoeledsec</code>	380, 466
<code>optionnoend</code>	464
<code>optionnopenalties</code>	78
<code>optionnoprefix</code>	66, 471
<code>optionnoresetlinenumannotation</code>	25, 44, 470
<code>optionnotenumber</code>	464
<code>optionprefixmore</code>	66, 471
<code>optionsameparallelpagenumber</code>	470
<code>optionswcaseinsensitive</code>	34, 469

## P

<code>\page@action</code>	1
<code>\page@num</code>	1
<code>\pagelinesep</code>	1, 69
<code>\pageref</code>	64
<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> .....	53
<code>\parafootstart</code> .....	1
<code>\parafootstartX</code> .....	1
<code>\paravfootnote</code> .....	1
<code>\parindentX</code> .....	51
<code>\parse@annot</code> .....	1
<code>\pausenumbering</code> .....	1, 20
<code>\pausenumbering@page@num</code> .....	1
<code>\pend</code> .....	1, 18
Plato of Tivoli .....	14
<code>\postbodyfootmark</code> .....	1
<code>\prebodyfootmark</code> .....	1
<code>\prenotesX</code> .....	54
<code>\prepare@edindex@fornote</code> .....	1
<code>\prepare@prenotesX</code> .....	1
<code>\prepare@Xgroupbyline</code> .....	1
<code>\prepare@Xprenotes</code> .....	1
<code>\prev@nopb</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

\printXbeforenumber	1
\pstart	1, 18
\pstarteref	1
\pstartnum	1
\pstartref	62
\pstarts@read@L	1
\pstarts@typeset@L	1

## Q

\quotation	1
\quote	1

## R

\raggedX	53
\raw@text	1
\rbracket	1
\read@linelist	1
\ref	64
\ref@reg@later	1
\Relax	1
\reledmac@error	1
\reledmac@warning	1
\removehboxes	1
\reset@current@annot	1
\reset@msd@options@iffullpage	1
\resetprevline@	1, 112
\resetprevpage@	1
\resetprevpage@num	112
\restore@endnotes	1
\restore@familiarnotes	1
\restore@notes	1
\restore@sidenotes	1
\resumenumbering	1, 20
\rightctab	1
\rightlinenum	1, 23
\rightltab	1
\rightnoteupfalse	68
\rightrtab	1
\rightstartnum	1
\rigidbalance	1
\rigidbalanceX	1
\rtab	1
\rtabtext	1

## S

Sacrobosco	14
\sameword	1, 33
\sameword@inedtext	1
Schöpf, Rainer	13
\section@num	1

<code>\select@lemmafонт</code>	1
<code>\select@lemmafонт</code>	1, 56
<code>\Seref</code>	1, 64
<code>\Serefontypage</code>	64
<code>\Serefwithpage</code>	1, 64
<code>\series</code>	1
<code>\seriesatbegin</code>	1, 41
<code>\seriesatend</code>	1, 41
<code>\set@continuousnumberingforL</code>	1
<code>\set@line</code>	1
<code>\set@line@action</code>	1
<code>\set@txtbeforenotesX</code>	1
<code>\set@Xtxtbeforenotes</code>	1
<code>\setapprefprefixmore</code>	64
<code>\setapprefprefixsingle</code>	64
<code>\setcommand@series</code>	1
<code>\sethangindentX</code>	1
<code>\sethangingsymbol</code>	1, 59
<code>\setistwofollowinglines</code>	1
<code>\setl@dlp@rbox</code>	1
<code>\setl@drpr@box</code>	1
<code>\setline</code>	1, 23
<code>\setlinenum</code>	1, 24
<code>\setlinenumannotationsep</code>	1, 27
<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, 39
<code>\setmsdataposition</code>	1, 40
<code>\setmsdataseries</code>	1, 39
<code>\setnoteswidthliketwocolumnsX@</code>	1
<code>\setnotesXpositionliketwocolumns@</code>	1
<code>\setprintendlines</code>	1
<code>\setprintlines</code>	1
<code>\setSErefonlypageprefixmore</code>	65
<code>\setSErefonlypageprefixsingle</code>	65
<code>\setSErefprefixmore</code>	65
<code>\setSErefprefixsingle</code>	65
<code>\setsidenotessep</code>	68
<code>\setstanzaindents</code>	1, 57
<code>\setstanzapenalties</code>	1, 58
<code>\setstanzavalues</code>	1
<code>\settcclcenter</code>	1
<code>\settcclleft</code>	1
<code>\settcclright</code>	1
<code>\settoggle@series</code>	1
<code>\setthrowcenter</code>	1

<code>\settrorleft</code>	1
<code>\settrorright</code>	1
<code>\setXnotespositionliketwocolumns@</code>	1
<code>\setXnoteswidthliketwocolumns@</code>	1
<code>\showlemma</code>	1, 77
<code>\showwordrank</code>	1, 37
<code>\sidenote@margin</code>	1
<code>\sidenotemargin</code>	1, 67
<code>\sidepstartnumtrue</code>	20
<code>\skip@lockoff</code>	1
<code>\skipnumbering</code>	1, 24
<code>\splitoff</code>	1
<code>\spreadmath</code>	1, 72
<code>\spreadtext</code>	1, 72
<code>\stanza</code>	1, 57
<code>\stanza@count</code>	1
<code>\stanza@hang</code>	1
<code>\stanza@line</code>	1
<code>\stanzaindent</code>	1, 58
<code>\stanzaindent*</code>	1, 58
<code>\stanzaindentbase</code>	1, 57
<code>\stanzanumwrapper</code>	1, 60
<code>\startlock</code>	1, 23
<code>\startsub</code>	1, 23
<code>\step1@dcolcount</code>	1
<code>\stopmsdata</code>	1, 39
<code>\store@annot@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, 62
<code>\sublines@false</code>	1
<code>\sublines@true</code>	1
<code>\sublock@disp</code>	1
<code>\sublockdisp</code>	1
Sullivan, Wayne	13, 14, 78, 93, 98, 195, 196, 292, 332
<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, 60
<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@dcsnote</code>	1
<code>\vl@disnote</code>	1
<code>\vl@dlsnote</code>	1
<code>\vl@dosnote</code>	1
<code>\vl@drsnote</code>	1

\vnumfootnoteX ..... 1

## W

Whitney, Ron ..... 13  
 \widthX ..... 55  
 \wrap@edcrossref ..... 1  
 \wrapcontentX ..... 51  
 \wrapped@bodyfootmarkX ..... 1  
 \wrapped@footfootmarkX ..... 1  
 Wujastyk, Dominik ..... 13

## X

\X@atbegininsertion ..... 1  
 \X@beforeinsertion ..... 1  
 \X@doreinfeet ..... 1  
 \xabslineref ..... 1  
 \Xafterlemmaseparator ..... 49  
 \Xafternote ..... 53  
 \Xafternumber ..... 47  
 \Xafterrule ..... 55  
 \Xaftersymlinenum ..... 48  
 \xannotationref ..... 1, 62  
 \Xarrangement ..... 1, 43  
 \Xarrangement@normal ..... 1  
 \Xarrangement@paragraph ..... 1  
 \Xarrangement@threecol ..... 1  
 \Xarrangement@twocol ..... 1  
 \Xbeforeinserting ..... 52  
 \Xbeforelemmaseparator ..... 49  
 \Xbeforenotes ..... 54  
 \Xbeforenumber ..... 45, 47  
 \Xbeforesymlinenum ..... 48  
 \Xbhookgroup ..... 54  
 \Xbhooknote ..... 52  
 \Xboxlinenum ..... 48  
 \Xboxlinenumalign ..... 48  
 \Xboxsymlinenum ..... 48  
 \Xcolalign ..... 52  
 \xedindex ..... 1  
 \xedlabel ..... 1  
 \xedtext ..... 1  
 \Xendafterlemmaseparator ..... 50  
 \Xendafternote ..... 56  
 \Xendafterpagenumber ..... 45  
 \Xendaftersymlinenum ..... 48  
 \Xendaftrenumber ..... 47  
 \Xendahookinplaceofnumber ..... 49  
 \Xendahooklinenum ..... 49  
 \Xendbeforelemmaseparator ..... 50  
 \Xendbeforenumber ..... 47

\Xendbeforepagenumber .....	45
\Xendbeforeymlinenum .....	48
\Xendbhookinplaceofnumber .....	49
\Xendbhooklinenum .....	49
\Xendbhooknote .....	52
\Xendboxendlinenumalign .....	49
\Xendboxlinenum .....	49
\Xendboxlinenumalign .....	49
\Xendboxstartlinenumalign .....	49
\Xendboxymlinenum .....	48
\Xendhangindent .....	52
\Xendinplaceoflemmaseparator .....	50
\Xendinplaceofnumber .....	48
\Xendinplaceofpagenumber .....	45
\Xendlemmadisablefontselection .....	50
\Xendlemmafont .....	51
\Xendlemmaseparator .....	50
\Xendlinenumannotationonlyfirst .....	44
\Xendlinenumannotationonlyfirstintwo .....	44
\Xendlinenumannotationposition .....	26
\Xendlineprefixmore .....	45
\Xendlineprefixsingle .....	45
\Xendlinerrangeseparator .....	46
\Xendmorethantwolines .....	46
\Xendnoidenticallinenumannotation .....	27
\Xendnonumber .....	46
\Xendnotefontsize .....	51
\Xendnotenumfont .....	50
\Xendnumberonlyfirstinline .....	44
\Xendnumberonlyfirstintwolines .....	44
\Xendpagenumberonlyfirst .....	45
\Xendparagraph .....	56
\Xendsep .....	56
\Xendsublinesep .....	47
\Xendsymlinenum .....	44
\Xendsymlinenumannotation .....	44
\Xendsympagenum .....	45
\Xendtwolines .....	46
\Xendtwolinesbutnotmore .....	46
\Xendtxtbeforenotes .....	54
\Xendwrapcontent .....	51
\xflagref .....	1
\Xgroupbyline .....	53
\Xgroupbylineseparetwolines .....	54
\Xhangindent .....	51
\Xhsizethreecol .....	53
\Xhsizetwocol .....	53
\Xinplaceoflemmaseparator .....	50
\Xinplaceofnumber .....	48
\Xinsertparafootsep .....	1

\Xledsetnormalparstuff .....	1
\xleft@appenditem .....	1
\Xlemmadisablefontselection .....	50
\Xlemmafont .....	51
\Xlemmaseparator .....	49
\Xlinenumannotationonlyfirst .....	44
\Xlinenumannotationonlyfirstintwo .....	44
\Xlinenumannotationposition .....	26
\Xlinenumannotationposition@side .....	1
\Xlinrangeseparator .....	46
\xlineref .....	1, 62
\Xmaxhnotes .....	55
\Xmorethantwolines .....	46
\Xnoidenticallinenumannotation .....	27
\Xnolemmaseparator .....	1, 50
\Xnonbreakableafternumber .....	48
\Xnonumber .....	46
\Xnotefontsize .....	51
\Xnotenumfont .....	50
\Xnoteswidthliketwocolumns .....	56
\Xnumberonlyfirstinline .....	44
\Xnumberonlyfirstintwolines .....	44
\Xonlypstart .....	47
\Xpagelinesep .....	47
\xpageref .....	1, 62
\Xparafootsep .....	53
\Xparindent .....	51
\Xprenotes .....	1, 54
\Xprenotes@ .....	1
\Xpstart .....	46
\Xpstarteverytime .....	47
\xpstartref .....	1, 62
\XR@test .....	1
\XR@test@mac .....	1
\XR@test@mac@test .....	1
\Xragged .....	53
\xright@appenditem .....	1
\Xrigidbalance .....	1
\Xsethangindent .....	1
\Xstanza .....	47
\Xstanzaseparator .....	47
\Xstorelineinfo .....	1
\xsublineref .....	1, 62
\Xsublinesep .....	23, 47
\Xsublinesepside .....	23
\Xsymlinenum .....	44
\Xsymlinenumannotation .....	44
\Xtoendnotes .....	31
\Xtwolines .....	46
\Xtwolinesonlyinsamepage .....	46

<code>\Xtxtbeforenotes</code> .....	54
<code>\Xtxtbeforenotesonlyonce</code> .....	54
<code>\Xunvxh</code> .....	<u>1</u>
<code>\Xwidth</code> .....	55
<code>\Xwrapcontent</code> .....	51
<code>\Xwrapendlemma</code> .....	51
<code>\Xwraplemma</code> .....	51
<code>\Xwraplinenumannotation@ref</code> .....	<u>1</u>
<code>\Xwraplinenumannotation@side</code> .....	<u>1</u>
<code>\xxref</code> .....	<u>1</u> , 63

**Z**

<code>\zz@@@</code> .....	<u>1</u>
<code>\zz@@@@</code> .....	<u>1</u>

## Change History

v0.1.0.	
General: First public release	1
v0.2.0.	
\ifl@dmemoir: Added \ifl@dmemoir for memoir class having been used	85
\morenoexpands: Added \l@dtabnoexpands to \no@expands	138
\reledmac@error: Added \eledmac@error and replaced error messages	86
General: Added tabmac code, and extended indexing	1
v0.2.1.	
\@lab: Removed page setting from \@lab	295
\edlabel: Tweaked \edlabel to get correct page numbers	292
\l@ddodoreintrafeet: Renamed \dodoreintrafeet to \l@ddodoreintrafeet	286
\l@ddoxtrafeet: Renamed \doxtrafeet to \l@ddoxtrafeet	284
\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	138
General: Added text about normal labeling	64
Bug fixes and match with mempatch v1.8	1
Major changes to insert code when memoir is loaded	287
v0.2.2.	
\footfudgefiddle: Added \footfudgefiddle	193
\next@line@list@stuff: Added initial write of page number in \line@list@stuff	129
\para@footsetup: Added \footfudgefiddle to \para@footsetup	193
\para@footsetupX: Added \footfudgefiddle to \para@footsetupX	237
General: Improved paragraph footnotes	1
New Dekker example	1
Used \providecommand for \@gobblethree to avoid clash with the amsfonts package	92
v0.3.0.	
\@lab: Replaced \the\line@num by \linenumr@p\line@num in \@lab, and similar for sub-lines	295
\@nl@reg: Added a bunch of code to \@nl for handling \setlinenum	116
\ledlinenum: Added \linenumr@p and \sublinenumr@p to \leftlinenum and \rightlinenum	107
\linenumberlist: Added \linenumberlist mechanism	93
\printendlines: Added \linenumr@p and \sublinenumr@p to \printendlines	256
\printlines: Added \linenumr@p and \sublinenumr@p to \printlines	219
\sublinenumr@p: Added \linenumberstyle and \sublinenumberstyle	106
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	319
\Xarrangement@normal: Added minpage footnote setup to \footnormal	188
\edtext: Added \showlemma to \edtext	139
\l@dfeetendmini: Added \l@dfeetbeginmini, \l@dfeetendmini and all their supporting code	317
\mpnormalfootgroup: Added \mpnormalfootgroup	192

\mpnormalvfootnote: Added \mpnormalvfootnote	190
\showlemma: Added \showlemma	93
General: Added final/draft options	82
Added ledgroup environment	320
Added ledgroupsize environment	320
Added minipage, etc., support	1
v0.4.1.	
\do@Xfeet: Changed \do@Xfeet code for easier extensions	285
\edindex: Let eledmac take advantage of memoir's indexing	325
\print@Xnotes: Added \opXfeet	285
General: Added code for changing \doclearpage	288
Not released. Minor editorial improvements and code tweaks	1
Only change \@footnotetext and \@footnotemark if memoir not used	223
v0.5.0.	
\@footnotetext: Enabled regular \footnote in numbered text	223
\@xympar: Eliminated \marginpar disturbance	309
General: Added left and right sidenotes	309
Added sidenotes, familiar footnotes in numbered text	1
v0.5.1.	
\affixline@num: Changed \affixline@num to cater for sidenotes	172
\l@getsidenote@margin: Added \sidenotemark and \sidenote@margin	309
General: Added moveable sidenote	309
Fixed right line numbers killed in v0.5	1
Only change \hsize in ledgroupsize environment otherwise page number can be in wrong place	320
v0.6.0.	
\@lopR: Added \@pend, \@pendR, \@lopL and \@lopR in anticipation of parallel processing	118
\@nl@reg: Extended \@nl to include the page number	116
\get@thisfootnote: Changed \l@dbfnote and \v1@dbfnote as originals could give incorrect markers in the footnotes	224
\l@the@c@page: Extended \new@line to output page numbers	130
General: Fixed long paragraphs looping	1
Fixed minor typos	1
Prepared for eledpar package	1
v0.7.0.	
\@nl@reg: Added \@nl@reg	116
\@ref@reg: Added \@ref@reg	126
\affixline@num: Added skipnumering to \affixline@num	172
\do@actions@fixedcode: Added \do@actions@fixedcode	168
\do@actions@next: Added number skipping to \do@actions	167
\do@insidelinehook: Added \do@linehook for use in \do@line	164
\endnumbering: Changed \endnumbering for eledpar	97
\fx@l@cks: Added \ch@cksub@l@ck, \ch@ck@l@ck and \fx@l@cks	175
\footsplitskips: Added \footsplitskips for use in many footnote styles	186
\get@linelistfile: Added \get@linelistfile	114
\initnumbering@reg: Added \initnumbering@reg	95
\l@advance@parledgroup@beforenormalnotes: Added \l@dunboxmpfoot containing some common code	319
\l@dcsnotetext@r: Added \l@emptyd@ta	165

\l@dgetline@margin: Added \l@dgetline@margin	103
\l@dgetlock@disp: Added \l@dgetlock@disp	105
\l@dgetsidenote@margin: Added \l@dgetsidenote@margin	309
\l@dnumstartsL: Added \l@dnumstartsL, \ifl@dpairing and \ifpst@rted for/from eledpar	93
\l@drsn@te: Added \l@dlsn@te and \l@drsn@te for use in \do@line	165
\l@dzeropenalties: Added \l@dzeropenalties	159
\ledlinenum: Added \ledlinenum for use by \leftlinenum and \rightlinenum	107
\list@clearing@reg: Added \list@clearing@reg	114
\n@num: Added \n@num	123
\next@line@list@stuff: Deleted \page@start from \line@list@stuff	129
\normalbfnoteX: Removed extraneous space from \normalbfnoteX	229
\pausenumbering@page@num: Changed \resumenumbering for eledpar	98
\setprintendlines: Added \setprintendlines for use by \printendlines	252
\setprintlines: Added \setprintlines for use by \printlines	214
\skipnumbering: Added \skipnumbering and supports	135
\sublinenumincrement: Added \firstlinenum, \linenumincrement, \firstsublinenum and \linenumincrement	104
\sublinenumr@p: Using \linenumrep instead of \linenumr@p	106
Using \sublinenumrep instead of \sublinenumr@p	106
\vnumfootnoteX: Removed extraneous space from \vnumfootnoteX	231
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	86
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@dnumstartsL: Added \ifledRcol and \ifnumberingR for/from eledpar	93
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>\pstart</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.	335
General: New <code>stanzaindentsrepetition</code> counter to repeat stanza indents every $n$ verses.	57
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.	292
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.	172
<code>\ifinserthangingsymbol</code> : New management of <code>hangingsymbol</code> insertion, preventing undesirable insertions.	333
<code>\printlines</code> : Line numbering can be reset at each <code>pstart</code> .	218
General: Line numbering can be reset at each <code>pstart</code> .	101
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.	333
v1.0.0.	
<code>\morenoexpands</code> : Change to be compatible with new features	138
General: <code>\lemma</code> can contain commands.	32
Debug in lineation command	22
New generic commands to customize footnote display.	42, 274
Options <code>nonum</code> and <code>nosep</code> in <code>\Xfootnote</code> .	30
Options of <code>\Xfootnotes</code> .	184
Possibility to have commands in sidenotes.	67
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>	245
<code>\settoggle@series</code> : <code>\settoggle@series</code> switch the global value of the toggle, not only the local value.	274
General: Add <code>\labelpstarttrue</code> .	20
Add <code>\Xnumberonlyfirstintwolines</code>	44
Add <code>\Xpstart</code> and <code>\Xonlypstart</code>	46
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	273
Sectioning commands.	74
v1.2.0.	
<code>\Xprenotes</code> : Debug in familiar footnotes (bug introduced by v1.1).	245
<code>\endquote</code> : Compatibility of <code>\ledchapter</code> with the <i>memoir</i> class.	372

v1.3.0.	
\endquote:	<i>Quotation</i> and quote environment inside numbered sections. . . . . 372
v1.4.0.	
\edtext:	Compatibility of \edtext with the right-to-left direction (with Polyglossia). 139
\ledsetnormalparstuffX:	Direction of footnotes with polyglossia. . . . . 242
\newseries@:	Remembers the language of the lemma, in order to create a correct direction for the footnote separator. . . . . 263
\rbracket:	Switch the right bracket to a left bracket when the lemma is RTL (needs polyglossia or LuaTeX). . . . . 208
General:	Compatibility with LuaTeX of RTL notes. . . . . 1
v1.4.1.	
\affixside@note:	Remove spurious spaces. . . . . 315
\endquote:	New option <i>noquotation</i> . . . . . 372
\get@thisfootnote:	Compatibility of standard footnotes with eledmac when these footnotes contain any commands. . . . . 224
\labelrefsparsesubline:	Fix a bug with \edlabel. . . . . 294
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. . . . . 372
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. . . . . 229
v1.5.0.	
\do@insidelinehook:	Added \do@insidelinehook for use in \do@line . . . . . 164
\edindex:	Compatibility with imakeidx package, and possibility to use multiple index with \edindex. . . . . 325
General:	Correct indexing when the call is made in critical notes. . . . . 321
v1.5.1.	
\managestanza@modulo:	Correct stanzaindentsrepetition counter . . . . . 335
\normalvfootnoteX:	Fix a bug with normal familiar footnotes when mixing RTL and LTR text. . . . . 226
v1.6.0.	
\newverse:	Add \falseverse macro. . . . . 338
v1.6.1.	
\AtStartEveryPstart:	Spurious space in \pstart. . . . . 154
\ifinserthangingsymbol:	Hang verse is now not automatically flush right. . . . . 333
\l@dunhbox@line:	Move the call to \inserthangingsymbol to allow use \hfill inside. . . . . 160

\pend: Spurious space in \pend. . . . .	156
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. . . . .	76
v1.8.0.	
\endquote: Correction of sectioning commands in parallel texts. . . . .	372
\get@index@command: Debug \get@index@command and compatibility with hyperref package. . . . .	324
\newhookcommand@series@reload: Debug \beforenotesX and \maxhnotesX which did not work. . . . .	277
\prevpage@num: Correct \parafootsep when using with ledgroup. . . . .	199
General: Compatibility with parledgroup option of eledpar package. . . . .	1
If imakeidx and hyperref are loaded, adds hyperref in the index. . . . .	321
v1.8.1.	
General: Debug endnotes when more than one series is used (change the position where tools for endnotes are defined). . . . .	247
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. . . . .	321
v1.9.0.	
\l@ddoxtrafeet: Add \fnpos to choice the order of footnotes. . . . .	284
\l@dfteetendmini: Add \mpfnpos to choice the order of footnotes in minipage / ledgroup. . . . .	317
v1.10.0.	
\endquote: Correction of sectioning commands in parallel texts. . . . .	372
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. . . . .	50
v1.11.1.	
General: Correct a bug when a critical note starts with plus or minus. . . . .	1
v1.12.0.	
\@nl@reg: To ensure compatibility with \musixtex, \@l becomes \@l. Consequently, \@l@reg becomes \@nl@reg. . . . .	115

\AtStartEveryPstart: New optional argument for \pstart, to execute code before it.	154
\edindex: Use correctly default index when imakeidx is loaded.	325
\endquote: \ledxxx sectioning commands are deprecated and replaced by \eledxxx commands.	372
\initnumbering@reg: \beginnumbering is defined only on eledmac, not on eledpar.	95
\l@dgetsidenote@margin: \sidenotemargin is now directly defined in eledmac to be able to manage eledpar.	309
\l@disnote: \l@dlsnote, \l@drsnote and \l@dcnote defined only one time, in eledmac, including needs for eledpar case.	311
\l@dnumpstartsL: Add \ifledRcol@ for eledpar	93
\l@dunhbox@line: \do@line is split in more little commands.	161
\newhookcommand@series@reload: Debug \beforenotesX and \maxhnotesX which did not work when called after \footparagraphX.	277
Debug \Xbeforenotes and \Xmaxhnotes which did not work when called after \footparagraph.	277
\pend: New optional argument for \pend, to execute code after it.	156
\stanza: &can have an optional argument: content to be printed after.	338
\Stanza can have an optional argument: content to be printed before.	338
Add \newverse macro, \falseverse deprecated.	338
General: Add \ledinnernote and \ledouternote commands.	67
Add \Xendparagraph and related settings.	56
Add hyperlink to crossref (needs hyperref package).	62
Compatibility with musixtex.	1
Debug eledmac sectioning command after using \resumenumbering.	1
Ensure that imakeidx is loaded <i>before</i> eledmac	322
New hooks: \Xafterrule and \afterruleX	55
New options for ragged-paragraph notes	53
New sectioning commands.	74
Optional arguments for \pstart and \pend.	19
v1.12.1.	
\wrap@edcrossref: Fix spurious spaces.	297
v1.12.2.	
\l@dunhbox@line: Fix a bug with critical notes at the tops of pages (added by v12.0.0)	160
v1.12.3.	
\flag@end: \flag@start and \flag@end are now defined only one time for eledmac and eledpar	131
\flag@start send a error message when a \edtext is done without insert (note)	131
\reledmac@error: Replaced error messages	86
General: Add macros for new messages since v0.7	86
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.	67
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	328
\@ssect: Debug \eledchapter in parallel with memoir	376
\doinsidelinehook: Added \dolinehook and \doinsidelinehook	164

\endnumbering: Allow to mix parallel columns and normal text when using	
\pausenumbering	97
\l@dobbleoptarg: \l@dobblearg becomes \l@dobbeloptyarg	356
\l@restoreforedtext: Debug optional arguments of \Xfootnote in tabular context	356
\pausenumbering@page@num: Debug \resumenumbering	98
v1.12.7.	
\wrap@edcrossref: \wrap@edcrossref is now robust	297
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	131
v1.13.0.	
\newhooktoggle@series@reload: Add \newhookcommand@toggle@reload	277
\para@footsetupX: In \para@footsetupX, use \columnwidth instead of \hsize	237
\settoggle@series: \settoggle@series can take an optional arguments to reload	
series setup.	274
General: Add \Xnoteswidthliketwocolumns and \noteswidthliketwocolumnsX	56
Added widthliketwocolumns option	82
v1.13.1.	
\ifat@every@pstart@star@: Add \l@dzeropenalties in \pstart	154
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	93
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	325
General: Debug spurious characters before endnotes.	247
Delete previous override of \l@d@wrindexhyp at the beginning of a document	
when hyperref is not loaded.	329
Move gobbling command	92
Provide \@gobblefour	92
v1.14.1.	
\@ssect: Debug sectioning commands when using both handout and hyperref	
package.	378
v1.14.2.	
\@ssect: Debug \edtext after starred sectioning commands when using memoir class.	376
v1.15.0.	
\@edtext@level: New boolean \if@edtext@.	139
\arrangementX@paragraph: Correct bug with paragraphed familiar footnotes setting.	237
\endsub: Restore subline feature (disabled by mistake in v1.8.0).	133
\if@lemmacommand@: New boolean \iflemmacommand@.	145
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.	19
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. . . . .
	129
v1.16.0.	
General:	\edtext is now defined only in eledmac, not in eledpar. Debug wrong numbering when using \sameword + eledpar + \tag command. . . . .
	139
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. . . . .
	298
v1.17.0.	
\edtext:	Error message when calling \edtext outside of a numbered paragraph. . . .
	139
v1.18.0.	
\@edindex@hyperref:	Fix spurious space with \edindex when using imakeidx/indextools + hyperref. . . . .
	328
\edlabel:	\edlabel is now defined only one time for both eledmac and eledpar . . . .
	292
\l@d@section:	Option parapparatus works for endnotes. . . . .
	248
\l@dnumpstartsL:	Add \ifl@dprintingpages and \@dprintingcolumns for eledpar . . . . .
	93
\print@line:	Compatibility with Lua $\TeX$ RTL languages. . . . .
	161
\printlinefootnote:	Code refactoring in \printlinefootnote: the printing of the numbers are factorized in \printlinefootnotearea . . . . .
	210
\printpstart:	Debug \Xpstart with parallel pages and columns (eledpar) . . . . .
	208
General:	Add \Xpstarteverytime . . . . .
	46
Compatibility with Lua $\TeX$ RTL languages.	. . . . .
	1
Debug \Xonlypstart when using \Xnumberonlyfirstinline and the current line number differs from the previous.	. . . . .
	46
v1.19.0.	
\footplitskips:	\footplitskips doesn't set \floatingpenalty to \@MM when processing parallel pages. . . . .
	186
\xxref:	\xxref works also with right side numbers, when \@Rlineflag is not empty. . . .
	300
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@d@Xmorethantwolines and \ifl@d@Xmorethantwolines to \printlines . . . . .
	219
\stanza:	& and \& can be preceded by spaces. . . . .
	338
\xxref:	Debug \xxref when not loading eledpar (Fix a bug added in 1.19.0). . . . .
	300
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	328
$\backslash$ l@d@section: $\backslash$ endnotes take five arguments.	248
$\backslash$ ledinnotemark: Add $\backslash$ ledinnotemark.	325
$\backslash$ ledsetnormalparstuffX: $\backslash$ ledsetnormalparstuff is deprecated and becomes $\backslash$ ledsetnormalparstuffX and $\backslash$ Xledsetnormalparstuff.	242
$\backslash$ n@num: $\backslash$ n@num@ref deleted	123
$\backslash$ n@num defined only one time for both Eledmac and Eledpar	123
$\backslash$ newhookcommand@series: $\backslash$ newhookcommand@series can take an optional argument.	276
$\backslash$ newhooktoggle@series: $\backslash$ newhooktoggle@series can take an optional argument.	276
$\backslash$ print@footnotexrule: Code refactoring: the spaces after the footnote rules are directly managed in $\backslash$ print@Xfootnoterule and $\backslash$ print@footnotexrule	244
$\backslash$ seriesatend: Fix spurious space in $\backslash$ seriesatend	273
$\backslash$ skipnumbering: $\backslash$ skipnumbering defined only one time for both Eledmac and Eledpar.	135
Correct $\backslash$ skipnumbering for stanza.	135
Delete $\backslash$ skipnumbering@reg.	135
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	273
Add $\backslash$ applabel and related	64
Add $\backslash$ beforenotesX and $\backslash$ Xbeforenotes features for notes set in two and three column.	1
Add $\backslash$ hidenumbering	24
Add $\backslash$ Xcolalign and $\backslash$ colalignX hooks	52
Add $\backslash$ Xendtwolines, $\backslash$ Xendmorethantwolines, $\backslash$ Xendtwolinesbutnotmore and $\backslash$ Xendtwolinesonlyinsamepage.	46
Add $\backslash$ Xparindent and $\backslash$ hangindentX	51
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 $\backslash$ footthreecolX.	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>	92
v1.22.0.	
<code>\ledinnote</code> : <code>\ledinnote</code> takes a first optional argument, which is the label for hyperlinks.	325
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> .	199
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> .	139
<code>\Serefwithpage</code> : Debug <code>\Xendtwolines</code> , <code>\Xendmorethantwolines</code> , <code>\Xendtwolinesbutnotmore</code> and <code>\Xendtwolinesonlyinsamepage</code> when using <code>\apprefwithpage</code> .	305
<code>\lemma</code> : Fix spurious space after <code>\lemma</code> command	144
<code>\newseries@</code> : Prevent spurious spaces when <code>\Afootnote</code> and similar commands are followed by spaces (bug added on 1.0.0).	263
<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.	150
General: Add <code>\Xboxlinenumalign</code> and <code>\Xendboxlinenumalign</code> .	48
Add <code>\Xboxstartlinenum</code> , <code>\Xendboxstartlinenum</code> , <code>\Xboxendlinenum</code> , <code>\Xendboxendlinenum</code> .	48
Allow use of <code>\sameword</code> with <code>inputenc</code> managing of UTF-8.	1
Compatibility between <code>nofamiliar/nocriticals</code> option and <code>minipage/ledgroup</code> .	1
Error message when using <code>\beginnumbering ... \endnumbering</code> without <code>\pstart</code> .	1
Fix 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. . . . .	319
<code>\@xympar</code> : Patching <code>\@xympar</code> instead of redefining it . . . . .	309
<code>\endminipage</code> : Patch <code>\endminipage</code> instead of redefining it. . . . .	319
<code>\initnumbering@quote</code> : <code>\initnumbering@sectcmd</code> becomes <code>\initnumbering@quote</code> . . . . .	372
<code>\l@advance@parledgroup@beforenormalnotes</code> : Some conde of <code>\l@dumboxmpfoot</code> moved to <code>\l@advance@parledgroup@beforenormalnotes</code> . . . . .	319
<code>\newseries@</code> : One endnotes file by series. . . . .	269
General: <code>\@makecol</code> , <code>\@reinserts</code> and <code>\@doclearpage</code> are patched instead of begin redefined . . . . .	287
<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> . . . . .	287
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 $\TeX$ ’s output macros are now patched and not override. . . . .	1
Package’s name becomes <code>reledmac</code> . . . . .	1
Patch <code>\@footnotemark</code> instead of redefine it . . . . .	223
Suppress indexing command specific to <code>memoir</code> . . . . .	325
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. . . . .	242
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>\Xendsymmlinenum</code> , <code>\Xendbeforenumber</code> , <code>\Xendafternumber</code> , <code>\Xendbeforesymmlinenum</code> , <code>\Xendaftersymmlinenum</code> , <code>\Xendboxsymmlinenum</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. . . . .	184
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>\Xlinerangeseparator</code> and <code>\Xendlinerangeseparator</code> . . . . .	46

Option <code>linrangesep</code> for critical footnotes and endnotes. . . . .	46
v2.4.1.	
General: Fix a bug with <code>\appref</code> and <code>\apprefwithpage</code> (introduced in v2.4.0). . . . .	1
Fix a bug with tabular environments when using <code>babel</code> or <code>polyglossia</code> languages that override $\TeX$ <code>\roman</code> command, like Greek language. . . . .	1
Fix a bug with tabular environments when using <code>babel</code> or <code>polyglossia</code> languages that override $\TeX$ <code>\roman</code> command, like Greek. . . . .	1
v2.5.0.	
<code>\SErefwithpage</code> : Debug <code>\setapprefprefixsingle</code> . . . . .	304
<code>\edlabel</code> : Fix a bug when calling <code>\edlabel</code> in a footnotes of the right side . . . . .	292
<code>\l@d@section</code> : <code>\endnotes</code> take six arguments. . . . .	248
<code>\printlines</code> : <code>\printlines</code> takes an eighth argument: the line flag . . . . .	218
<code>\xlineref</code> : <code>\xlineref</code> does not include anymore the side flag. Use <code>\xflagref</code> to get it. Not that <code>\edlineref</code> still contains the flag. . . . .	298
General: <code>\apprefwithpage</code> and <code>\appref</code> print double quotation mark when the label was not defined. . . . .	1
<code>\apprefwithpage</code> and <code>\appref</code> work with right side crossref. . . . .	1
<code>\apprefwithpage</code> works also when <code>noend</code> option is enabled. . . . .	1
<code>\appref</code> and <code>\apprefwithpage</code> can take <code>linrangesep</code> optional argument. . . . .	1
<code>\edlabel</code> works now in <code>\Xfootnote</code> . . . . .	1
<code>\lemma</code> can be used even when the <code>nocritical</code> is enabled. . . . .	1
Compatibility with new hook and tools of <code>reledpar</code> 2.6.0. . . . .	1
Fix spurious vertical space in <code>astanza</code> environment ( <code>reledpar</code> ) . . . . .	1
Log now states ‘There were undefined references’ when using wrong references in <code>\edlineref</code> or <code>edpageref</code> . . . . .	1
New hooks to customize page and line number appearance in endnotes. . . . .	1
New hooks: <code>\Xhookgroup</code> and <code>\bhookgroupX</code> . . . . .	1
New tools to easily make cross-reference to a passage defined by a start and an end line	64
v2.6.0.	
General: Adds compatibility with <code>innnote</code> and <code>notenumber</code> options of <code>indextools</code> package. . . . .	1
Fix a bug with footnote counter in <code>ledgroup</code> (added in v2.5.0). . . . .	1
Fix bug, introduced in v2.5.0, with footnote numbering in parallel typesetting when using <code>perpage</code> package. . . . .	1
v2.7.0.	
<code>\@k</code> : <code>\rigidbalance</code> is split in <code>\Xrigidbalance</code> and <code>\rigidbalanceX</code> . . . . .	200
<code>\l@d@section</code> : <code>\endnotes</code> take seven arguments. . . . .	248
General: Add dash as default page range separator for <code>\SEonlypage</code> . . . . .	1
Debug <code>\SErefonlypage</code> when referring to only one page. . . . .	1
Delete parenthesis after <code>\SErefonlypage</code> . . . . .	1
Fix (again) bugs with footnote numbering in parallel typesetting while using <code>ledgroup</code> environments (bug added in v2.5.0). . . . .	1
Fix a bug with <code>\SErefwithpage</code> . . . . .	1
Fix bugs in compatibility with <code>innnote</code> and <code>notenumber</code> options of <code>indextools</code> package, when indexing outside of a <code>ledgroup</code> . . . . .	1
New commands to make glossaries connected to page and linenumber with the <code>glossaries</code> package . . . . .	1
New hooks: <code>\Xhsize</code> and <code>\hsizeX</code> . . . . .	55
New hooks: <code>\Xlemmafont</code> and <code>\Xendlemmafont</code> . . . . .	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 <code>related</code> , <code>\appref</code> and <code>related</code> . . . . .	1
When using <code>hyperref</code> package, add links from critical footnotes and critical endnotes to the line of text they refers . . . . .	1
v2.7.1.	
General: Debug <code>\Xhookgroup</code> hooks executed on columnar footnotes (moved to a larger group, to take effect). . . . .	1
v2.7.2.	
General: <code>\Xhsize</code> and <code>\hsizeX</code> become <code>\Xwidth</code> and <code>\widthX</code> . . . . .	55
Fix problem of hyphenation when using <code>hyperref</code> package (added in v2.7.0). . . . .	1
v2.8.0.	
<code>\l@d@section</code> : <code>\Xendhangindent</code> and <code>\Xendafternote</code> can take values which are relative to the font size of the endnote. . . . .	248
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</code> ... <code>\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</code> ... <code>\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. . . . .	208
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. . . . .	51
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 . . . . .	168
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>	287
Add <code>\Xendpagenumberonlyfirst</code> , <code>\Xendpagenumberonlyfirstifsingle</code> , <code>\Xendpagenumberonlyfirstintwo</code> , <code>\Xendinplaceofpagenumber</code> and <code>\Xendsympagenum</code> hooks.	1
Add <code>\Xpagelinesep</code> hook.	47
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.	287
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>\doennotes</code> when referring to right side line in parallel typesetting. . . . .	1
Take into account <code>\linenumberstyle</code> when using <code>\edlineref</code> . . . . .	1
v2.16.7.	
General: Fix a bug with <code>\msdata</code> when using multiple <code>\beginnumbering</code> ... <code>\endnumbering</code> . . . . .	1
Fix a bug with <code>\numberpstarttrue</code> when using multiple <code>\beginnumbering</code> ... <code>\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.	129
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 L <sup>A</sup> T <sub>E</sub> X 2018/04/01 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>\Xnoidenticallylinenumannotation</code> and <code>\Xendnoidenticallylinenumannotation</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 not 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