

# 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 . . . . .	12
1.2.1 edmac . . . . .	12
1.2.2 ledmac . . . . .	14
1.2.3 eledmac . . . . .	14

---

\*This file (**reledmac.dtx**) has version number v2.27.1, last revised 2018/10/12.

<sup>†</sup>maieul at maieul dot net

1.2.4 <code>reledmac</code> . . . . .	14
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>15</b>
4.1 Specific features . . . . .	16
4.2 Optimizing package performance . . . . .	16
<b>5 Text lines and paragraphs numbering</b>	<b>17</b>
5.1 Text lines numbering . . . . .	17
5.2 Paragraphs . . . . .	17
5.2.1 Basics . . . . .	17
5.2.2 Automatically producing <code>\pstart ... \pend</code> . . . . .	18
5.2.3 Content before specific <code>\pstart</code> and after specific <code>\pend</code> . . . . .	18
5.2.4 Content before every <code>\pstart</code> and after every <code>\pend</code> . . . . .	19
5.2.5 Numbering paragraphs ( <code>\pstart</code> ) . . . . .	19
5.2.6 Languages written in Right to Left . . . . .	20
5.2.7 Memory limits . . . . .	20
5.3 Lineation commands . . . . .	20
5.3.1 Disabling lineation . . . . .	20
5.3.2 Setting lineation start and step . . . . .	21
5.3.3 Setting lineation reset . . . . .	21
5.3.4 Setting line number margin . . . . .	21
5.3.5 Other settings . . . . .	22
5.4 Changing the line numbers . . . . .	22
5.4.1 Sublineation . . . . .	22
5.4.2 Locking lineation . . . . .	22
5.4.3 Setting and changing line number . . . . .	22
5.4.4 Line number style . . . . .	23
5.4.5 Skipping and hiding number . . . . .	23
5.5 Executing code at each line . . . . .	24
5.6 Executing code at a specific line . . . . .	24
<b>6 Apparatus commands</b>	<b>24</b>
6.1 Terminology . . . . .	24
6.2 Critical notes . . . . .	25
6.2.1 The lemma . . . . .	25
6.2.2 Footnotes . . . . .	25
6.2.3 Endnotes . . . . .	26
6.2.4 Paragraph in critical apparatus . . . . .	27
6.2.5 Change lemma and line number . . . . .	28
6.2.6 Changing the names of commands for critical apparatus . . . . .	29
6.3 Disambiguation of identical words in the apparatus . . . . .	29

6.3.1 Basic use . . . . .	29
6.3.2 Case setting . . . . .	29
6.3.3 Notes about input encoding with UTF-8 processor . . . . .	30
6.3.4 Use with <code>\lemma</code> command . . . . .	30
6.3.5 Sameword for a group of words . . . . .	32
6.3.6 Customizing . . . . .	33
6.3.7 Problems with some macros . . . . .	33
6.3.8 Automatic sameword annotation . . . . .	34
6.4 Apparatus of manuscripts . . . . .	34
6.4.1 Marking sections of text . . . . .	34
6.4.2 Layout of the apparatus of manuscripts . . . . .	35
6.4.3 Settings . . . . .	35
6.5 Familiar notes . . . . .	36
6.5.1 Basic use . . . . .	36
6.5.2 Customizing mark . . . . .	36
6.5.3 Separator for multiple footnotes . . . . .	36
6.6 Printing the footnote mark without printing the footnote text . . . . .	36
6.7 Changing series . . . . .	37
6.7.1 Create a new series . . . . .	37
6.7.2 Delete series . . . . .	37
6.7.3 Series order . . . . .	37
6.8 Position of critical and familiar footnotes . . . . .	37
<b>7 Apparatus customization</b> . . . . .	<b>38</b>
7.1 Notes arrangement in a series . . . . .	39
7.2 Control line number printing . . . . .	39
7.2.1 Print line number only at first time . . . . .	39
7.2.2 Print page number only at first time . . . . .	40
7.2.3 Arbitrary text before line number . . . . .	41
7.2.4 Separator for line range . . . . .	41
7.2.5 Abbreviate line range . . . . .	41
7.2.6 Disable line number . . . . .	42
7.2.7 Printing pstart number . . . . .	42
7.2.8 Printing stanza number . . . . .	42
7.2.9 Separator between line and subline numbers . . . . .	42
7.2.10 Separator between page and line numbers . . . . .	43
7.2.11 Space around number . . . . .	43
7.2.12 Space around line symbol . . . . .	43
7.2.13 Space in place of number . . . . .	43
7.2.14 Boxing line number and line symbol . . . . .	43
7.3 Arbitrary code around line number . . . . .	44
7.4 Separator between the lemma and the note . . . . .	45
7.4.1 For footnotes . . . . .	45
7.4.2 For endnotes . . . . .	45
7.5 Font style . . . . .	45
7.5.1 For line number . . . . .	45

7.5.2 For the lemma . . . . .	46
7.5.3 For all notes . . . . .	46
7.6 Wrapping notes . . . . .	46
7.6.1 Wrapping lemmas . . . . .	46
7.6.2 Wrapping contents . . . . .	47
7.7 Indent of notes content . . . . .	47
7.8 Arbitrary code at the beginning of notes . . . . .	47
7.9 Arbitrary code before inserting note . . . . .	47
7.10 Options for footnotes in columns . . . . .	48
7.10.1 Alignment . . . . .	48
7.10.2 Size of the columns . . . . .	48
7.11 Options for paragraphed footnotes and notes grouped by line . . . . .	48
7.11.1 Mark separation of notes . . . . .	48
7.11.2 Ragged text . . . . .	49
7.12 Options for block of notes . . . . .	49
7.12.1 Grouping notes by line . . . . .	49
7.12.2 Text before notes . . . . .	49
7.12.3 Code before notes . . . . .	49
7.12.4 Spacing . . . . .	50
7.12.5 Rule . . . . .	50
7.12.6 Maximum height . . . . .	50
7.12.7 Width . . . . .	51
7.13 Footnotes and the reledpar columns . . . . .	51
7.14 Line number annotation . . . . .	51
7.15 Endnotes in one paragraph . . . . .	51
<b>8 Fonts</b>	<b>51</b>
<b>9 Verse</b>	<b>52</b>
9.1 Basic . . . . .	52
9.2 Define stanza indents . . . . .	52
9.3 Repeating stanza indents . . . . .	53
9.4 Manual stanza indent . . . . .	53
9.5 Stanza breaking . . . . .	54
9.6 Hanging symbol . . . . .	54
9.7 Long verse and page break . . . . .	54
9.8 Content before/after verses . . . . .	55
9.9 Numbering stanza . . . . .	55
9.10 Adding annotations to line numbers . . . . .	56
9.10.1 Modifying annotation associated with a specific note . . . . .	57
9.10.2 Changing the position of the annotation . . . . .	57
9.10.3 Changing the macro in which annotations are wrapped . . . . .	58
9.11 Various tools . . . . .	59
9.12 Notes on empty lines . . . . .	59
<b>10 Grouping</b>	<b>59</b>

<b>11 Cross referencing</b>	<b>60</b>
11.1 Basic use . . . . .	60
11.2 Cross-referencing to a critical note . . . . .	60
11.3 Cross-referencing which return a number in any case . . . . .	60
11.3.1 Cross-referencing in order to define line number of a critical note . . . . .	61
11.4 Not automatic cross-referencing . . . . .	61
11.5 Normal $\LaTeX$ cross-referencing . . . . .	62
11.6 References to start and end lines . . . . .	62
11.6.1 Reference to main text lines . . . . .	62
11.6.2 References to lines that are commented on in the apparatus . . . . .	62
11.6.3 Settings . . . . .	62
11.7 Compatibility with <code>xr</code> package . . . . .	64
<b>12 Sidenotes</b>	<b>65</b>
12.1 Basics . . . . .	65
12.2 Setting . . . . .	65
12.2.1 Width . . . . .	65
12.2.2 Vertical position . . . . .	65
12.2.3 Distance to the main text . . . . .	65
12.2.4 Font . . . . .	65
12.2.5 Separator between notes . . . . .	66
<b>13 Indexing</b>	<b>66</b>
13.1 Basics . . . . .	66
13.2 Use with <code>imakeidx</code> or <code>indextools</code> . . . . .	66
13.3 Referring to critical notes . . . . .	66
13.4 Separator between page and line numbers . . . . .	67
13.5 Using <code>xindy</code> . . . . .	67
13.6 Advanced setting . . . . .	68
<b>14 Glossary</b>	<b>68</b>
14.1 Preamble setting . . . . .	68
14.2 Commands . . . . .	68
<b>15 Tabular material</b>	<b>69</b>
<b>16 Sectioning commands</b>	<b>72</b>
16.1 Sectioning commands without line numbers or critical notes . . . . .	72
16.2 Sectioning commands with line numbering and critical notes . . . . .	72
16.3 Optimization . . . . .	73
<b>17 Quotation environments</b>	<b>73</b>
<b>18 Page breaks</b>	<b>73</b>
18.1 Control page breaking . . . . .	73
18.2 Prevent page break in a long verses . . . . .	74

<b>19 Miscellaneous</b>	<b>74</b>
19.1 Known and suspected limitations	75
19.1.1 Non-standard geometry	75
19.1.2 floatrow package compatibility	75
19.1.3 ‘No room for a new’	75
19.1.4 Marginal notes	75
19.1.5 Paragraph shape	76
19.1.6 Paragraphed footnotes	76
19.1.7 Use with other packages	76
19.2 Parallel typesetting	77
<b>I Implementation overview</b>	<b>78</b>
<b>II Preliminaries</b>	<b>78</b>
II.1 Links with original edmac	78
II.2 Package declaration	78
II.3 Package options	79
II.3.1 Options of reledpar	79
II.3.2 Options of reledmac	79
II.4 Loading packages	81
II.5 Compatibility with LuaTeX	81
II.6 Boolean flags	82
II.7 Messages	83
II.8 Gobbling	89
II.9 Wrapping content	89
II.10 Miscellaneous commands	90
II.11 Prepare reledpar	90
II.12 Booleans provided by other optional packages which are required in any case	91
<b>III Sectioning commands</b>	<b>91</b>
<b>IV List macros</b>	<b>97</b>
<b>V Line counting</b>	<b>98</b>
V.1 Choosing the system of lineation	98
V.2 Line number margin	100
V.3 Line number initialization and increment	101
V.4 Line number locking	102
V.5 Line number style	103
V.6 Line number printing	103
V.7 Line number counters and lists	105
V.8 Line number locking counter	106
V.9 Line number associated to lemma	106
V.10 Reading the line-list file	109
V.11 Commands within the line-list file	112

V.12 Writing to the line-list file . . . . .	125
<b>VI Marking text for notes</b>	<b>133</b>
VI.1 <code>\edtext</code> itself . . . . .	134
VI.2 Substitute lemma . . . . .	141
VI.3 Substitute line numbers . . . . .	142
VI.4 Lemma disambiguation . . . . .	143
<b>VII Paragraph decomposition and reassembly</b>	<b>150</b>
VII.1 Boxes, counters, <code>\pstart</code> and <code>\pend</code> . . . . .	151
VII.2 Processing one line . . . . .	157
VII.2.1 General process . . . . .	157
VII.2.2 Process for “normal” line . . . . .	158
VII.2.3 Process for line containing <code>\eledsection</code> command . . . . .	160
VII.2.4 Hooks . . . . .	161
VII.2.5 Sidenotes and marginal line number initialization . . . . .	162
<b>VIII Line and page number computation</b>	<b>163</b>
VIII.1 Continuous line numbering between parallel typesetting and normal typesetting . . . . .	166
<b>IX Line number annotation</b>	<b>167</b>
<b>X Line number printing</b>	<b>169</b>
<b>XI Pstart number printing inside</b>	<b>173</b>
<b>XII Restoring footnotes and penalties</b>	<b>174</b>
XII.1 Add insertions to the vertical list . . . . .	174
XII.2 Penalties . . . . .	176
XII.3 Printing leftover notes . . . . .	177
XII.4 Text before notes . . . . .	178
<b>XIII Critical footnotes</b>	<b>180</b>
XIII.1 Fonts . . . . .	180
XIII.2 Individual note options . . . . .	180
XIII.3 Notes language . . . . .	181
XIII.4 General survey of the way we manage notes . . . . .	182
XIII.5 General setup . . . . .	182
XIII.6 Footnotes arrangement . . . . .	183
XIII.6.1 User level macro . . . . .	183
XIII.6.2 Normal footnote . . . . .	184
XIII.6.3 Paragraphed footnotes . . . . .	189
XIII.6.4 Columnar footnotes . . . . .	196
XIII.7 Critical notes presentation . . . . .	203
XIII.7.1 Font tools . . . . .	203
XIII.7.2 Pstart number in footnote . . . . .	204

XIII.7.3 Lemma printing . . . . .	204
XIII.7.4 Line number printing . . . . .	205
XIII.7.5 Footnote grouped by line . . . . .	215
<b>XIV Familiar footnotes</b>	<b>216</b>
XIV.1 Adjacent footnotes . . . . .	216
XIV.2 Regular footnotes for numbered texts . . . . .	218
XIV.3 Footnote formats . . . . .	220
XIV.4 Footnote arrangement . . . . .	220
XIV.4.1 User level macro . . . . .	220
XIV.4.2 Normal footnotes . . . . .	221
XIV.4.3 Two columns footnotes . . . . .	226
XIV.4.4 Three columns footnotes . . . . .	229
XIV.4.5 Paragraphed footnotes . . . . .	231
XIV.5 Wrapping footnote marks in hyperlink . . . . .	235
<b>XV Code common to both critical and familiar footnote in normal arrangement</b>	<b>236</b>
<b>XVI Footnotes' width for two columns</b>	<b>237</b>
<b>XVII Footnotes' order</b>	<b>238</b>
<b>XVIII Footnotes' rule</b>	<b>238</b>
<b>XIX Specific skip for first series of footnotes</b>	<b>239</b>
XIX.1 Overview . . . . .	239
XIX.2 User level command . . . . .	239
XIX.3 Internal commands . . . . .	240
<b>XX Endnotes</b>	<b>241</b>
XX.1 Internal commands . . . . .	241
XX.2 User level commands . . . . .	245
XX.2.1 Inserting contents to endnotes . . . . .	245
XX.2.2 Printing endnotes . . . . .	246
<b>XXI Generate series of notes</b>	<b>253</b>
XXI.1 Test if series is still existing . . . . .	254
XXI.2 Init specific to <code>reledpar</code> . . . . .	254
XXI.3 For critical footnotes . . . . .	254
XXI.3.1 Options . . . . .	255
XXI.3.2 Create inserts, needed to add notes in foot . . . . .	256
XXI.3.3 Create commands for critical apparatus, <code>\Afootnote</code> , <code>\Bfootnote</code> etc. . . . .	256
XXI.3.4 Set standard display . . . . .	259
XXI.4 For familiar footnotes . . . . .	259
XXI.4.1 Options . . . . .	259



XXI.4.2 Create tools for familiar footnotes ( <code>\footnoteX</code> ) . . . . .	260
XXI.5 The endnotes . . . . .	262
XXI.5.1 The auxiliary file . . . . .	262
XXI.5.2 The main macro . . . . .	263
XXI.5.3 Tools . . . . .	263
XXI.5.4 Internal commands . . . . .	264
XXI.5.5 The options . . . . .	264
XXI.6 Init standards series (A,B,C,D,E) . . . . .	266
<b>XXII Setting series display</b>	<b>266</b>
XXII.1 Change series order . . . . .	266
XXII.2 Test series order . . . . .	266
XXII.2.1 Get the first series . . . . .	267
XXII.3 Series setting . . . . .	267
XXII.3.1 General way of working . . . . .	267
XXII.3.2 Tools to set options . . . . .	267
XXII.3.3 Tools to generate options commands . . . . .	269
XXII.3.4 Options for critical notes . . . . .	270
XXII.3.5 Options for familiar notes . . . . .	272
XXII.3.6 Options for endnotes . . . . .	273
XXII.4 Hooks for a particular footnote . . . . .	275
XXII.5 Alias . . . . .	275
<b>XXIII Output routine</b>	<b>276</b>
XXIII.1 Extra footnotes output . . . . .	276
XXIII.2 Patching standard output's commands . . . . .	279
<b>XXIV Page numbering in parallel typesetting</b>	<b>282</b>
<b>XXV Cross referencing</b>	<b>283</b>
XXV.1 Compatibility with <code>xref</code> . . . . .	299
<b>XXVI Sidenotes</b>	<b>300</b>
<b>XXVII Minipages and such</b>	<b>308</b>
<b>XXVIII Indexing</b>	<b>313</b>
XXVIII.1 Looking on package order . . . . .	313
XXVIII.2 Auxiliary macros for <code>\edindex</code> . . . . .	314
XXVIII.3 Code specific to <code>\edindexin</code> critical footnotes . . . . .	315
XXVIII.4 Analysis of command in indexed text . . . . .	316
XXVIII.5 Code for the formatted index . . . . .	316
XXVIII.6 Main code . . . . .	317
XXVIII.7 Hyperlink . . . . .	319
XXVIII.8 'innote' and 'notenumber' option of <code>indextols</code> package . . . . .	321
<b>XXIX Glossaries</b>	<b>322</b>

<b>XXX Verse</b>	<b>324</b>
XXX.1 Hanging symbol management . . . . .	324
XXX.2 Using & character . . . . .	325
XXX.3 Code category setting . . . . .	325
XXX.4 Stanza count and indent . . . . .	325
XXX.5 Numbering stanza . . . . .	327
XXX.6 Stanza number in note . . . . .	328
XXX.7 Main work . . . . .	328
XXX.8 Restore catcode and penalties . . . . .	331
<b>XXXI Apparatus of Manuscripts</b>	<b>332</b>
XXXI.1 User level macro . . . . .	332
XXXI.2 Setting macro . . . . .	333
XXXI.3 Counters and lists . . . . .	334
XXXI.4 Auxiliary file macros . . . . .	334
XXXI.5 Action macro . . . . .	335
XXXI.6 Inserting footnote . . . . .	340
XXXI.7 Other . . . . .	340
<b>XXXII Arrays and tables</b>	<b>341</b>
XXXII.1 Preamble: macro as environment . . . . .	341
XXXII.2 Tabular environments . . . . .	344
XXXII.2.1 Disabling and restoring commands . . . . .	344
XXXII.2.2 Counters, boxes and lengths . . . . .	348
XXXII.2.3 Tabular typesetting . . . . .	352
XXXII.2.4 Environments . . . . .	363
<b>XXXIII Quotation's commands</b>	<b>363</b>
<b>XXXIV Section's title commands</b>	<b>365</b>
XXXIV.1 Commands to disable some feature . . . . .	365
XXXIV.2 General overview . . . . .	365
XXXIV.3 \beforeeledchapter command . . . . .	366
XXXIV.4 Auxiliary commands . . . . .	366
XXXIV.5 Patching standard commands . . . . .	367
XXXIV.6 Main code of \eledxxx commands . . . . .	372
XXXIV.7 Macros written in the auxiliary file . . . . .	375
<b>XXXV Page breaking or no page breaking depending of specific lines</b>	<b>377</b>
<b>XXXVI Long verse: prevents being separated by a page break</b>	<b>379</b>
<b>XXXVII Tools for hyperref package</b>	<b>379</b>
<b>XXXVIII Compatibility with eledmac</b>	<b>380</b>

<b>Appendix A Things to do when changing versions</b>	<b>383</b>
A.1 Migrating from edmac to ledmac . . . . .	383
A.2 Migration from ledmac to eledmac . . . . .	384
A.3 Migration to eledmac 1.5.1 . . . . .	385
A.4 Migration to eledmac 1.12.0 . . . . .	385
A.5 Migration to eledmac 17.1 . . . . .	386
A.6 Migration to eledmac 1.21.0 . . . . .	386
A.6.1 \Xledsetnormalparstuff and \ledsetnormalparstuffX . . .	386
A.6.2 Endnotes . . . . .	386
A.7 Migration to eledmac 1.22.0 . . . . .	386
A.8 Migration to eledmac 1.23.0 . . . . .	386
A.9 Migration from eledmac to reledmac . . . . .	387
A.9.1 Risk of ‘no room for a new’ . . . . .	387
A.9.2 Multiple indices with memoir . . . . .	387
A.9.3 Deprecated commands and options . . . . .	387
A.9.4 \renewcommand replaced by command . . . . .	388
A.9.5 Commands the names of which have been changed . . . . .	388
A.9.6 Endnotes . . . . .	390
A.9.7 Z Series . . . . .	390
A.9.8 Internal commands . . . . .	390
A.10 Migration to reledmac 2.1.0 . . . . .	390
A.11 Migration to reledmac 2.1.3 . . . . .	390
A.12 Migration to reledmac 2.3.0 . . . . .	390
A.13 Migration to reledmac 2.4.0 . . . . .	391
A.14 Migration to reledmac 2.5.0 . . . . .	391
A.15 Migration to reledmac 2.7.0 . . . . .	391
A.16 Migration to reledmac 2.7.2 . . . . .	391
A.17 Migration to reledmac 2.8.0 . . . . .	391
A.18 Migration to reledmac 2.13.1 . . . . .	391
A.19 Migration to reledmac 2.18.0 . . . . .	392
A.20 Migration to reledmac 2.21.0 . . . . .	392
A.21 Migration to reledmac 2.24.0 . . . . .	392
A.22 Migration to reledmac 2.26.0 . . . . .	392
A.23 Migration to reledmac 2.27.1 . . . . .	392
<b>References</b>	<b>393</b>
<b>Index</b>	<b>393</b>
<b>Change History</b>	<b>444</b>

# 1 Introduction

## 1.1 Aim of the package

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

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

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

Apart from `reledmac` there are other  $\text{\LaTeX}$  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 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 Hun-

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

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

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

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

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

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

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

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

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

gary 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` (L<sup>A</sup>T<sub>E</sub>X `edmac`).

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

### 1.2.3 eledmac

Important changes were put in version 1.0, to make `ledmac` more easily extensible (see 7 p. 38). 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. 384.

### 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. 387.

<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.

### 1.3 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. 50.

A file may mix *numbered* and *unnumbered* text.

Numbered text is printed with marginal line numbers and can include footnotes and endnotes that are referenced to those line numbers: this is how you will want to print the text that you are editing.

Unnumbered text is not printed with line numbers, and you can't use `reledmac`'s note commands with it: this is appropriate for introductions and other material added by the editor around the edited text.

## 4 Options

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

## 4.1 Specific features

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

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

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

**eledmac-compat** help to migrate from `eledmac` to `reledmac` (see A.9.5 p. 388).

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

**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. 73).

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

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

**xindy** and `xindy+hyperref` select `xindy` as the index processor (13.5 p. 67).

## 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. 72).

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

---

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

Note that a `\noindent` will be automatically added before this argument, and, consequently, a `\parskip` will be inserted. You can use a second optional argument in brackets to 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. 52) or `reledpar` (see 19.2 p. 77).

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

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

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

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

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

```
\AtStartEveryPstart{*}
```

Instead of manually doing

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

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

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

```
\setcounter{pstart}{value}
```

On each `\beginnumbering` the numbering restarts.

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

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

### 5.2.6 Languages written in Right to Left

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

### 5.2.7 Memory limits

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

`\pausenumbering`  
`\resumenumbering` reledmac stores a lot of information about line numbers and footnotes in memory as it goes through a numbered section. But at the end of such a section, it empties its memory out, so to speak. If your text has a very long numbered section it is possible that your  $\TeX$  may reach its memory limit. There are two solutions to this.

The first solution is to get a larger  $\TeX$  with increased memory.

The second solution is to split your long section into several smaller ones. The trouble with this is that your line numbering will start again at zero with each new section. To avoid this problem, we provide `\pausenumbering` and `\resumenumbering` which are just like `\endnumbering ... \beginnumbering`, except that they arrange for your line numbering to continue across the break. Use `\pausenumbering` only between numbered paragraphs:

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

\resumenumbering
\pstart
Another paragraph.
\pend
\endnumbering
```

We have defined these commands as two macros, in case you find it necessary to insert text between numbered sections without disturbing the line numbering. But if you are really just using these macros to save memory, you might as well type,  
`\newcommand{\memorybreak}{\pausenumbering\resumenumbering}`

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

## 5.3 Lineation commands

### 5.3.1 Disabling lineation

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

### 5.3.2 Setting lineation start and step

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

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

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

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

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

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

```
\gdef\linenumberlist{}
```

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

### 5.3.3 Setting lineation reset

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

You may only use this command at places where numbering is not in effect; you can't change the lineation system within a section. You can change it between sections: they don't all have to use the same lineation system. The package's standard setting is `\lineation{section}`. If the lineation is by `pstart`, the `pstart` number will be printed before the line number in the notes.

### 5.3.4 Setting line number margin

`\linenummargin` The command `\linenummargin{<location>}` specifies the margin where the line (or `pstart`) numbers will be printed. The permissible values for `<location>` are `left`, `right`, `inner`, or `outer`: for example, `\linenummargin{inner}`. The package's default setting is

```
\linenummargin{left}
```

to typeset the numbers in the left hand margin. You can change this whenever you're not in the middle of making a paragraph.

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

### 5.3.5 Other settings

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

## 5.4 Changing the line numbers

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

### 5.4.1 Sublineation

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

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

`Xsublinesep` You can change the separator between the line number and the subline number ei-  
`Xsublinesepside` ther by using `Xsublinesep` without any optional argument (7.2.9 p. 42) or by using `Xsublinesepside`. In the second case, it will change the separator only for line numbers in the margins, not in the footnotes.

### 5.4.2 Locking lineation

`\startlock` The `\startlock` command, used in running text, locks the line number at its current  
`\endlock` value, until you insert `\endlock`. It can tell for itself whether you are in a patch of line or sub-line numbering. One use for line-number locking is in printing poetry: there the line numbers should be those of verse lines rather than of printed lines, even when a verse line requires several printed lines. But in this case you may use the `\stanza` mechanism, see 9 p. 52.

`\lockdisp` When line-number locking is used, several printed lines may have the same line number, and you have to specify whether you want the number attached to the first printed line or the last, or whether you just want the number printed by them all, assuming that the settings of the previous parameters requires the display of a line number for this line. You specify your preference using `\lockdisp{<arg>}`; its argument is a word, either `first`, `last`, or `all`. The package initially sets this as `\lockdisp{first}`.

### 5.4.3 Setting and changing line number

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

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

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

#### 5.4.4 Line number style

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

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

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

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

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

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

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

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

#### 5.4.5 Skipping and hiding number

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

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

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

## 5.5 Executing code at each line

`\dolinehook`  
`\doinsidelinehook`

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

## 5.6 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}</code> { <code>\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. These are the main macros that you will use to construct the critical apparatus of your text.

If you need more series of critical notes, please look at 6.7.1 p. 37.

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

- `fulllines` to disable `\Xtwolines` and `\Xmorethantwolines` features for this note (cf. 7.2.5 p. 41).
- `nonum` disables line numbering for this note. A horizontal blank space is added instead. You can use `\Xinplaceofnumber` to set it (7.2.13 p. 43).
- `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. 45).
- `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** The mechanism is similar to the one for footnotes: each macro takes one or more optional arguments and one single argument, like:

**\Dendnote** `\Aendnote[<option>]{<text>}`.

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

- `fulllines` to disable `\Xendtwolines` and `\Xendmorethantwolines` features for this particular note (cf. 7.2.5 p. 41).
- `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. 45).
- `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. 45).

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

`toendnotes`  
`Xtoendnotes`

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

```
\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 :

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

```
\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. 60) to compute them automatically.

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

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

The commands for generating the apparatus have been given rather bland names, because editors in different fields have widely divergent notions of what sort of notes are required, where they should be printed, and what they should be called. But this does not mean you have to type `\Afootnote` when you would rather type something you find more meaningful, like `\variant`.

We recommend that you create a series of such aliases and use them instead of the names chosen here; all you have to do is put commands of this form at the start of your file:<sup>15</sup>

```
\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.

### 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”.

<sup>15</sup>We use `\newcommand` and `\newcommandx` instead of classical `\let` command because the `edtabular` environments have to modify the notes definition, and we need to use the newest definition of notes. Read the handbook of `xargs` to know more about `\newcommandx`.

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

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

For instance, in Greek, “α” has two possible unicode numbers:

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

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

Inside `reledmac`, the `\sameword` command considers these two unicodes (code positions) as different characters. If you use only one unicode number consistently, the distinction will probably make no difference to how your text looks, but `\sameword` will process the text inaccurately, based on the unicode numbers. To prevent this, do the following:

- If you use  $\text{Xe}\text{\LaTeX}$ , add this line in your preamble: `\XeTeXinputnormalization 1`.
- If you use  $\text{Lua}\text{\LaTeX}$ , use the `uninormalize` package of Michal Hoftich<sup>16</sup> with the `buffer` option set to true.

With these tools,  $\text{Xe}\text{\LaTeX}$  /  $\text{Lua}\text{\LaTeX}$  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}}.%
```

`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`:

<sup>16</sup><https://github.com/michal-h21/uninormalize>.

- 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`.

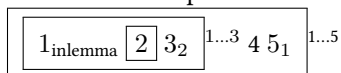
Note that only words that are actually referenced in a `\lemma` need the optional argument. Therefore, the first `\sameword` in the example above should have “1” as its optional argument, to be referenced correctly in the lemma.

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

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

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

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

Here, the corresponding code:

```

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

```

1    A (1) A (2) A (3) A (4) A (5)

---

1    A<sup>1</sup>...A<sup>5</sup> ] level 1

1    A<sup>1</sup>...A<sup>3</sup> ] level 2

1    A<sup>2</sup> (2) ] level 3

### 6.3.5 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<sup>2</sup>] 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  $\TeX$  argumentless macro, it will gobble the following space. To keep that space, you have to either append a backslash followed by a space or `{}` to `\stopmsdata`, like so:

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

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

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

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

#### 6.4.2 Layout of the apparatus of manuscripts

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

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

#### 6.4.3 Settings

As the apparatus of manuscripts technically consists of first-level critical notes (‘A’ series), any setting available for critical notes can be applied (7 p. 38). However, the following *additional* commands are available.

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

`\setmsdatalabel` As the apparatus of manuscripts consists of regular critical footnotes, a lemma is associated to them. By default, it is “Ms.”. You can change it using `\setmsdatalabel{<txt>}`.

`\setmsdataposition` If you want the manuscript apparatus to be on the same level of critical footnotes as the other apparatuses, for each line, `reledmac` will first insert the manuscript apparatus, then the other footnotes. You can change it using:

```
\msdataposition{regular-msdata}
```

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

## 6.5 Familiar notes

### 6.5.1 Basic use

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

### 6.5.2 Customizing mark

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

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

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

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

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

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

There are similar command triples for the other series.

### 6.5.3 Separator for multiple footnotes

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

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

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

and can be changed if necessary.

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

`\footnoteXmark` 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.

`\footnoteXtext`

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  $\TeX$ 's command `\footnotemark` and `\footnotetext`. For example, to use with `\uline`, do:

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

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

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

## 6.7 Changing series

### 6.7.1 Create a new series

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

### 6.7.2 Delete series

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

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

### 6.7.3 Series order

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

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

## 6.8 Position of critical and familiar footnotes

`\fnpos` There is a historical incoherence in (r)(e)ledmac. The familiar footnotes are before the critical footnotes in a normal page, but after in a minipage or in a ledgroup. However, `\mpfnpos` 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  $\text{\LaTeX}$ .

## 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  $\text{\LaTeX}$  to have:  $a - c \leq m \leq a + b$ . If you use some relative unit<sup>17</sup>, it will be relative to font size of the footnote, except for commands concerning the place kept by the notes — including blank space.

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

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

---

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

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 `⟨s⟩` series of critical footnotes and `\arrangementX[⟨s⟩]{⟨a⟩}` to change the arrangement of the `⟨s⟩` series of familiar footnotes.

The value of `⟨a⟩` 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. 76.
- `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  $\TeX$  will try to put too many or too few of these notes on each page. If you need to change the `\hsize` within the document, call the arrangement macro again afterwards to take account of the new value.

## 7.2 Control line number printing

### 7.2.1 Print line number only at first time

`\Xnumberonlyfirstinline`

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

`numberonlyfirstintwolines`

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 page number only at first time

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

`\Xendpagenumberonlyfirst`

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

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

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

In any case, you can use:

`\Xendsympagenum`

- `\Xendsympagenum[⟨series⟩]{⟨c⟩}` to print `⟨c⟩` when the page number is not printed.

`\Xendinplaceofpagenumber`

- `\Xendinplaceofpagenumber[⟨series⟩]{⟨l⟩}` to print a `⟨l⟩` length horizontal space in case no symbol is printed instead of the page number.



### 7.2.3 Arbitrary text before line number

<code>\Xbeforenumber</code>	<code>\Xbeforenumber[⟨s⟩]{⟨txt⟩}</code> allows the insertion of $\langle txt \rangle$ before the line number only when the line number is printed, so taking into account <code>\Xnumberonlyfirstinline</code> and the like.
<code>\Xendbeforepagenumber</code>	<b>For endnotes</b> <code>\Xendbeforepagenumber[⟨s⟩]{⟨text⟩}</code> defines the text before the page number in endnotes. Default value is p. (“p” followed by a dot).
<code>\Xendafterpagenumber</code>	<code>\Xendafterpagenumber[⟨s⟩]{⟨text⟩}</code> defines the text after the page number in endnotes. Default value is ) (open parenthesis followed by a single space).
<code>\Xendlineprefixsingle</code>	<code>\Xendlineprefixsingle[⟨s⟩]{⟨text⟩}</code> defines the text before the line number in endnotes when there is only one line. Default value is empty.
<code>\Xendlineprefixmore</code>	<code>\Xendlineprefixmore[⟨s⟩]{⟨text⟩}</code> 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 <code>\Xendlineprefixsingle</code> .

### 7.2.4 Separator for line range

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

### 7.2.5 Abbreviate line range

<code>\Xtwolines</code>	If a lemma is printed on two subsequent lines, <code>reledmac</code> 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)”.
<code>\Xmorethantwolines</code>	
To achieve this, use <code>\Xtwolines[⟨s⟩]{⟨text⟩}</code> and <code>\Xmorethantwolines[⟨s⟩]{⟨text⟩}</code> . The <code>⟨text⟩</code> argument of <code>\Xtwolines</code> will be printed if the lemma is on two lines, and the <code>⟨text⟩</code> argument of <code>\Xmorethantwolines</code> will be printed if the lemma is on three or more lines. For example:	
<pre>\Xtwolines{sq.} \Xmorethantwolines{sqq.}</pre>	

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[⟨series⟩]`.

When you use lineation by page, the final page number, if different from the initial page number, will not be printed, because the final page number is included in the `\Xendtwolines` symbol.

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

You can disable `\Xtwolines` and related for a specific note by using the ‘`[fulllines]`’ argument in the note macro cf. 6.2.2 p. 26.

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

### 7.2.6 Disable line number

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

### 7.2.7 Printing pstart number

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

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

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

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

### 7.2.8 Printing stanza number

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

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

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

### 7.2.9 Separator between line and subline numbers

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

**Employed without optional argument, it also 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{.}`.

### 7.2.10 Separator between page and line numbers

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

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

### 7.2.11 Space around number

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

`\Xafternumber` With `\Xafternumber[<s>]{<l>}` you can add some space after the line number in a footnote. If the line number is not printed, neither is the space. The default value is 0.5 em.

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

`\Xnonbreakableafternumber` By default, the space defined by `\Xafternumber` is breakable. With `\Xnonbreakableafternumber[<s>]{<l>}` it becomes nonbreakable.

### 7.2.12 Space around line symbol

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

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

`\Xendbeforesymmlinenumber` `\Xendaftersymmlinenumber` and `\Xendafternumber` are the equivalents of `\Xbeforesymmlinenumber` and `\Xaftersymmlinenumber` for the endnotes.

### 7.2.13 Space in place of number

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

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

### 7.2.14 Boxing line number and line symbol

`\Xboxlinenum` It could be useful to put the line number inside a fixed box: the content of the note will be printed after this box. You can use `\Xboxlinenum[<s>]{<l>}` to do that. To subsequently disable this feature, use `\Xboxlinenum` with length equal to 0 pt. One use of this feature is to print line number in a column, and the note in an other column:

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

`\Xboxlinenum{1em}`

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

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

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

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

**R** to align right;

**C** to center.

When using `\Xboxlinenum`, `reledmac` put all the line number description in the same box. That is, the same box will contain: the start line number, the dash, and either the end line number or the range symbol (like `ff.`). However, it is possible to box them in two different boxes.

- `\Xboxstartlinenum[⟨s⟩]{⟨l⟩}` will box the start line number in a box of length `⟨l⟩`. The content will be put at the right of the box.
- `\Xboxendlinenum[⟨s⟩]{⟨l⟩}` will box the dash plus the end line number or the range symbol in a box of length `⟨l⟩`. The content will be put at the left of the box.

With these two commands, it is possible to horizontally align the dash of line number when using critical notes, to obtain something like:

```
1
12-23
24ff.
```

`\Xendboxlinenum` `\Xendboxlinenum[⟨s⟩]{⟨l⟩}`, `\Xendboxlinenumalign[⟨s⟩]{⟨text⟩}`, `\Xendboxstartlinenum[⟨s⟩]{⟨l⟩}`

`\Xendboxendlinenum[⟨s⟩]{⟨l⟩}` are the same as, respectively, `\Xboxlinenum` and

`\Xendboxstartlinenumalign` `\Xboxlinenumalign`, `\Xboxstartlinenum`, `\Xboxendlinenum` except in endnotes.

`\Xendboxendlinenumalign`

### 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 `\Xendbeforesymlinenum` 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 `\Xendaftersymlinenum` 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>18</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

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

numbers in critical footnotes.  $\langle command \rangle$  must be one (or more) switching command, like `\bfseries`.

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

### 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[\langle s \rangle]` 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 command allows `\Xendlemmadisablefontselection[\langle s \rangle]` to disable it for a specific series.

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

`\Xendlemmafont`

`\Xlemmafont{\bfseries}`

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

### 7.5.3 For all notes

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

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

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

## 7.6 Wrapping notes

### 7.6.1 Wrapping lemmas

`\Xwraplemma` `\Xwraplemma[\langle s \rangle]{\langle cmd \rangle}` is used to wrap, in the footnote, the lemma in a  $\text{\TeX}$  command. For example, with the  `bidi`  package, to ensure having a lemma written right to left, use `\Xwraplemma{\RL}`.

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

### 7.6.2 Wrapping contents

<code>\Xwrapcontent</code>	<code>\Xwrapcontent[⟨s⟩]{⟨cmd⟩}</code> 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 <code>\Xwrapcontent{\foreignlanguage{⟨language⟩}}</code> (with <code>babel</code> ) or <code>\Xwrapcontent{\text{⟨language⟩}}</code> (for <code>babel</code> ).
<code>\Xendwrapcontent</code>	<code>\Xendwrapcontent[⟨s⟩]{⟨cmd⟩}</code> is the same for endnotes.
<code>\wrapcontentX</code>	<code>\wrapcontentX[⟨s⟩]{⟨cmd⟩}</code> is the same for familiar footnotes.

## 7.7 Indent of notes content

<code>\Xparindent</code>	By default, <code>reledmac</code> does not add indentation before the paragraphs inside critical footnotes. Use <code>\Xparindent[⟨s⟩]</code> to enable indentation.
<code>\parindentX</code>	By default, <code>reledmac</code> does not add indentation before the paragraphs inside familiar footnotes. Use <code>\parindentX[⟨s⟩]</code> to enable indentation.
<code>\Xhangindent</code>	For critical notes NOT paragraphed you can define an indent with <code>\Xhangindent[⟨s⟩]{⟨l⟩}</code> , which will be applied in the second line of notes. It can help to make distinction between a new note and a break in a note. The default value is 0 pt.
<code>\hangindentX</code>	For familiar notes NOT paragraphed you can define an indentation with <code>\hangindentX[⟨s⟩]{⟨l⟩}</code> , which will be applied in the second line of notes. It can help to make a distinction between a new note and a break in a note.
<code>\Xendhangindent</code>	For critical endnotes NOT paragraphed you can define an indentation with <code>\Xendhangindent[⟨s⟩]{⟨l⟩}</code> , which will be applied in the second line of notes. It can help to make a distinction between a new note and a break in a note.

## 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 :

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

<code>\Xhooknote</code>	<code>\Xhooknote[⟨s⟩]{⟨code⟩}</code> is to be used at the beginning of each critical footnote.
<code>\hooknoteX</code>	<code>\hooknoteX[⟨s⟩]{⟨code⟩}</code> is to be used at the beginning of each familiar footnote.
<code>\Xendhooknote</code>	<code>\Xendhooknote[⟨s⟩]{⟨code⟩}</code> is to be used at the beginning of each endnote.

## 7.9 Arbitrary code before inserting note

<code>\Xbeforeinserting</code> <code>beforeinserting</code>	<code>\Xbeforeinserting[⟨s⟩]{⟨code⟩}</code> and <code>\beforeinsertingX[⟨s⟩]{⟨code⟩}</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. 46)).



Note that the changes are local to the footnote.

## 7.10 Options for footnotes in columns

### 7.10.1 Alignment

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

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

`\justifying` to have text justified, as usual with  $\TeX$ . 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.

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

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

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

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

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

### 7.11.1 Mark separation of notes

`\Xafternote` You can add some horizontal space after a note by using `\Xafternote[⟨s⟩]{⟨l⟩}` (for critical footnotes) or `\afternoteX[⟨s⟩]{⟨l⟩}` (for familiar footnotes). The default value is `1em plus.4em minus.4em`.

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



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

### 7.11.2 Ragged text

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

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

## 7.12 Options for block of notes

### 7.12.1 Grouping notes by line

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

In many cases, you might like to use it in combination with `\Xnumberonlyfirstinline` (7.2.1 p. 39).

`\groupbylineseparetwolines` Note that the `\Xafternote` and `\Xparafootsep` settings are used to determine space and content between footnotes (7.11 p. 48). 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. 39).

### 7.12.2 Text before notes

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

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

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

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

### 7.12.3 Code before notes

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

### 7.12.4 Spacing

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

**Be careful, the standard L<sup>A</sup>T<sub>E</sub>X footnote rule used by reledmac decreases by 3pt. This 3pt decrease is not changed by this command.**

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

**Be careful, the standard L<sup>A</sup>T<sub>E</sub>X footnote rule, which is used by reledmac, decreases 3pt. These 3pt are not changed by this command.**

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

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

### 7.12.5 Rule

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

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

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

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

### 7.12.6 Maximum height

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

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

Note that in many cases, you should call these commands in the begin of the document, because the `\vsize` in the preamble is not the same as `\vsize` after the preamble. That why we recommend to you to add in your preamble

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

Be careful with the two previous commands. Actually, for technical purposes, one paragraphed note is considered as one block. Consequently, it cannot be broken between two pages, even if you used these commands. The debug is in the *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 width of the block of notes. If you want to change the width of each column when typesetting notes in columns, use `\Xhsizetwocol`, `\Xhsizethreecol`, `\hsizetwocolX`, `\hsizethreecolX`, see 7.10.2 p. 48.

## 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 9.10.2 p. 57 and 9.10.3 p. 58.

### 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. 55)

### 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. 54.

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. 385.**

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` You can set the indent of some specific verse by calling `\stanzaindent{⟨value⟩}` at the  
`\stanzaindent*` 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. 74 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 the stanza (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 stanza (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`.

## 9.9 Numbering stanza

### `\numberstanzatrue`

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

### `\numberstanzafalse`

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

You can use this feature in combination with `\Xstanza` (7.2.8 p. 42).

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

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

You can change the value of the stanza counter with the usual commands of  $\text{\LaTeX}$ .

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

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

## 9.10 Adding annotations to line numbers

You may want to have two or more numbers associated with a line of verse. 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.

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>19</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 verse. It is reset with each new line of verse.
- It is printed alongside the line number in the margin.
- It is printed following the line number in critical footnotes and endnotes.

---

<sup>19</sup>However, you may create your own commands dealing with such patterns which will in turn call `reledmac` commands.



- 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}&
```

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.

### 9.10.1 Modifying annotation associated with a specific note

`\lineannot` The `\linenum` (6.2.5 p. 28) 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
```

### 9.10.2 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.

### 9.10.3 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. 60).
- `\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. 60).  
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>20</sup>

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

---

<sup>20</sup>The `\@gobble` command takes one argument, and returns nothing.

## 9.11 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`.

## 9.12 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. 55).

## 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. 62).

### 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. 28).

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. 28) and `\lineannot` (?? p. ??) 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 to label which are named `<label>:start` and `<label>:end`.  
`\edlabelSE`      Use `\edlabelSE{<label>}` to mark just one location in the text. Contrary to a 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. 28) 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. 63). 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

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`
- `\Xendlinerrangeseparator`
- `\Xendmorethantwolines`
- `\Xendsublinesep`
- `\Xendtwolines`
- `\Xendtwolinesbutnotmore`
- `\Xendtwolinesonlyinsamepage`
- `\Xendlinenumannotationposition`
- `\Xendwraplinenumannotation`

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

When calling `\apprefwithpage`, `\Serefwithpage` or `\Serefonlypage` you can use as a first optional argument, in brackets (`[]`), any optional argument which can be used for critical endnotes (6.2.3 p. 26).

## 11.7 Compatibility with `xr` package

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

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

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



## 12 Sidenotes

### 12.1 Basics

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

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

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

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

If two note commands for the same side are called in the same line, they will be appended and separated by a comma.

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

### 12.2 Setting

#### 12.2.1 Width

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

#### 12.2.2 Vertical position

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

#### 12.2.3 Distance to the main text

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

#### 12.2.4 Font

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

```
\newcommand*{\ledrsnotefontsetup}{\raggedright\footnotesize}% right
```

These can of course be changed to suit.

### 12.2.5 Separator between notes

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

## 13 Indexing

### 13.1 Basics

`\edindex` L<sup>A</sup>T<sub>E</sub>X 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 & lineNumber should be added to the raw index file.

Note that the file .idx will contain the right reference only after the third run, because of the internal indexing mechanism of reledmac. That means you must first run (Xe/Lua)L<sup>A</sup>T<sub>E</sub>X three times, then run `makeindex`, and then finally run (Xe/Lua)L<sup>A</sup>T<sub>E</sub>X 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.

### 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-linenum`

An example of such a file is provided in the “examples” folder. Read the `xindy` handbook to learn how to use it.<sup>23</sup>

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

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

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

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

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

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

### 13.6 Advanced setting

`\edindexlab` The `\edindex` process uses a `\label` and `\ref` mechanism to get the correct line number. It automatically generates labels of the form `\label{\edindexlab N}`, where `N` is a number, and the default definition of `\edindexlab` is:

```
\newcommand*{\edindexlab}{\&\&}
```

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

## 14 Glossary

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

### 14.1 Preamble setting

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

- Or set `.glossaries`:  
`\glsSetCompositor{-}`
- Or set `reledmac`:  
`\renewcommand{\pagelinesep}{-}`  
 In this case, the above will have consequences for your use of `\edindex` and you should set your `.ist` file (13.4 p. 67).

### 14.2 Commands

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

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

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

## 15 Tabular material

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

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

```

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

```

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

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

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

```

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

```

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

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

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

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

```

\begin{edtabularr}
1 & & & & & & & & & & \\
Q & & & & & & & & & & \\
v & & & & & & & & & & \\
g & & & & & & & & & & \\
\edrowfill{1}{3}{\downbracefill} & & & & & & & & & & \\
k & & & & & & & & & & \\
1 & & & & & & & & & & \\
\end{edtabularr}

```

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

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

```

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

```

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

```

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

```

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

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

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

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

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

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

For example:

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

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

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

`\edvertdots`

```
\begin{edarrayr}
a & b & c & d & & \\
v & w & x & y & & \end{arrayr}
```

```

m & n & o & p & \\\
k & & L & cvb & \edvertline{4pc}
\end{edarrayr}

```

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

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

## 16 Sectioning commands

### 16.1 Sectioning commands without line numbers or critical notes

The standard sectioning commands (`\chapter`, `\section` etc.) can be used inside numbered text. In this case, you must call them as an optional argument of `\pstart` (5.2.3 p. 18):

```

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

```

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

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

You have to use the following commands:

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



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

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

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

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

## 16.3 Optimization

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

## 17 Quotation environments

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

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

## 18 Page breaks

### 18.1 Control page breaking

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

`\ledpb`  
`\lednopb`

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

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

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

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

## 18.2 Prevent page break in a long verses

You can also decide to prevent page breaks between two lines of a long verse. To do this, use `nopbinverse` when loading package, or add `\lednopbinversetrue` in the beginning of your file (better: in your preamble).

This feature works only with verse of 2 lines and no more. It works on the third run, or on the fourth run if using `reledpar`. By default, when a long verse runs between two pages, a page break will be placed at the beginning of the verse. However, if you have added `\ledpbsetting{after}`, the page break will be placed at the end of the long verse and the page containing the long verse will have one extra line.

## 19 Miscellaneous

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

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

The lemma within the text is printed via `\showlemma{lemma}`. Normally, or with the ‘final’ option, the definition of `\showlemma` is:

```
\newcommand*{\showlemma}[1]{#1}
```

so it just produces its argument. With the ‘draft’ option it is defined as

```
\newcommand*{\showlemma}[1]{\textit{#1}}
```

so that its argument is typeset in an italic font, which may make it easier to check that all lemmas have been treated.

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

```
\ifledfinal\else
```

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

## 19.1 Known and suspected limitations

### 19.1.1 Non-standard geometry

If you use classes other than `article` or `book`, or if you use the `geometry` package, you should use `maxhnotesX` and/or `\Xmaxhnotes` as explained in 7.12.6 p. 50 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. 73 in order to know which are the available options.

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

‘no room for a new count’ is often caused by `biblatex` being used at the same time.

Load `reledmac` (and `reledpar`) *before* `biblatex`.

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

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

### 19.1.4 Marginal notes

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

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

### 19.1.5 Paragraph shape

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

`\ballast`

$\TeX$  is a three-pass system, but even after a document has been processed three times, there are some tricky situations in which the page breaks decided by  $\TeX$  never settle down. At each successive run, `reledmac` may oscillate between two different sets of page decisions. To stop this happening, should it arise, Wayne Sullivan suggested the inclusion of the quantity `\ballast`. The amount of `\ballast` will be subtracted from the penalties which apply to the page breaks calculated on the *previous* run through  $\TeX$ , thus reinforcing these breaks. So if you find your page breaks oscillating, insert `\setcounter{ballast}{100}` or some such figure, and with any luck the page breaks will settle down. Luckily, this problem does not crop up at all often.

### 19.1.6 Paragraphed footnotes

The restriction on explicit line-breaking in paragraphed footnotes, mentioned on 7.1 p. 39, and described in more detail on XIII.6.3 p. 192, 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. 50), in order to define the maximum height relative to `\textheight` and not to `\vsize`, because the `\vsize` value is not the same inside and outside of the preamble.

### 19.1.7 Use with other packages

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

`\morenoexpands`

You can define the macro `\morenoexpands` to modify macros that you call within `\edtext`. Because of the way `reledmac` numbers the lines the arguments to `\edtext` can be processed more than once and in some cases a macro should only be processed

once. One example is the `\colorbox` macro from the `color` package, which you might use like this:

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

If you actually try this<sup>25</sup> you will find  $\TeX$  whinging ‘Missing { inserted’, and then things start to fall apart. The trick in this case is to specify either:

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

or

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

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

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

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

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

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

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

## 19.2 Parallel typesetting

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

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

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

## I Implementation overview

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

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

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

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

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

## II Preliminaries

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

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

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

### II.2 Package declaration

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

```

1 %<*code>
2 \NeedsTeXFormat{LaTeX2e}
3 \ProvidesPackage{reledmac}[2018/10/12 v2.27.1 typesetting critical editions]
4 %

```

## II.3 Package options

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

```

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

```

### II.3.1 Options of reledpar

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

```

7 \newif\ifparledgroup
8 \newif\ifsameparallelpagenumber%
9 \newif\ifprevpgnotnumbered%%
10 %
11 \newif\ifnoledgroup%
12 \DeclareOptionX{noledgroup}{\noledgroup@true}%
13 %
14 \newif\ifnoend@%
15 \DeclareOptionX{noend}{%
16 \let\l@dend@open\@gobble%
17 \let\l@dend@close\relax%
18 \global\let\l@dend@stuff=\relax%
19 \noend@true%
20 }%

```

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

```



```

82 \sw@caseinsensitivetrue%
83 }%
84 %

```

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.

```

85 \ProcessOptionsX*\relax
86
87 %

```

## 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 `LuaTeX` or `XYTeX` is running, and `ragged2e` to manage ragged justification for paragraphed notes.

```

88 \RequirePackage{xargs}
89 \RequirePackage{xparse}[2017/03/07]%
90 \RequirePackage{etoolbox}
91 \@ifl@t@r\fmtversion{2015/10/01}
92 {\ifbool{expr}{not test{\@ifl@t@r\fmtversion{2016/03/31}} or (test{\
93   ifdefstring{\fmtversion}{2016/03/31}} and test {\ifnumless{\patch@level
94   }{3}})}}}%
95   {\PackageWarning{reledmac}{You are using a LaTeX version older than
96   2016/03/31 patch 3.%
97   \MessageBreak You are strongly encouraged to use a newer version.}}}%
98   {}%
99 }%
100 \RequirePackage{suffix}
101 \RequirePackage{xstring}
102 \RequirePackage{ifluatex}
103 \RequirePackage{ragged2e}
104 \RequirePackage{ifxetex}%
105 %

```

## II.5 Compatibility with LuaTeX

Here, we enable some primitives for `LuaTeX`.

```

106 \ifx\directlua\undefined\else%
107   \directlua{tex.enableprimitives("",{"texdir","pdir","bodydir"})}
108 \fi
109 %

```

## II.6 Boolean flags

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

```

110 \newif\ifl@dmemoir
111 \ifclassloaded{memoir}{\l@dmemoirtrue}{\l@dmemoirfalse}
112
113 %

```

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

```

114 \newif\if@ledgroup%
115 %

```

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

```

116 \newif\ifl@imakeidx
117 \@ifpackageloaded{imakeidx}{\l@imakeidxtrue}{}%False is the default value
118 %

```

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

```

119 \newif\ifl@indextools%
120 \@ifpackageloaded{indextools}{%
121   \l@indextoolstrue%
122   \l@imakeidxtrue%
123   \let\imki@wrindexentry\indtl@wrindexentry%
124 }{}%
125 %

```

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.

```

126 \newif\ifl@footmisc
127 \@ifpackageloaded{footmisc}{\l@footmisctrue}{}%False is the default value
128 %

```

**\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.

```

129 \ifdef{\if@RTL}{\newif\if@RTL}
130 %

```

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

```
131 \newif\if@firstlineofpage%
132 \newif\if@firstlineofpageR%
133 %
```

## II.7 Messages

All the messages are grouped here as macros. This saves  $\text{\TeX}$ 's memory when the same message is repeated and also lets them be edited easily.

`\reledmac@warning` Write a warning message.

```
134 \newcommand{\reledmac@warning}[1]{\PackageWarning{reledmac}{#1}}
135 %
```

`\reledmac@error` Write an error message.

```
136 \newcommand{\reledmac@error}[2]{\PackageError{reledmac}{#1}{#2}}
137 %
```

```
\led@err@NumberingStarted 138 \newcommand*\led@err@NumberingStarted}{%
\led@err@NumberingNotStarted 139 \reledmac@error{Numbering has already been started}{\@ehc}}
NumberingShouldHaveStarted 140 \newcommand*\led@err@NumberingNotStarted}{%
141 \reledmac@error{Numbering was not started}{\@ehc}}
142 \newcommand*\led@err@NumberingShouldHaveStarted}{%
143 \reledmac@error{Numbering should already have been started}{\@ehc}}
144 %
```

```
\led@err@edtextoutsidestart 145 \newcommand*\led@err@edtextoutsidestart}{%
\led@err@PstartInEdtext 146 \reledmac@error{\string\edtext\space outside numbered paragraph (\pstart\
\led@err@PendInEdtext 147 \space ... \space\pend)}{\@ehc}}%
148
149 \newcommand{\led@err@PstartInEdtext}[1]{%
150 \reledmac@error{\string\pstart\space in \string\edtext\space #1 argument
151 }{\@ehc}}%
152 }%
153
154 \newcommand{\led@err@PendInEdtext}[1]{%
155 \reledmac@error{\string\pend\space in \string\edtext\space #1 argument}{\@ehc}}%
156 }%
157 %
```

```

\led@mess@NotesChanged56 \newcommand*{\led@mess@NotesChanged}{%
157   \typeout{reledmac reminder: }%
158   \typeout{ The number of the footnotes in this section
159     has changed since the last run.}%
160   \typeout{ You will need to run LaTeX two more times
161     before the footnote placement}%
162   \typeout{ and line numbering in this section are
163     correct.}}
164 %

```

```

\led@mess@SectionContinued65 \newcommand*{\led@mess@SectionContinued}[1]{%
166   \message{Section #1 (continuing the previous section)}%
167 %

```

```

\led@err@LineationInNumbered68 \newcommand*{\led@err@LineationInNumbered}{%
169   \reledmac@error{You can't use \string\lineation\space within
170     a numbered section}{\@ehc}}
171 %

```

```

\led@warn@BadLineation72 \newcommand*{\led@warn@BadLineation}{%
\led@warn@BadLinenummargin73   \reledmac@warning{Bad \string\lineation\space argument}}
\led@warn@BadLockdisp74 \newcommand*{\led@warn@BadLinenummargin}{%
\led@warn@BadSublockdisp75   \reledmac@warning{Bad \string\linenummargin\space argument}}
176 \newcommand*{\led@warn@BadLockdisp}{%
177   \reledmac@warning{Bad \string\lockdisp\space argument}}
178 \newcommand*{\led@warn@BadSublockdisp}{%
179   \reledmac@warning{Bad \string\sublockdisp\space argument}}
180 %

```

```

\led@warn@NoFile81 \newcommand*{\led@warn@NoFile}[1]{%
182   \reledmac@warning{File `#1' not found}}
183 %

```

```

\led@warn@LineFileObsolete84 \newcommand*{\led@warn@Obsolete}[1]{%
185   \reledmac@warning{Line-list file #1 was obsolete. We have not read it.
186     Please run LaTeX again.}}
187 %

```

```

\led@warn@BadAdvancelineSubline87 \newcommand*{\led@warn@BadAdvancelineSubline}{%
\led@warn@BadAdvancelineLine88   \reledmac@warning{\string\advanceline\space produced a sub-line
189     number less than zero.}}
190 \newcommand*{\led@warn@BadAdvancelineLine}{%
191   \reledmac@warning{\string\advanceline\space produced a line
192     number less than zero.}}
193 %

```

```

\led@warn@BadSetline94 \newcommand*{\led@warn@BadSetline}{%
\led@warn@BadSetlinenum95 \reledmac@warning{Bad \string\setline\space argument}}
196 \newcommand*{\led@warn@BadSetlinenum}{%
197 \reledmac@warning{Bad \string\setlinenum\space argument}}
198 %

\led@err@PstartNotNumbered99 \newcommand*{\led@err@PstartNotNumbered}{%
\led@err@PstartInPstart100 \reledmac@error{\string\pstart\space must be used within a
\led@err@PendNotNumbered101 numbered section %
\led@err@PendNoPstart102 (\string\beginnumbering\space ...\space\string\endnumbering)
} {\@ehc}}%
\led@err@AutoparNotNumbered103 \newcommand*{\led@err@PstartInPstart}{%
\err@NumberingWithoutPstart203 \reledmac@error{\string\pstart\space encountered while another
204 \string\pstart\space was in effect} {\@ehc}}
205 \newcommand*{\led@err@PendNotNumbered}{%
206 \reledmac@error{\string\pend\space must be used within a
207 numbered section} {\@ehc}}
208 \newcommand*{\led@err@PendNoPstart}{%
209 \reledmac@error{\string\pend\space must follow a \string\pstart} {\@ehc}}
210 \newcommand*{\led@err@AutoparNotNumbered}{%
211 \reledmac@error{\string\autopar\space must be used within a
212 numbered section} {\@ehc}}
213 \newcommand*{\led@err@NumberingWithoutPstart}{%
214 \reledmac@error{\string\beginnumbering...\string\endnumbering\space
215 without \string\pstart} {\@ehc}}%
216 %

\led@warn@BadAction117 \newcommand*{\led@warn@BadAction}{%
218 \reledmac@warning{Bad action code, value \next@action.}}
219 %

\led@warn@DuplicateLabel120 \newcommand*{\led@warn@DuplicateLabel}[1]{%
\ppLabelOutSecondArgEdtext221 \reledmac@warning{Duplicate definition of label `#1'\@gobble}%
\led@warn@RefUndefined222 \@latex@warning@no@line{Label `#1' multiply defined}%
\led@warn@RefUndefined223 }%
224 \newcommand*{\led@warn@AppLabelOutSecondArgEdtext}[1]{%
225 \reledmac@warning{\string\applabel\space outside of the second argument
of an \string\edtext\space `#1' on page \thepage.}}%
226 \newcommand*{\led@warn@RefUndefined}[1]{%
227 \G@refundefinedtrue%
228 \reledmac@warning{Reference `#1' on page \thepage\space undefined.%
229 Using `000'.}%
230 \@latex@warning{Reference `#1' undefined\on@line}%
231 }%
232 \newcommand*{\led@warn@pairRefUndefined}[1]{%
233 \G@refundefinedtrue%

```

```

234 \reledmac@warning{Reference `#1:start' and/or `#1:end' on page \thepage\
space undefined.
235         Using `??'.}%
236 \@latex@warning{Reference `#1:start' and/or `#1:end' undefined\on@line}%
237 }
238 %

```

```

\led@warn@NoMarginpars39 \newcommand*\led@warn@NoMarginpars{%
240 \reledmac@warning{You can't use \string\marginpar\space in numbered text
}}
241 %

```

```

\led@warn@BadSidenotemargin42 \newcommand*\led@warn@BadSidenotemargin{%
243 \reledmac@warning{Bad \string\sidenotemmargin\space argument}}
244 %

```

```

\led@warn@NoIndexFile45 \newcommand*\led@warn@NoIndexFile[1]{%
246 \reledmac@warning{Undefined index file #1}}
247 %

```

```

\led@warn@SeriesStillExist48 \newcommand*\led@warn@SeriesStillExist[1]{%
249 \reledmac@warning{Series #1 is still existing !}}%
250 }%
251 %

```

```

\led@err@BadAction52 \newcommand*\led@err@StanzaIndentNotDefined{%
253 \reledmac@error{You have not defined the indentation for the line \number
\stanza@count}\@ehc}}%
254 %

```

```

\led@err@ManySidenotes55 \newcommand*\led@err@ManySidenotes{%
\led@err@ManyLeftnotes56 \ifledRcol%
\led@err@ManyRightnotes57 \reledmac@warning{\itemcount@\space sidenotes on line \the\line@numR\
space p. \the\page@numR}%
258 \else%
259 \reledmac@warning{\itemcount@\space sidenotes on line \the\line@num\
space p. \the\page@num}%
260 \fi%
261 }%
262 \newcommand*\led@err@ManyLeftnotes{%
263 \ifledRcol%
264 \reledmac@warning{\itemcount@\space leftnotes on line \the\line@numR\
space p. \the\page@numR}%

```

```

265 \else%
266 \reledmac@warning{\itemcount@\space leftnotes on line \the\line@num\
space p. \the\page@num}%
267 \fi%
268 }%
269 \newcommand{\led@err@ManyRightnotes}{%
270 \ifledRcol@%
271 \reledmac@warning{\itemcount@\space rightnotes on line \the\line@numR\
space p. \the\page@numR}%
272 \else%
273 \reledmac@warning{\itemcount@\space rightnotes on line \the\line@num\
space p. \the\page@num}%
274 \fi%
275 }%
276 %

```

```

\led@err@TooManyColumns 77 \newcommand*\led@err@TooManyColumns}{%
\led@err@UnequalColumns 78 \reledmac@error{Too many columns}{\@ehc}}
\led@err@LowStartColumn 79 \newcommand*\led@err@UnequalColumns}{%
\led@err@HighEndColumn 80 \reledmac@error{Number of columns is not equal to the number
\led@err@ReverseColumns 81 in the previous row (or \protect\\ \space forgotten?)}{\@ehc}}
282 \newcommand*\led@err@LowStartColumn}{%
283 \reledmac@error{Start column is too low}{\@ehc}}
284 \newcommand*\led@err@HighEndColumn}{%
285 \reledmac@error{End column is too high}{\@ehc}}
286 \newcommand*\led@err@ReverseColumns}{%
287 \reledmac@error{Start column is greater than end column}{\@ehc}}
288 %

```

```

\endnotes@outsidenumbering 89 \newcommand{\led@err@toendnotes@outsidenumbering}{%
290 \reledmac@error{\string\toendnotes\space and related commands must be
called inside a numbered texte (\string\beginnumbering\space ... \space \string
\endnumbering)}{\@ehc}%
291 }%
292 %

```

```

\err@EdtextWithoutFootnote 93 \newcommand{\led@err@EdtextWithoutFootnote}{%
294 \reledmac@error{edtext without Xfootnote. Check syntax}{\@ehc}%
295 }%
296 %

```

```

\footnoteNotInSecondArgEdtext 97 \newcommand{\led@err@FootnoteNotInSecondArgEdtext}[1]{%
298 \reledmac@error{#1footnote outside of the second argument of an edtext.
Check syntax}{\@ehc}%
299 }%
300 %

```

```

\led@error@PackageAfterEledmac01 \newcommand{\led@error@PackageAfterEledmac}[1]{%
302   \reledmac@error{#1 must be loaded before reledmac}{\@ehc}%
303 }%
304 %

```

```

\led@error@fail@patch@@makecol05 \newcommand{\led@error@fail@patch@@makecol}{%
306   \reledmac@error{Fail to patch \string\@makecol\space command}{\@ehc}%
307 }%
308 %

```

```

\led@error@fail@patch@@reinserts09 \newcommand{\led@error@fail@patch@@reinserts}{%
310   \reledmac@error{Fail to patch \string\@reinserts\space command}{\@ehc}%
311 }%
312 %

```

```

\led@error@fail@patch@@docclearpage13 \newcommand{\led@error@fail@patch@@docclearpage}{%
314   \reledmac@error{Fail to patch \string\@docclearpage\space command}{\@ehc}%
315 }%
316 %

```

```

\led@error@fail@patch@@iiiminipage17 \newcommand{\led@error@fail@patch@@iiiminipage}{%
318   \reledmac@error{Fail to patch \string\@iiiminipage\space command}{\@ehc}%
319 }%
320 %

```

```

\led@error@fail@patch@endminipage21 \newcommand{\led@error@fail@patch@endminipage}{%
322   \reledmac@error{Failed to patch the \string\endminipage\space command}{\@ehc}%
323 }%
324 %

```

```

\led@error@fail@patch@endminipage25 \newcommand{\led@error@fail@patch@makeindex}{%
326   \reledmac@error{Failed to patch the \string\makeindex\space command}{\@ehc}%
327 }%
328 %

```

```

\led@warn@edinde@outsidenumbering29 \newcommand{\led@warn@edinde@outsidenumbering}{%
330   \reledmac@warning{\string\edindex\space called outside of \string\beginnumbering\space ... \space \string\endnumbering. \MessageBreak Automatically switched to \string\index.}%
331 }%
332 %

```



```

warning@hsizeX@deprecated33 \newcommand{\led@warning@hsizeX@deprecated}{%
334 \reledmac@warning{\string\hsizeX\space command deprecated, use \string\
widthX\space instead.}%
335 }%
336 %

```

```

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

```

```

warning@msdatawithoutstop41 \newcommand{\led@warning@msdatawithoutstop}{%
342 \reledmac@warning{\string\msdata\space without corresponding \string\
stopmsdata}%
343 }%
344 %

```

```

ning@preXnotes@deprecated45 \newcommand{\led@warning@preXnotes@deprecated}{%
346 \reledmac@warning@preXnotes@deprecated%
347 }%
348 %

```

## II.8 Gobbling

Here, we define some commands which gobble their arguments.

```

\@gobblethree49 \providecommand*\@gobblethree}[3]{}
\@gobblefour50 \providecommand*\@gobblefour}[4]{}
\@gobbleseven51 \providecommand*\@gobbleseven}[7]{}
352 %

```

## II.9 Wrapping content

`\l@wrap@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@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).

```

353 \newcommand{\l@wrap@ifnotemptybox}[2]{%
354   \setbox0=\hbox{#2}%
355   \ifdim\wd0=\z@ \else%
356     #1{#2}%
357   \fi%
358 }%
359 \newcommand{\l@wrapcs@ifnotemptybox}[2]{%
360   \l@wrap@ifnotemptybox{\csname #1\endcsname}{#2}%
361 }%
362 %

```

## II.10 Miscellaneous commands

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

```

363 \ifledfinal
364   \newcommand*{\showlemma}[1]{#1}
365 \else
366   \newcommand*{\showlemma}[1]{\underline{#1}}
367 \fi
368
369 %

```

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

```

370 \let\linenumberlist=\empty
371
372 %

```

`\@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).

```

373 \newcount\@l@tempcnta \newcount\@l@tempcntb
374 %

```

## 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`  
`\ifpstartedsL`  
`\l@dnumprintstartsl`

```

375 \newif\ifl@dpairing
376 \newif\ifl@dpaging%
377 \newif\ifl@dprintingpages%

```

```

378 \newif\ifl@dprintingcolumns%
379 \newif\ifpst@rtedL
380 \newcount\l@dnumstartsL
381 %

```

`\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`.

```

382 \newif\ifledRcol
383 \newif\ifledRcol@
384 %

```

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

```

385 \newif\ifnumberingR
386 %

```

`\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`.

```

387 \newif\ifafterendnumberingR%
388 %

```

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

```

389 \newif\ifXnote@%
390 %

```

## 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.

```

391 \providebool{indtl@innote}%
392 \providebool{indtl@notenumber}%
393 %

```

## 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.  $\text{\LaTeX}$  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.

```
394 \newcount\section@num
395 \section@num=0
396 \let\extensionchars=\empty
397 %
```

`\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.

`\numberingtrue`

`\numberingfalse`

```
398 \newif\ifnumbering
399 %
```

`\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.

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

```

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

```

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.

```

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

```

```

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

```

`\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.

```

476 \def\endnumbering{%
477   \ifnumbering
478     \global\numberingfalse
479     \normal@pars
480     \ifnum\l@dnumpstartsL=0%
481       \led@err@NumberingWithoutPstart%
482     \fi%
483 %

```

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`.

```

484   \global\page@num=\this@section@page@num%
485   \global\last@page@num=\this@section@last@page@num%
486   \global\let\next@page@num\this@section@next@page@num%
487   \ifl@dpairing

```

```

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

```

`\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 `\ifresumenumbering@start` `\pausenumbering@page@num` counter stores the `\this@section@next@page@num` when the `\pause@numbering` is called.

```

513 \newcount\pausenumbering@page@num%
514 \newcommand{\pausenumbering}{%
515 \ifx\this@section@next@page@num\relax%
516 \global\pausenumbering@page@num=0%
517 \else%
518 \global\pausenumbering@page@num=\this@section@next@page@num%
519 \fi%
520 \ifautopar\global\autopar@pausettrue\fi%
521 \endnumbering\global\numberingtrue}
522 %

```

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.

```

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

```

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

```

561 \newcommand{\set@continuousnumberingforL}{%
562   \ifcontinuousnumberingwithcolumns%
563     \ifl@dpairing%
564       \unless\ifl@dpaging%
565         \global\c@pstartL=\c@pstart%
566       \fi%
567     \fi%
568   \fi%

```



```

569 }%
570 %

```

## IV List macros

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

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

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

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

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

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

```

571 \newcommand*{\list@create}[1]{%
572   \global\let#1=\empty%
573 }%
574 %

```

`\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 .

```

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

```

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

```

579 \newtoks\led@toksa \newtoks\led@toksb
580 \global\led@toksa={\}
581 \long\def\xright@appenditem#1\to#2{%
582   \global\led@toksb=\expandafter{#2}%
583   \xdef#2{\the\led@toksb\the\led@toksa\expandafter{#1}}%
584   \global\led@toksb={}}
585 %

```

`\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`.

```
586 \long\def\xleft@appenditem#1\to#2{%
587   \global\led@toksb=\expandafter{#2}%
588   \xdef#2{\the\led@toksa\expandafter{#1}\the\led@toksb}%
589   \global\led@toksb={}}
590 %
```

`\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.

```
591 \def\gl@p#1\to#2{\expandafter\gl@poff#1\gl@poff#1#2}
592 \long\def\gl@poff\#1#2\gl@poff#3#4{\gdef#4{#1}\gdef#3{#2}}
593
594 %
```

## V Line counting

### V.1 Choosing the system of lineation

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

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

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

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

`\ifbypage@`

`\bypage@true`

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

```
595 \newif\ifbypage@
596 \newif\ifbypstart@
597 %
```

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

```
\ifbypage@R 598 \newif\ifbypage@R
\ifbypstart@R 599 \newif\ifbypstart@R
600 %
```

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

```

601 \newcommand*{\lineation}[1]{%
602 %

```

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

```

603 \ifnumbering
604 \led@err@LineationInNumbered
605 \else
606 %

```

If the argument is page.

```

607 \def\@tempa{#1}\def\@tempb{page}%
608 \ifx\@tempa\@tempb
609 \global\bypage@true
610 \global\bypstart@false
611 \unless\ifnocritical@%
612 \Xpstart[] [false]%
613 \fi%
614 %

```

If the argument is pstart.

```

615 \else
616 \def\@tempb{pstart}%
617 \ifx\@tempa\@tempb
618 \global\bypage@false
619 \global\bypstart@true
620 \unless\ifnocritical@%
621 \Xpstart%
622 \fi%
623 %

```

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

```

624 \else
625 \def\@tempb{section}
626 \ifx\@tempa\@tempb
627 \global\bypage@false
628 \global\bypstart@false
629 \unless\ifnocritical@%
630 \Xpstart[] [false]%
631 \fi%
632 %

```

In other case, it is an error.

```

633 \else
634 \led@warn@BadLineation
635 \fi
636 \fi
637 \fi
638 \fi}}
639 %

```

## 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.

```

640 \newcount\line@margin%
641 \newcount\line@margin@columns%Only for parallel typesetting
642 \line@margin@columns=\m@ne%
643
644 \newcommand*\linenummargin}[1]{%
645   \l@dgetline@margin{#1}%
646   \ifnum\@l@dttempcntb>\m@ne
647     \ifledRcol
648       \global\line@marginR=\@l@dttempcntb
649       \led@warn@setting@in@rightside{\linenummargin}%
650     \else
651       \global\line@margin=\@l@dttempcntb
652     \fi
653   \fi}}
654
655 \newcommand*\l@dgetline@margin}[1]{%
656   \def\@tempa{#1}\def\@tempb{left}%
657   \ifx\@tempa\@tempb
658     \@l@dttempcntb \z@
659   \else
660     \def\@tempb{right}%
661     \ifx\@tempa\@tempb
662       \@l@dttempcntb \@ne
663     \else
664       \def\@tempb{outer}%
665       \ifx\@tempa\@tempb
666         \@l@dttempcntb \tw@
667       \else
668         \def\@tempb{inner}%
669         \ifx\@tempa\@tempb
670           \@l@dttempcntb \thr@@
671         \else
672           \led@warn@BadLinenummargin
673           \@l@dttempcntb \m@ne
674         \fi
675       \fi
676     \fi
677   \fi}
678
679 %

```

### 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.

```
680 \newcounter{firstlinenum}
681 \setcounter{firstlinenum}{5}
682 \newcounter{linenumincrement}
683 \setcounter{linenumincrement}{5}
684 %
```

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

```
685 \newcounter{firstsublinenum}
686 \setcounter{firstsublinenum}{5}
687 \newcounter{sublinenumincrement}
688 \setcounter{sublinenumincrement}{5}
689
690 %
```

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

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

```

715 }
716 \newcommand*\sublinenumincrement}[1]{%
717   \ifledRcol%
718     \setcounter{sublinenumincrementR}{#1}%
719     \led@warn@setting@in@rightside{\sublinenumincrement}%
720   \else%
721     \setcounter{sublinenumincrement}{#1}%
722   \fi%
723 }
724
725 %

```

## 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.

```

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

```

`\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.

```

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

```

## 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.

```

763 \newcommand*{\linenumberstyle}[1]{\%
764   \def\linenumrep##1{\@nameuse{@#1}{##1}}}
765 \newcommand*{\sublinenumberstyle}[1]{\%
766   \def\sublinenumrep##1{\@nameuse{@#1}{##1}}}
767 %

```

Initialise the number styles to arabic.

```

768 \linenumberstyle{arabic}
769 \let\linenumr@p\linenumrep
770 \sublinenumberstyle{arabic}
771 \let\sublinenumr@p\sublinenumrep
772
773 %

```

## 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.

```

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

```



## 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`.

```
812 \newcount\line@num
813 %
```

`\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.

```
814 \newcount\subline@num
815 %
```

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

`\sublines@true`  
`\sublines@false`

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

```
816 \newif\ifsublines@
817 %
```

`\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.

```

818 \newcount\absline@num
819 %

```

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.

```

820 \newcount\@lock
821 \newcount\sub@lock
822 %

```

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

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

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

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

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

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

The action code  $-1001$  specifies the start of sub-lineation: meaning that, starting with the next line, we should be advancing `\subline@num` at each start-of-line command, rather than `\line@num`.

The action code  $-1002$  specifies the end of sub-lineation. At the next start-of-line, we should clear the sub-line counter and start advancing the line number. The action codes for starting and ending sub-lineation are added to the list by the `\sub@action` macro, as called to implement the `\startsub` and `\endsub` macros.

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

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

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

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

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

`\startlock` and `\endlock` macros.

An action code of  $-5000$  or less sets the current visible line number (either the line number or the sub-line number, whichever is currently being advanced) to a specific positive value. The value of the code is  $-(5000 + n)$ , where  $n$  is the value (always  $\geq 0$ ) assigned to the current line number. Action codes of this type are added to the list by the `\set@line@action` macro, as called to implement the `\advance` 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 `\printlines` already have height arguments, and a  $\text{\TeX}$  macro can take at most nine arguments.

Here are the commands to create these lists:

```
823 \list@create{\line@list}
824 \list@create{\insertlines@list}
825 \list@create{\actionlines@list}
826 \list@create{\actions@list}
827 \list@create{\annot@list}%
828 %
```

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

```
\endpage@num
\endline@num
\endsubline@num
829 \newcount\page@num
830 \newcount\endpage@num
831 \newcount\endline@num
832 \newcount\endsubline@num
833 %
```

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

```
834 \newcount\this@section@page@num%
835 %
```

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

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

The `\ifnoteschanged@` flag is set if such a change in the number of notes is discovered at any point.

```
836 \newif\ifnoteschanged@
837 %
```

`\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@38 \newcommand*\resetprevline@{%
839   \def\do##1{\global\csundef{prevline##1}}%
840   \dolistloop{\@series}%
841 }
842 %
```

`\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@43 \newcommand*\resetprevpage@num{%
844   \def\do##1{%
845     \ifcsdef{prevpage##1@num}{%
846       \global\csname prevpage##1@num\endcsname=\z@%
847       \global\csname prevpage##1@numR\endcsname=\z@%
848     }%
849     {}%
850     \ifcsdef{##1prevpage@num}{%
851       \global\csname ##1prevpage@num\endcsname=\z@%
852       \global\csname ##1prevpage@numR\endcsname=\z@%
853     }%
854     {}%
855     }%
856   \dolistloop{\@series}%
857 }
858 %
```

## 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.

```
859 \newread\@inputcheck
860 \newcommand*\read@linelist[1]{%
```

```

861 \ifledRcol%
862   \list@clearing@regR%
863 \else%
864   \list@clearing@reg%
865 \fi%
866 %

```

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).

```

867 \list@clear{\maxlinesinpar@list}
868 %

```

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.

```

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

```

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.

```

880 \ifledRcol
881   \global\page@numR=\m@ne
882   \ifx\actionlines@listR\empty
883     \gdef\next@actionlineR{1000000}%
884   \else

```

```

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

```

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

```

899 \newcommand*{\list@clearing@reg}{%
900   \list@clear{\line@list}%
901   \list@clear{\insertlines@list}%
902   \list@clear{\actionlines@list}%
903   \list@clear{\actions@list}%
904   \list@clear{\linesinpar@listL}%
905   \list@clear{\linesonpage@listL}%
906   }%
907 %

```

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

```

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

```

This version of `\read@linelist` creates list macros containing data for the entire section, so they could get rather large. It would be no more difficult to read the line-list file incrementally rather than all at once: we could read, at the start of each paragraph, only the commands relating to that paragraph. But this would require that we have two line-lists open at once, one for reading, one for writing, and on systems without version numbers we would have to do some file renaming outside of  $\TeX$  for that to

work. We have retained this slower approach to avoid that sort of hacking about, but have provided the `\pausenumbering` and `\resumenumbering` macros to help you if you run into macro memory limitations (see 5.2.7 p. 20 above).

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

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

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

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

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

```

920 \newcommand{\line@list@version}[1]{%
921   \IfStrEq{#1}{\this@line@list@version}%
922   {}%
923   {\ifledRcol%
924     \led@warn@Obsolete{\jobname.\extensionchars\the\section@num}%
925     \else%
926     \led@warn@Obsolete{\jobname.\extensionchars\the\section@num}%
927     \fi%
928     \endinput%
929   }%
930 }%
931 %

```

`\@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`.

```

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

```

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

```

952   \advance\absline@num \@ne
953     \ifx\next@page@num\relax \else
954       \page@action
955       \let\next@page@num=\relax
956     \fi
957     \ifx\sub@change\relax \else
958       \ifnum\sub@change>\z@
959         \sublines@true
960       \else
961         \sublines@false
962       \fi
963       \sub@action
964       \let\sub@change=\relax
965     \fi
966   %

```

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

```

967   \ifcase\@lock
968     \or
969       \@lock \tw@
970     \or \or
971       \@lock \z@

```

```

972     \fi
973     \ifcase\sub@lock
974         \or
975             \sub@lock \tw@
976         \or \or
977             \sub@lock \z@
978     \fi
979 %

```

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

```

980     \ifsublines@
981         \ifnum\sub@lock<\tw@
982             \advance\subline@num \@ne
983         \fi
984     \else
985         \ifnum\@lock<\tw@
986             \advance\line@num \@ne \subline@num \z@
987         \fi
988     \fi}
989 %
990 %

```

`\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
991 \newcount\last@page@num
992 \last@page@num=-10000
993
994 \newcount\this@section@last@page@num%
995 \this@section@last@page@num=-10000%
996
997 \newcommand*{\@page}[1]{%
998     \ifledRcol
999         \ifnum #1=\last@page@numR
1000     \else
1001         \csnumgdef{\lastabsline@forpageR@the\page@numR}{\the\absline@numR}%
1002         \csnumgdef{\lastline@forpageR@the\page@numR}{\the\line@numR}%
1003         \ifbypage@R
1004             \ifx\l@dchset@num\relax%Not resetting if preceded by a \setlinenum
1005                 \ifboolexpr{%
1006                     bool{resumenumberingR@start}%
1007                     and test {\ifnumequal{\last@page@numR}{-10000}}%
1008                 }%
1009                 {}%
1010                 {%
1011                     \line@numR \z@%
1012                     \subline@numR \z@%
1013                     \global\csdef{reset@lineR@the\numexpr\absline@numR+1\relax @\
the\section@numR}{}%
1014                 }%

```

```

1015     \fi%
1016     \fi
1017     \page@numR=#1%
1018     \this@section@page@numR=#1%
1019     \last@page@numR=#1%
1020     \this@section@last@page@numR=#1%
1021     \def\next@page@numR{#1}%
1022     \gdef\this@section@next@page@numR{#1}%
1023     \fi
1024     \else
1025     \ifnum #1=\last@page@num
1026     \else
1027     \csnumgdef{\lastabsline@forpage@the\page@num}{\the\absline@num}%
1028     \csnumgdef{\lastline@forpage@the\page@num}{\the\line@num}%
1029     \ifbypage@
1030     \ifx\l@dchset@num\relax%Not resetting if preceded by a \setlinenum
1031     \line@num \z@%
1032     \subline@num \z@%
1033     \global\csdef{reset@line\the\numexpr\absline@num+1\relax @the\
section@num}{}%
1034     \fi%
1035     \fi
1036     \page@num=#1%
1037     \global\this@section@page@num=#1%
1038     \last@page@num=#1%
1039     \global\this@section@last@page@num=#1%
1040     \def\next@page@num{#1}%
1041     %

```

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.

```

1042     \gdef\this@section@next@page@num{#1}%
1043     \listxadd{\normal@page@break}{\the\absline@num}
1044     \fi
1045     \fi}
1046     %

```

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

```

1047 \@lopR \newcommand*{\@pend}[1]{}
1048 \@pendR \newcommand*{\@pendR}[1]{}
1049 \@lopL \newcommand*{\@lopL}[1]{}
1050 \@lopR \newcommand*{\@lopR}[1]{}
1051
1052 %

```

`\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.

```

1053 \newcommand*{\sub@on}{\ifsublines@
1054     \let\sub@change=\relax
1055 \else
1056     \def\sub@change{1}%
1057 \fi}
1058 \newcommand*{\sub@off}{\ifsublines@
1059     \def\sub@change{-1}%
1060 \else
1061     \let\sub@change=\relax
1062 \fi}
1063
1064 %

```

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

```

1065
1066 \newcommand*{\@adv}[1]{%
1067     \ifsublines@
1068     \ifledRcol
1069         \advance\subline@numR by #1\relax
1070         \ifnum\subline@numR<\z@
1071             \led@warn@BadAdvancelineSubline
1072             \subline@numR \z@
1073         \fi
1074     \else
1075         \advance\subline@num by #1\relax
1076         \ifnum\subline@num<\z@
1077             \led@warn@BadAdvancelineSubline
1078             \subline@num \z@
1079         \fi
1080     \fi
1081 \else
1082     \ifledRcol
1083         \advance\line@numR by #1\relax
1084         \ifnum\line@numR<\z@
1085             \led@warn@BadAdvancelineLine
1086             \line@numR \z@
1087         \fi
1088     \else
1089         \advance\line@num by #1\relax
1090         \ifnum\line@num<\z@
1091             \led@warn@BadAdvancelineLine
1092             \line@num \z@
1093         \fi
1094     \fi
1095 \fi

```

```

1096 \set@line@action}
1097
1098 %

```

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

```

1099
1100 \newcommand*{\@set}[1]{%
1101   \ifledRcol
1102     \ifsublines@
1103       \subline@numR=#1\relax
1104     \else
1105       \line@numR=#1\relax
1106     \fi
1107     \set@line@action
1108   \else
1109     \ifsublines@
1110       \subline@num=#1\relax
1111     \else
1112       \line@num=#1\relax
1113     \fi
1114     \set@line@action
1115   \fi}
1116
1117 %

```

**\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.

```

1118
1119 \newcommand*{\l@d@set}[1]{%
1120   \ifledRcol
1121     \line@numR=#1\relax
1122     \advance\line@numR \@ne
1123     \def\l@dchset@num{#1}
1124   \else
1125     \line@num=#1\relax
1126     \advance\line@num \@ne
1127     \def\l@dchset@num{#1}
1128   \fi}
1129 \let\l@dchset@num\relax
1130
1131 %

```

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

```

1132 \newcommand*{\page@action}{%
1133   \ifledRcol
1134     \xright@appenditem{\the\absline@numR}\to\actionlines@listR
1135     \xright@appenditem{\next@page@numR}\to\actions@listR
1136   \else
1137     \xright@appenditem{\the\absline@num}\to\actionlines@list
1138     \xright@appenditem{\next@page@num}\to\actions@list
1139   \fi}
1140 %
1141

```

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

```

1142 \newcommand*{\set@line@action}{%
1143   \ifledRcol
1144     \xright@appenditem{\the\absline@numR}\to\actionlines@listR
1145     \ifsublines@
1146       \@l@tempcnta=-\subline@numR
1147     \else
1148       \@l@tempcnta=-\line@numR
1149     \fi
1150     \advance\@l@tempcnta by -5000\relax
1151     \xright@appenditem{\the\@l@tempcnta}\to\actions@listR
1152   \else
1153     \xright@appenditem{\the\absline@num}\to\actionlines@list
1154     \ifsublines@
1155       \@l@tempcnta=-\subline@num
1156     \else
1157       \@l@tempcnta=-\line@num
1158     \fi
1159     \advance\@l@tempcnta by -5000\relax
1160     \xright@appenditem{\the\@l@tempcnta}\to\actions@list
1161   \fi}
1162 %
1163

```

**\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.

```

1164 \newcommand*{\sub@action}{%
1165   \ifledRcol
1166     \xright@appenditem{\the\absline@numR}\to\actionlines@listR
1167     \ifsublines@
1168       \xright@appenditem{-1001}\to\actions@listR
1169     \else
1170       \xright@appenditem{-1002}\to\actions@listR
1171     \fi
1172   \else
1173

```

```

1174 \xright@appenditem{\the\absline@num}\to\actionlines@list
1175 \ifsublines@
1176 \xright@appenditem{-1001}\to\actions@list
1177 \else
1178 \xright@appenditem{-1002}\to\actions@list
1179 \fi
1180 \fi}
1181 %

```

`\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.

```

1182 \newcommand*{\lock@on}{\futurelet\next\do@lockon}
1183
1184 \newcommand*{\do@lockon}{%
1185 \ifx\next\lock@off
1186 \global\let\lock@off=\skip@lockoff
1187 \else
1188 \ifledRcol
1189 \do@lockonR
1190 \else
1191 \do@lockonL
1192 \fi
1193 \fi}
1194
1195
1196 \newcommand*{\do@lockonL}{%
1197 \xright@appenditem{\the\absline@num}\to\actionlines@list
1198 \ifsublines@
1199 \xright@appenditem{-1005}\to\actions@list
1200 \ifnum\sub@lock=\z@
1201 \sub@lock \@ne
1202 \else
1203 \ifnum\sub@lock=\thr@@
1204 \sub@lock \@ne
1205 \fi
1206 \fi
1207 \else
1208 \xright@appenditem{-1003}\to\actions@list
1209 \ifnum\@lock=\z@
1210 \@lock \@ne
1211 \else
1212 \ifnum\@lock=\thr@@
1213 \@lock \@ne

```

```

1214     \fi
1215     \fi
1216   \fi}
1217
1218 %

```

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

```

\do@lockoff
\do@lockoffL
\skip@lockoff
1219 \newcommand*{\do@lockoffL}{%
1220   \xright@appenditem{\the\absline@num}\to\actionlines@list
1221   \ifsublines@
1222     \xright@appenditem{-1006}\to\actions@list
1223     \ifnum\sub@lock=\tw@
1224       \sub@lock \thr@@
1225     \else
1226       \sub@lock \z@
1227     \fi
1228   \else
1229     \xright@appenditem{-1004}\to\actions@list
1230     \ifnum\@lock=\tw@
1231       \@lock \thr@@
1232     \else
1233       \@lock \z@
1234     \fi
1235   \fi}
1236
1237 \newcommand*{\do@lockoff}{%
1238   \reset@current@annot%
1239   \ifledRcol
1240     \do@lockoffR
1241   \else
1242     \do@lockoffL
1243   \fi}
1244 \newcommand*{\skip@lockoff}{\global\let\lock@off=\do@lockoff}
1245 \global\let\lock@off=\do@lockoff
1246
1247 %

```

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

```

1248 \newcommand*{\n@num}{%
1249   \ifledRcol%
1250     \xright@appenditem{\the\absline@numR}\to\actionlines@listR
1251     \xright@appenditem{-1007}\to\actions@listR
1252   \else%
1253     \xright@appenditem{\the\absline@num}\to\actionlines@list%
1254     \xright@appenditem{-1007}\to\actions@list%
1255   \fi%
1256 }%

```



1257  
1258 %

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

```
1259 \newcommand*{\n@num@stanza}{%
1260   \ifledRcol%
1261     \xright@appenditem{\the\absline@numR}\to\actionlines@listR%
1262     \xright@appenditem{-1008}\to\actions@listR%
1263   \else%
1264     \xright@appenditem{\the\absline@num}\to\actionlines@list%%
1265     \xright@appenditem{-1008}\to\actions@list%
1266   \fi%
1267 }
1268 %
```

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

```
\hidenumberingonleftpage
\hidenumberingonrightpage
1269 \newif\ifl@dhidenumber
1270 \newcommand*{\hidenumbering}{%
1271   \ifledRcol%
1272     \write\linenum@outR{\string\hide@num}%
1273   \else%
1274     \write\linenum@out{\string\hide@num}%
1275   \fi%
1276 }%
1277 \newcommand*{\hide@num}{%
1278   \ifledRcol%
1279     \xright@appenditem{\the\absline@numR}\to\actionlines@listR%
1280     \xright@appenditem{-1009}\to\actions@listR%
1281   \else%
1282     \xright@appenditem{\the\absline@num}\to\actionlines@list%%
1283     \xright@appenditem{-1009}\to\actions@list%
1284   \fi%
1285 }
1286 \newcommand*{\hidenumberingonleftpage}{%
1287   \ifledRcol%
1288     \write\linenum@outR{\string\hide@num@left}%
1289   \else%
1290     \write\linenum@out{\string\hide@num@left}%
1291   \fi%
1292 }%
1293
1294 \newcommand*{\hide@num@left}{%
1295   \ifledRcol%
1296     \ifodd\page@numR\else%
1297       \xright@appenditem{\the\absline@numR}\to\actionlines@listR%
```

```

1298 \xright@appenditem{-1009}\to\actions@listR%
1299 \fi%
1300 \else%
1301 \ifodd\page@num\else%
1302 \xright@appenditem{\the\absline@num}\to\actionlines@list%%
1303 \xright@appenditem{-1009}\to\actions@list%
1304 \fi%
1305 \fi%
1306 }%
1307
1308 \newcommand*{\hidenumberingonrightpage}{%
1309 \ifledRcol%
1310 \write\linenum@outR{\string\hide@num@right}%
1311 \else%
1312 \write\linenum@out{\string\hide@num@right}%
1313 \fi%
1314 }%
1315
1316 \newcommand*{\hide@num@right}{%
1317 \ifledRcol%
1318 \ifodd\page@numR%
1319 \xright@appenditem{\the\absline@numR}\to\actionlines@listR%
1320 \xright@appenditem{-1009}\to\actions@listR%
1321 \fi%
1322 \else%
1323 \ifodd\page@num%
1324 \xright@appenditem{\the\absline@num}\to\actionlines@list%%
1325 \xright@appenditem{-1009}\to\actions@list%
1326 \fi%
1327 \fi%
1328 }%
1329
1330 %

```

**\@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.

```

1331 \newcount\insert@count
1332 %

```

- #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.

```

1333 \newcommand*\dummy@ref}[2]{#2}
1334 %

```

**\@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.

```

1335 \newcommand*\@ref}[2]{%
1336   \ifledRcol%
1337     \@ref@regR{#1}{#2}%
1338   \else%
1339     \@ref@reg{#1}{#2}%
1340   \fi%
1341 }%
1342 \newcommand*\@ref@reg}[2]{%
1343   \global\insert@count=#1\relax
1344   \global\advance\@edtext@level by 1%
1345   \loop\ifnum\insert@count>\z@
1346     \xright@appenditem{\the\absline@num}\to\insertlines@list
1347     \global\advance\insert@count \m@ne
1348   \repeat
1349 %

```

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.

```

1350 \begingroup
1351   \let\@ref=\dummy@ref
1352   \let\@lopL\@gobble
1353   \let\page@action=\relax
1354   \let\sub@action=\relax
1355   \let\set@line@action=\relax
1356   \let\@lab=\relax
1357   \let\@lemma=\relax%
1358   \let\@sw\@gobblethree%
1359   \let\store@annot@to@absline\@gobble%
1360   #2
1361   \global\endpage@num=\page@num
1362   \global\endline@num=\line@num
1363   \global\endsubline@num=\subline@num
1364   \global\let\endcurrent@annot=\current@annot%
1365 \endgroup
1366 %

```

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

```

1367 \xright@appenditem%
1368   {\the\page@num|\the\line@num}%
1369   \ifsublines@ \the\subline@num \else 0\fi%

```

```

1370 \the\endpage@num|\the\endline@num|
1371 \ifsublines@ \the\endsubline@num \else 0\fi}\to\line@list
1372 \xright@appenditem%
1373 {\current@annot|\endcurrent@annot}\to\annot@list%
1374 %

```

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

```

1375 \@ref@reg@parse{#2}%
1376 %

```

Decrease edtext level counter.

```

1377 \global\advance\@edtext@level by -1%
1378 }
1379 %

```

**`\@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.

```

1380 \newcommand{\@ref@reg@parse}[1]{%
1381 \expandafter\list@create\expandafter{\csname sw@list@edtext@tmp@\the\
1382 @edtext@level\endcsname}%
1383 \providebool{lemmacommand@\the\@edtext@level}%
1384 \boolfalse{lemmacommand@\the\@edtext@level}%
1385 }

```

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

```

1385 #1%
1386 %

```

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.

```

1387 \ifnum\@edtext@level>0%
1388 \def\create@this@edtext@level{\expandafter\list@create\expandafter{\
1389 csname sw@list@edtext@\the\@edtext@level\endcsname}}%
1390 \ifcsundef{sw@list@edtext@\the\@edtext@level}{\create@this@edtext@level}%
1391 }{}%
1392 \letcs{\@tmp}{sw@list@edtext@\the\@edtext@level}%
1393 \letcs{\@tmpp}{sw@list@edtext@tmp@\the\@edtext@level}%
1394 \xright@appenditem{\expandonce\@tmpp}\to\@tmp%
1395 \global\cslet{sw@list@edtext@\the\@edtext@level}{\@tmp}%
1396 \fi%
1397 }
1398 %

```

`\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.

```

1399 \newcommand{\@ref@later}[1]{%
1400   \global\advance\@edtext@level by \@ne%
1401   \ifledRcol%
1402     \@ref@reg@parseR{#1}%
1403   \else%
1404     \@ref@reg@parse{#1}%
1405   \fi%
1406   \global\advance\@edtext@level by -\@ne%
1407 }%
1408 %

```

## 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`.

```

1409 \newwrite\linenum@out
1410 %

```

`\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.

```

1411 \newif\iffirst@linenum@out
1412 \iffirst@linenum@out@true
1413 %

```

`\this@line@list@version` The commands allowed in the line-list file and their arguments can change between two versions 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 an older version, that means the commands used inside are deprecated, and we can't use them.

```

1414 \newcommand{\this@line@list@version}{7}%
1415 %

```

`\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.

```
1416 \let\next@line@list@stuff\relax%
1417 \newcommand*{\line@list@stuff}[1]{%
1418 %
```

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

```
1419 \global\newtoggle{notfirstrun@#1}%
1420 \ifFileExists{\l@auxdir#1}%
1421 {\global\toggletrue{notfirstrun@#1}}%
1422 {\global\togglefalse{notfirstrun@#1}}%
1423 %
```

A internal hook (not used yet).

```
1424 \next@line@list@stuff%
1425 \global\let\next@line@list@stuff\relax%
1426 %
```

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

```
1427 \read@linelist{#1}%
1428 %
```

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  $n$ .

```
1429 \iffirst@linenum@out@
1430 \global\first@linenum@out@false%
1431 \immediate\openout\linenum@out=\l@auxdir#1\relax%
1432 \immediate\write\linenum@out{\string\line@list@version{\
this@line@list@version}}%
1433 \ifl@dpaging%
1434 \immediate\write\linenum@out{\string\@par@sync@option{\
@par@this@sync@option}}%
1435 \fi%
1436 \else
1437 %
```

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.

```
1438 \if@minipage%
1439 \leavevmode%
1440 \fi%
1441 \closeout\linenum@out%
1442 \openout\linenum@out=\l@auxdir#1\relax%
```

```

1443 \write\linenum@out{\string\line@list@version{\this@line@list@version}}
1444 %
1445 \ifl@dpaging%
1446 \write\linenum@out{\string\@par@sync@option{\@par@this@sync@option}}
1447 %
1448 \fi%
1449 \fi%
1450 }%

```

`\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 `\sameparallelpagenunder` 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.

```

1451 \newcommand*{\new@line}{%
1452 \ifnumberline%
1453 \IfStrEq{\led@pb@setting}{after}%
1454 {\xifinlist{\the\absline@num}{\l@prev@nopb}%
1455 {\xifinlist{\the\absline@num}{\normal@page@break}%
1456 {\numgdef{\@next@page}{\c@par@page+\@ne}%
1457 \write\linenum@out{\string\@nl[\@next@page][\@next@page]}}%
1458 }%
1459 {\write\linenum@out{\string\@nl[\the\c@par@page][\thepage]}}%
1460 }%
1461 {\write\linenum@out{\string\@nl[\the\c@par@page][\thepage]}}}%
1462 {}%
1463 \IfStrEq{\led@pb@setting}{before}%
1464 {\numgdef{\next@absline}{\the\absline@num+\@ne}%
1465 \xifinlist{\next@absline}{\l@prev@nopb}%
1466 {\xifinlist{\the\absline@num}{\normal@page@break}%
1467 {\numgdef{\nc@page}{\c@par@page+\@ne}%
1468 \write\linenum@out{\string\@nl[\nc@page][\nc@page]}}%
1469 }%
1470 {\write\linenum@out{\string\@nl[\the\c@par@page][\thepage]}}%
1471 }%

```

```

1472     {\write\linenum@out{\string\@nl[\the\c@par@page][\thepage]}}%
1473     }%
1474     }%
1475     \IfStrEqCase{\led@pb@setting}%
1476     {%
1477     {before}{\relax}%
1478     {after}{\relax}%
1479     }[%
1480     \write\linenum@out{\string\@nl[\the\c@par@page][\thepage]}}%
1481     ]%
1482     \fi%
1483 }
1484
1485 \newcommand{\l@the\c@page}{%
1486   \ifboolexpr{%
1487     bool{sameparallelpagenumber}%
1488     or bool{prevpgnotnumbered}%
1489   }%
1490   {\the\c@par@page}%
1491   {\the\c@page}%
1492 }%
1493 %

```

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

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

**\flag@end**

```

1494 \newif\if@noneed@Footnote%
1495
1496 \newcommand*{\flag@start}{%
1497   \ifledRcol%
1498     \edef\next{\write\linenum@outR{%
1499       \string\@ref[\the\insert@countR][\thepage]}}%
1500     \next%
1501     \ifnum\insert@countR<1%
1502       \if@noneed@Footnote\else%
1503         \led@err@EdtextWithoutFootnote%
1504       \fi%
1505     \fi%
1506   \else%
1507     \edef\next{\write\linenum@out{%
1508       \string\@ref[\the\insert@count][\thepage]}}%
1509     \next%
1510     \ifnum\insert@count<1%
1511       \if@noneed@Footnote\else%
1512         \led@err@EdtextWithoutFootnote%
1513       \fi%

```



```

1514 \fi%
1515 \fi}%
1516
1517 \newcommand*{\flag@end}{%
1518 \ifledRcol%
1519 \write\linenum@outR{}}%
1520 \else%
1521 \write\linenum@out{}}%
1522 \fi}%
1523
1524
1525 %

```

`\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.

```

1526 \newcommand{\flag@start@RTL}{%
1527 \edlabel{edtext:start:\csuse{thisedtext@the\@edtext@level}}%
1528 \IfStrEq{\xabslineref{edtext:start:\csuse{thisedtext@the\@edtext@level}}}%
1529 {000}%
1530 {}%
1531 {%
1532 \ifnumequal%
1533 {\xabslineref{edtext:start:\csuse{thisedtext@the\@edtext@level}}}%
1534 {\xabslineref{edtext:end:\csuse{thisedtext@the\@edtext@level}}}%
1535 {\flag@end}%
1536 {\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.

```

1537 }%
1538 }%
1539
1540 \newcommand{\flag@end@RTL}{%
1541   \edlabel{edtext:end:\csuse{thisedtext@the\@edtext@level}}%
1542   \IfStrEq{\xabslineref{edtext:start:\csuse{thisedtext@the\@edtext@level
1543     {000}}%
1544     }%
1545     {%
1546       \ifnumequal%
1547         {\xabslineref{edtext:start:\csuse{thisedtext@the\@edtext@level}}}%
1548         {\xabslineref{edtext:end:\csuse{thisedtext@the\@edtext@level}}}%
1549         {\flag@start}%
1550         {\flag@end}%
1551     }%
1552 }%
1553 %

```

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

```

1554 \newcommand*{\flag@start@later}{%
1555   \ifledRcol%
1556     \write\linenum@outR{\string\@ref@later []}%
1557   \else%
1558     \write\linenum@out{\string\@ref@later []}%
1559   \fi%
1560 }%
1561 \newcommand{\flag@end@later}{%
1562   \ifledRcol%
1563     \write\linenum@outR{[]}%
1564   \else%
1565     \write\linenum@out{[]}%
1566   \fi%
1567 }
1568 %

```

`\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.

```

1569
1570
1571 \newcommand*{\startsub}{\dimen0\lastskip
1572 \ifdim\dimen0>0pt \unskip \fi
1573 \ifledRcol \write\linenum@outR{\string\sub@on}%
1574 \else \write\linenum@out{\string\sub@on}%
1575 \fi
1576 \ifdim\dimen0>0pt \hskip\dimen0 \fi}
1577 \def\endsub{\dimen0\lastskip
1578 \ifdim\dimen0>0pt \unskip \fi
1579 \ifledRcol \write\linenum@outR{\string\sub@off}%
1580 \else \write\linenum@out{\string\sub@off}%
1581 \fi
1582 \ifdim\dimen0>0pt \hskip\dimen0 \fi}
1583
1584 %

```

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

```

1585 \newcommand*{\advanceline}[1]{\leavevmode%
1586 \ifledRcol \write\linenum@outR{\string\@adv[#1]}%
1587 \else \write\linenum@out{\string\@adv[#1]}%
1588 \fi%
1589 }
1590 %

```

**\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.

```

1591
1592 \newcommand*{\setline}[1]{%
1593 \leavevmode%
1594 \ifnum#1<\z@
1595 \led@warn@BadSetline%
1596 \else%
1597 \ifledRcol%
1598 \write\linenum@outR{\string\@set[#1]}%
1599 \else%
1600 \write\linenum@out{\string\@set[#1]}%
1601 \fi%
1602 \fi%
1603 }%
1604
1605 %

```

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

```

1606 \newcommand*{\setlinenum}[1]{%
1607   \ifnum#1<\z@
1608     \led@warn@BadSetlinenum
1609   \else
1610     \ifledRcol \write\linenum@outR{\string\l@d@set[#1]}
1611     \else      \write\linenum@out{\string\l@d@set[#1]} \fi
1612   \fi}
1613
1614
1615 %

```

`\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.

```

1616 \newcommand*{\startlock}{%
1617   \ifledRcol \write\linenum@outR{\string\lock@on}%
1618   \else      \write\linenum@out{\string\lock@on}%
1619   \fi}
1620
1621 \def\endlock{%
1622   \ifledRcol \write\linenum@outR{\string\lock@off}%
1623   \else      \write\linenum@out{\string\lock@off}%
1624   \fi}
1625 %

```

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

`\ifl@dskipversenumber`

`\l@dskipnumbertrue`

`\l@dskipnumberfalse`

`\skipnumbering`

```

1626 \newif\ifl@dskipnumber
1627 \newif\ifl@dskipversenumber%
1628 \newcommand*{\skipnumbering}{%
1629   \leavevmode%
1630   \ifledRcol%
1631     \ifinstanza%
1632       \write\linenum@outR{\string\n@num@stanza}%
1633     \else%
1634       \write\linenum@outR{\string\n@num}%
1635     \fi%
1636     \advanceline{-1}%
1637   \else%
1638     \ifinstanza%
1639       \write\linenum@out{\string\n@num@stanza}%
1640     \else%
1641       \write\linenum@out{\string\n@num}%
1642     \fi%
1643     \advanceline{-1}%
1644   \fi%
1645 }%
1646

```

1647 %

## 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. 157). 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.

```
1648 \list@create{\end@lemmas}
1649 %
```

`\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.

```
1650 \newcommand{\dummy@edtext}[2]{#1}
1651 %
```

`\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`.

```
1652 \newcommand{\dummy@edtext@showlemma}[2]{\showlemma{#1}}%
1653 %
```

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  $\langle X \rangle$ .

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

```

1654 \newcommand*{\no@expands}{%
1655   \let\select@lemmafont=0%
1656   \let\startsub=\relax   \let\endsub=\relax
1657   \let\startlock=\relax  \let\endlock=\relax
1658   \let\edlabel=\@gobble
1659   \let\setline=\@gobble  \let\advanceline=\@gobble
1660   \let\sameword\sameword@inedtext%
1661   \let\edtext=\dummy@edtext

```

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

```

1662 \let\edindex\dummy@edindex%
1663 \l@dtabnoexpands
1664 \let\linenumannotation=\@gobble%
1665 \morenoexpands}
1666 \let\morenoexpands=\relax
1667
1668 %

```

**\@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.

```

1669 \newcommand{\@tag}{}
1670 %

```

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

```

1671 \newcount\@edtext@level%
1672 \@edtext@level=0%
1673 %

```

**\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.

```

1674 \newif\if@edtext@secondarg@%
1675 %

```

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

```

1676 \newcounter{edtext}
1677 \renewcommand{\theedtext}{edtxt@arabic{edtext}}%
1678 %

```

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

```

1679 \newcommand{\edtext}[2]{\leavevmode%
1680 %

```

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

```

1681 \ifnumberedpar@%
1682 %

```

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

```

1683 \@check@edtext@args{#1}{#2}%
1684 %

```

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



```

1685 \global\advance\@edtext@level by 1%
1686 %

```

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

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

```

1687 \stepcounter{edtext}%
1688 \csxdef{thisedtext@the\@edtext@level}{\theedtext}%
1689 %

```

By default, we do not use `\lemma`

```

1690 \global\@lemmacommand@false%
1691 %

```

```

1692 \begingroup%
1693 %

```

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`.

```

1694 \ifledRcol%
1695 \ifcsvoid{sw@list@edtextR@the\@edtext@level}%
1696 {\global\let\sw@inthisedtext\empty}%
1697 {\expandafter\glp\csname sw@list@edtextR@the\@
@edtext@level\endcsname\to\sw@inthisedtext}%
1698 \else%
1699 \ifcsvoid{sw@list@edtext@the\@edtext@level}%
1700 {\global\let\sw@inthisedtext\empty}%
1701 {\expandafter\glp\csname sw@list@edtext@the\@edtext@level
\endcsname\to\sw@inthisedtext}%
1702 \fi%
1703 %

```

**`\@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.

```

1704 \global\renewcommand{\@tag}{%
1705 \no@expands #1%
1706 }%
1707 %

```

**`\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`.

```

1708 \set@line%
1709 %

```

`\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`.

```

1710 \ifledRcol \global\insert@countR \z@%
1711 \else      \global\insert@count \z@ \fi%
1712 %

```

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.

```

1713 \@edtext@secondarg@true%
1714 \ignorespaces #2\relax%
1715 \@edtext@secondarg@false%
1716 %

```

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.

```

1717 \if@RTL%
1718 \flag@start@RTL%
1719 \else%
1720 \flag@start%
1721 \fi%
1722 %

```

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

```

1723 \if@lemmacommand@%
1724 \ifledRcol%
1725 \write\linenum@outR{\string\@lemma}%
1726 \else%
1727 \write\linenum@out{\string\@lemma}%
1728 \fi%
1729 \fi%
1730 %

```

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.

```

1731 \endgroup%
1732 \ifdef{\hypertarget}%
1733 {%
1734 \Hy@raisedlink@left{\hypertarget{\csuse{thisedtext@the\
@edtext@level}:start}{}}}%
1735 \showlemma{#1}%
1736 \Hy@raisedlink{\hypertarget{\csuse{thisedtext@the\@edtext@level}:
end}{}}}%
1737 }%
1738 {%
1739 \showlemma{#1}%
1740 }%
1741 %

```

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.

```

1742 \ifx\end@lemmas\empty \else%
1743 \gl@p\end@lemmas\to\x@lemma%
1744 \x@lemma%
1745 \global\let\x@lemma=\relax%
1746 \fi%
1747 \if@RTL%
1748 \flag@end@RTL%
1749 \else%
1750 \flag@end%
1751 \fi%
1752 %

```

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`.

```

1753 \global\@noneed@Footnotefalse%
1754 \global\advance\@edtext@level by -1%
1755 \global\@lemmacommand@false%
1756 %

```

We also reset `\@beforeinsertofthisedtext`

```

1757 \global\let\@beforeinsertofthisedtext\relax%
1758 %

```

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

```

1759 \else%
1760 \showlemma{#1} (\textbf{\textsc{Edtext outside numbered paragraph}})\
    led@err@edtextoutsidepstart%
1761 \fi%
1762 }%
1763
1764
1765 %

```

**\@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.

```

1766 \newcommand{\@check@edtext@args}[2]{%
1767 \begingroup%
1768 \noexpandarg%
1769 \IfSubStr{#1}{\pstart}{\led@err@PstartInEdtext{first}}{}}%
1770 \IfSubStr{#1}{\pend}{\led@err@PendInEdtext{first}}{}}%
1771 \IfSubStr{#2}{\pstart}{\led@err@PstartInEdtext{second}}{}}%
1772 \IfSubStr{#2}{\pend}{\led@err@PendInEdtext{second}}{}}%
1773 \endgroup%
1774 }%
1775 %
1776 %

```

**\@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`.

```

1777 \let\@beforeinsertofthisedtext\relax
1778 %

```

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

```

1779 \newif\ifnumberline
1780 \numberlinetrue
1781 %

```

**\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.

```

1782 \newcommand*{\set@line}{%
1783 \ifledRcol

```

```

1784 \ifx\line@listR\empty
1785 \global\noteschanged@true
1786 \xdef\l@d@nums{000|000|000|000|000|000|\edfont@info}%
1787 \else
1788 \glp\line@listR\to\@tempb
1789 \xdef\l@d@nums{\@tempb|\edfont@info}%
1790 \global\let\@tempb=\undefined
1791 \fi
1792 \ifx\annot@listR\empty%
1793 \xdef\l@current@annotR{||}%
1794 \else%
1795 \glp\annot@listR\to\@tempb%
1796 \xdef\l@current@annotR{\@tempb}%
1797 \global\let\@tempb=\undefined%
1798 \fi%
1799 \else
1800 \ifx\line@list\empty
1801 \global\noteschanged@true
1802 \xdef\l@d@nums{000|000|000|000|000|000|\edfont@info}%
1803 \else
1804 \glp\line@list\to\@tempb
1805 \xdef\l@d@nums{\@tempb|\edfont@info}%
1806 \global\let\@tempb=\undefined
1807 \fi
1808 \ifx\annot@list\empty%
1809 \xdef\l@current@annot{||}%
1810 \else%
1811 \glp\annot@list\to\@tempb%
1812 \xdef\l@current@annot{\@tempb}%
1813 \global\let\@tempb=\undefined%
1814 \fi%
1815 \fi}
1816
1817 %

```

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

```

1818 \newcommand*{\edfont@info}{\f@encoding/\f@family/\f@series/\f@shape}
1819
1820 %

```

## 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. 137).

```

1821 \newcommand*{\lemma}[1]{%
1822 \global\@lemmacommand@true%

```

```

1823 \global\renewcommand{\@tag}{%
1824 \no@expands #1%
1825 }%
1826 \ignorespaces%
1827 }%
1828 %

```

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

```

1829 \newcommand{\@lemma}{%
1830 \booltrue{lemmacommand@the\@edtext@level}%
1831 }%
1832 %

```

**\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.

```

1833 \newif\if@lemmacommand@%
1834 %

```

### 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. 106): 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.

```

1835 \newcommand*{\linenum}[1]{%
1836 \xdef\@tempa{#1|}|}|}|}|noexpand\\l@d@nums}%
1837 \global\let\l@d@nums=\empty
1838 \expandafter\line@set\@tempa|\\ignorespaces}
1839 %

```

**\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.

```

1840 \def\line@set#1|#2\\#3|#4\\{%
1841 \gdef\@tempb{#1}%

```

```

1842 \ifx\@tempb\empty
1843   \l@d@add{#3}%
1844 \else
1845   \l@d@add{#1}%
1846 \fi
1847 \gdef\@tempb{#4}%
1848 \ifx\@tempb\empty\else
1849   \l@d@add{|\}\line@set#2\|#4\}%
1850 \fi}
1851 %

```

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

```

1852 \newcommand{\l@d@add}[1]{\xdef\l@d@nums{\l@d@nums#1}}
1853
1854 %

```

`\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.

```

1855 \newcommand*{\lineannot}[1]{%
1856   \lineannot@set#1|}%
1857 }%
1858 \def\lineannot@set#1|#2|{%
1859   \expandafter\parse@annot#1|#2|}%
1860   \IfStrEq{#1}{}%
1861     {\let\@tempa\annot@start}%
1862     {\def\@tempa{#1}}}%
1863   \IfStrEq{#2}{}%
1864     {\let\@tempb\annot@start}%
1865     {\def\@tempb{#2}}}%
1866   \xdef\l@current@annot{\@tempa|\@tempb}%
1867 }%
1868 %

```

## VI.4 Lemma disambiguation

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

The general mechanism is the following:

- **At the first run**, each `\sameword` command increments an `etoolbox` counter the name of which contains the argument of the `\sameword` commands.

- Then this counter, associated with the argument of `\sameword` is stored with the `\@sw` command in the auxiliary file of the current `reledmac` section (the .1, .2... file).
- **When this auxiliary file is read at the second run**, different operations are achieved:

1. Get the rank of each `\sameword` in a line (relative rank) from the rank of each `\sameword` in all the numbered section (absolute rank):

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

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

2. Create lists of lists of `\sameword` by depth of `\edtext`. That is: create a list for `\edtexts` of level 1, a list for `\edtexts` of level 2, a list for `\edtexts` of level 3 etc. For each `\edtext` in these lists, we store all of the relative ranks of `\saweword` which are called as lemma information. That is: 1) either called in the first argument of `\sameword`, or, 2) called in the `\lemma` macro of the second argument of `\sameword` AND marked by the optional argument of `\saweword` in first argument of `\edtext`.

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

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



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

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

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

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

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

After theory, implementation.

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

```
1869 \newcommand{\get@sw@txt}[1]{%
1870   \begingroup%
1871   \swnoexpands%
1872   %.
```

Using case sensibility option.

```
1873   \ifsw@caseinsensitive%
1874     \def\@tmpa##1{\lowercase{##1}}%
1875   \else%
1876     \def\@tmpa##1{##1}%
1877   \fi%
1878   %
```

And now, define `\sw@txt`.

```
1879   \ifxetex%
1880     \@tmpa{\xdef\sw@txt{#1}}%
1881   \else%
1882     \@tmpa{\expandafter\xdef\expandafter\sw@txt\expandafter{\detokenize
1883     {#1}}}%
1884   \fi%
1885   \endgroup%
1886   %
```

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}{%
1888   \let\sameword\l@secondmandarg% Allow to have nested \sameword
1889   \let\emph\@firstofone%
1890   \let\textit\@firstofone%
1891   \let\textbf\@firstofone%
1892   \let\textsc\@firstofone%
1893   \let\framebox\@firstofone%
1894   \let\edtext\dummy@edtext%
1895   \RenewExpandableDocumentCommand{\edindex}{om}{}%
1896   \ifdefined\index%
1897     \RenewExpandableDocumentCommand{\index}{om}{}%
1898   \fi%
1899   \let\selectlanguage\@gobble%
1900   \let\foreignlanguage\@secondoftwo%
```

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

```

1901 \ifdefined\xpg@loaded%
1902 \renewcommand\do[1]{%
1903 \expandafter\RenewExpandableDocumentCommand\expandafter{\csname
text##1\endcsname}{om}{###2}%
1904 }%
1905 \expandafter\docsvlist\expandafter{\xpg@loaded}%
1906 \fi%
1907 }%
1908 %

```

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

```

1909 \newcommandx{\sameword}[2][1,usedefault]{%
1910 \leavevmode%
1911 \get@sw@txt{#2}%
1912 %

```

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

```

1913 \unless\ifledRcol%
1914 \csnumgdef{sw@\sw@txt}{\csuse{sw@\sw@txt}+\@ne}%
1915 %

```

Then, write its value to the numbered file.

```

1916 \protected@write\linenum@out{\string\sw{\sw@txt}{\csuse{sw@\sw@txt}
}}{#1}}%
1917 %

```

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

```

1918 \else%
1919 \csnumgdef{sw@\sw@txt}{\csuse{sw@\sw@txt}+\@ne}%
1920 \protected@write\linenum@outR{\string\sw{\sw@txt}{\csuse{sw@\sw@txt}
}}{#1}}%
1921 \fi%
1922 %

```

And print the word.

```

1923 #2%
1924 }%
1925 %

```

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

```

\if@addsw% \newif\if@addsw%
1927 %

```

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

```

1928 \newcommand{\@sw}[3]{%
1929   \get@sw@txt{#1}%
1930   \unless\ifledRcol%
1931   %

```

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

```

1932   \csxdef{sw@sw@txt @\the\absline@num @\the\section@num}{#2}%
1933   %

```

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

```

1934   \numdef{\prev@line}{\the\absline@num-1}%
1935   \ifcsundef{sw@sw@txt @\prev@line @\the\section@num}{%
1936     \csnumgdef{sw@sw@txt @\prev@line @\the\section@num}{#2-1}%
1937   }{}%
1938   %

```

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

```

1939   \numdef{\the@sw}{#2-\csuse{sw@sw@txt @\prev@line @\the\section@num}}%
1940   %

```

And do the same thing for the right side.

```

1941   \else%
1942     \csxdef{sw@sw@txt @\the\absline@numR @\the\section@numR @R}{#2}%
1943     \numdef{\prev@line}{\the\absline@numR-1}%
1944     \ifcsundef{sw@sw@txt @\prev@line @\the\section@numR @R}{%
1945       \csnumgdef{sw@sw@txt @\prev@line @\the\section@numR @R}{#2-1}%
1946     }{}%
1947     \numdef{\the@sw}{#2-\csuse{sw@sw@txt @\prev@line @\the\section@numR @R}}%
1948   \fi%
1949   %

```

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

```

1950   \@tempcnta=\@edtext@level
1951   \@whilenum{\@tempcnta>0}\do{%
1952     \ifcsdef{sw@list@edtext@tmp@\the\@tempcnta}%
1953     {%
1954       \@addswfalse%
1955       \notbool{lemmacommand@\the\@tempcnta}%
1956       {\@addswtrue}%
1957       {\IfStrEq{#3}{inlemma}%
1958        {\@addswtrue}%
1959       }%
1960       \def\do##1{%
1961         \ifnumequal{##1}{\the\@tempcnta}%
1962         {\@addswtrue\listbreak}%
1963         }%
1964     }%

```

```

1965     \docsvlist{#3}%
1966     }%
1967   }%
1968   \if@addsw%
1969     \letcs{\@tmp}{sw@list@edtext@tmp@the\@tempcnta}%
1970     \ifledRcol%
1971       \xright@appenditem{\the@sw}{\the\absline@numR}}\to\@tmp%
1972     \else%
1973       \xright@appenditem{\the@sw}{\the\absline@num}}\to\@tmp%
1974     \fi%
1975     \cslet{sw@list@edtext@tmp@the\@tempcnta}{\@tmp}%
1976   \fi%
1977 }%
1978 {}%
1979 \advance\@tempcnta by -1%
1980 }%
1981 }%
1982 %

```

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

```

1983 \newcommandx{\sameword@inedtext}[2][1,usedefault]{%
1984   \get@sw@txt{#2}%
1985   \unless\ifledRcol%
1986   %

```

Just a precaution.

```

1987   \ifx\sw@list@inedtext\empty%
1988     \def\the@sw{999}%
1989     \def\this@absline{-99}%
1990   \else%
1991   %

```

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

```

1992   \gl@p\sw@list@inedtext\to\@tmp%
1993   \edef\the@sw{\expandafter\@firstoftwo\@tmp}%
1994   \edef\this@absline{\expandafter\@secondoftwo\@tmp}%
1995   \fi%
1996   %

```

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

```

1997   \ifcsdef{sw@\sw@txt @\this@absline @\the\section@num}{%
1998     \numdef{\prev@line}{\this@absline-1}%
1999     \numdef{\sw@atthisline}{\csuse{sw@\sw@txt @\this@absline @\the\
section@num}-\csuse{sw@\sw@txt @\prev@line @\the\section@num}}%
2000   }%
2001   {\numdef{\sw@atthisline}{0}}%
2002   %

```

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

```

2003     \ifnumgreater{\sw@atthisline}{1}%
2004         {\showwordrank{#2}{\the@sw}}%
2005         {#2}%
2006 %

```

And the same for right side.

```

2007 \else%
2008     \ifx\sw@list@inedtext\empty%
2009         \def\the@sw{999}%
2010         \def\this@absline{-99}%
2011     \else%
2012         \gl@p\sw@list@inedtext\to\@tmp%
2013         \edef\the@sw{\expandafter\@firstoftwo\@tmp}%
2014         \edef\this@absline{\expandafter\@secondoftwo\@tmp}%
2015     \fi%
2016     \ifcsdef{sw@\sw@txt @\this@absline @\the\section@numR @R}{%
2017         \numdef{\prev@line}{\this@absline-1}%
2018         \numdef{\sw@atthisline}{\csuse{sw@\sw@txt @\this@absline @\the\
section@numR @R}-\csuse{sw@\sw@txt @\prev@line @\the\section@numR @R}}%
2019     }%
2020     {\numdef{\sw@atthisline}{0}}%
2021     \ifnumgreater{\sw@atthisline}{1}%
2022         {\showwordrank{#2}{\the@sw}}%
2023         {#2}%
2024     \fi%
2025 }%
2026 %

```

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

```

2027 \newcommand{\showwordrank}[2]{%
2028     #1\textsuperscript{#2}%
2029 }%
2030 %

```

## 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.

```
2031 \newbox\raw@text
2032 \newif\ifnumberedpar@
2033 \newcount\num@lines
2034 \newbox\one@line
2035 \newcount\par@line
2036 %
```

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

```
2037 \newcount\pstarts@typeset@L%
2038 \newcount\pstarts@read@L%
2039 %
```

`\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 `\AtStartEveryPstart` `\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.

`\labelpstarttrue` `\labelpstartfalse` `\thepstart`

```
2040 \ifat@every@pstart@star@
2041 \newcommand{\AtStartEveryPstart}[1]{%
2042   \ifstrempy{#1}%
2043   {\gdef\@at@start@every@pstart{}}%
2044   {\gdef\@at@start@every@pstart{#1}}%
2045 }%
2046 \def\@at@start@every@pstart{}%
2047
2048 \newif\ifat@every@pstart@star@%
```

```

2049 \newcommand{\AtEveryPstart}[1]{%
2050   \ifstrempy{#1}%
2051     {\gdef\at@every@pstart{}}%
2052     {\gdef\at@every@pstart{\noindent#1}}%
2053   \global\at@every@pstart@star@false%
2054 }%
2055 \WithSuffix\newcommand\AtEveryPstart*[1]{%
2056   \ifstrempy{#1}%
2057     {\gdef\at@every@pstart{}}%
2058     {\gdef\at@every@pstart{#1}}%
2059   \global\at@every@pstart@star@true%
2060 }%
2061 \def\at@every@pstart{}%
2062
2063 \newcounter{pstart}
2064 \renewcommand{\thepstart}{\bfseries\@arabic\c@pstart}. }
2065 \newif\ifnumberpstart
2066 \numberpstartfalse
2067 \newif\iflabelpstart
2068 \labelpstartfalse
2069 \newcommandx*\pstart[2][1,2,usedefault]{%
2070   \normal@pars%
2071   \ifboolexpr{%
2072     test {\ifstrempy{#1}}%
2073     and test {\ifstrempy{#2}}%
2074   }%
2075     {\at@every@pstart}%
2076     {%
2077       \ifstrempy{#1}{\noindent#1}%
2078       \ifstrempy{#2}{#2}%
2079     }%
2080   \ifautopar%
2081     \autopar%
2082   \fi%
2083   \ifluatex%
2084     \edef\l@luatextextdir@L{\the\textdir}%
2085   \fi%
2086   \@nobraektrue%
2087   \ifnumbering \else%
2088     \led@err@PstartNotNumbered%
2089     \beginnumbering%
2090   \fi%
2091   \ifnumberedpar@%
2092     \led@err@PstartInPstart%
2093   \pend%
2094   \fi%
2095   \list@clear{\inserts@list}%
2096   \global\let\next@insert=\empty%
2097   \begingroup\normal@pars%
2098   \global\advance \l@dnumpstartsL\@ne

```



```

2099 \global\advance \pstarts@typeset@L\@ne%
2100 \global\advance \pstarts@read@L\@ne%
2101 \global\setbox\raw@text=\vbox\bgroup%
2102 \if@nobreak%
2103   \if@afterindent\else%
2104     \noindent%
2105     \global\@afterindenttrue%
2106   \fi%
2107 \fi%
2108 \ifboolexpr{%
2109   bool{autopar}%
2110   and bool{by@autopar}%
2111 }%
2112 {}%
2113 {%
2114   \ifnumberpstart%
2115     \ifinstanza\else%
2116       \ifsidepstartnum\else%
2117         \thepstart%
2118       \fi%
2119     \fi%
2120   \fi%
2121 }%
2122 \numberedpar@true%
2123 \iflabelpstart%
2124   \protected@edef\@currentlabel{\p@pstart\thepstart}%
2125 \fi%
2126 \l@dzero penalties%
2127 \@at@start@every@pstart%
2128 \global\by@autoparfalse%
2129 \ignorespaces%because not automatically ignored if an optional argument
is used (classical TeX behavior for space after commands)
2130 }
2131 %

```

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

```

2132 \newcommand*{\pend}[2][1,2,usedefault]{\ifnumbering \else%
2133   \led@err@PendNotNumbered%
2134 \fi%
2135 \global\l@dskipversenumberfalse%
2136 \ifnumberedpar@ \else%
2137   \led@err@PendNoPstart%
2138 \fi%
2139 %

```

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.

```

2140 \l@dzeroopenalties%
2141 \@at@end@every@pend%
2142 \endgraf\global\num@lines=\prevgraf\egroup%
2143 \global\par@line=0%
2144 %

```

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.

```

2145 \csnumdef{pstartline}{0}%
2146 \loop\ifvbox\raw@text%
2147   \csnumdef{pstartline}{\pstartline+\@ne}%
2148   \do@line%
2149   \ifbypstart@%
2150     \ifnumequal{\pstartline}{1}{%
2151       \bgroup%
2152       \let\leavevmode\relax%
2153       \setline{1}%
2154       \egroup%
2155       \resetprevline@}{}%
2156     \fi%
2157   \repeat%
2158 %

```

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

```

2159 \flush@notes%
2160 \endgroup%
2161 \ignorespaces%
2162 %

```

Increase pstart counter.

```

2163 \ifnumberpstart%
2164   \global\pstartnumtrue%
2165 \fi%
2166 \addtocounter{pstart}{1}%
2167 \ifcontinuousnumberingwithcolumns%
2168   \addtocounter{pstartL}{1}%
2169   \addtocounter{pstartR}{1}%
2170 \fi%
2171 %

```

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

```

2172 \normal@pars%
2173 \ifboolexpr{%
2174   test {\ifstrempy{#1}}%

```

```

2175     and test {\ifstrempy{#2}}%
2176   }%
2177   {\at@every@pend}%
2178   {%
2179     \ifstrempy{#1}{\noindent#1}%
2180     \ifstrempy{#2}{#2}%
2181   }%
2182 %

```

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.

```

2183 \@nobreakfalse%
2184 \ifautopar%
2185   \autopar%
2186 \fi%
2187 }
2188 %

```

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.

```

\AtEveryPend89
\at@every@pend90
\ifat@every@pend@star91 \newif\ifat@every@pend@star%
2192 \newcommand{\AtEveryPend}[1]{%
2193   \ifstrempy{#1}%
2194     {\gdef\at@every@pend{}}%
2195     {\gdef\at@every@pend{\noindent#1}}%
2196   \global\at@every@pend@star@false%
2197 }%
2198 \WithSuffix\newcommand\AtEveryPend*[1]{%
2199   \ifstrempy{#1}%
2200     {\gdef\at@every@pend{}}%
2201     {\gdef\at@every@pend{#1}}%
2202   \global\at@every@pend@star@true%
2203 }%
2204 \xdef\at@every@pend{}%
2205
2206 %

```

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

```

2207 \newcommand{\AtEndEveryPend}[1]{%
2208   \ifstrempy{#1}%
2209     {\xdef\@at@end@every@pend{}}%
2210     {\gdef\@at@end@every@pend{#1}}%
2211   }%
2212 \def\@at@end@every@pend{}%
2213 %

```

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

```

2214 \newcommand*\l@dzzeropenalties{%
2215   \brokenpenalty \z@ \clubpenalty \z@
2216   \displaywidowpenalty \z@ \interlinepenalty \z@ \predisplaypenalty \z@
2217   \postdisplaypenalty \z@ \widowpenalty \z@}
2218
2219 %

```

`\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.

```

2220 \newif\ifautopar
2221 \newif\ifby@autopar%
2222 \newcommand*\autopar{%
2223   \ifledRcol
2224     \ifnumberingR \else
2225       \led@err@AutoparNotNumbered
2226       \beginnumberingR
2227     \fi
2228   \else
2229     \ifnumbering \else
2230       \led@err@AutoparNotNumbered
2231     \beginnumbering

```

```

2232 \fi
2233 \fi
2234 \autopartrue
2235 \everypar{\setbox0=\lastbox
2236 \endgraf \vskip-\parskip
2237 \global\by@autopartrue%
2238 \pstart \noindent \kern\wd0%
2239 \ifnumberpstart%
2240 \ifinstanza\else%
2241 \thepstart%
2242 \fi%
2243 \fi%
2244 \let\par=\pend}%
2245 \ignorespaces}
2246 %

```

**\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.

```

2247 \newcommand*{\normal@pars}{\everypar{}\let\par\endgraf}
2248
2249 %

```

**\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.

```

2250 \newif\ifautopar@pause
2251 %

```

## 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.

```

2252 \newcommand*{\l@dunhbox@line}[1]{\unhbox #1}
2253 \newcommand*{\do@line}{%
2254 {\vbadness=10000
2255 \splittopskip=\z@
2256 \do@linehook
2257 \l@demptyd@ta
2258 \global\setbox\one@line=\vsplit\raw@text to\baselineskip}%
2259 \unvbox\one@line \global\setbox\one@line=\lastbox
2260 \getline@num
2261 \IfStrEq{\led@pb@setting}{before}{\led@check@pb\led@check@nopb}{}}

```

```

2262 \ifnum\@lock>\@ne
2263 \inserthangingsymboltrue
2264 \else
2265 \inserthangingsymbolfalse
2266 \fi
2267 \check@pb@in@verse
2268 \ifl@dhidenumber%
2269 \global\l@dhidenumberfalse%
2270 \f@x@l@cks%
2271 \else%
2272 \affixline@num%
2273 \fi%
2274 %

```

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

```

2275 \xifinlist{\the\pstarts@typeset@L}{\eled@sections@@}%
2276 {\print@eledsection}%
2277 {\print@line}%
2278 \IfStrEq{\led@pb@setting}{after}{\led@check@pb\led@check@nopb}{\}
2279 }%
2280 %

```

### VII.2.2 Process for “normal” line

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

```

2281 \def\print@line{
2282 %

```

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

```

2283 \affixpstart@num%
2284 %

```

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

```

2285 \hb@xt@ \linewidth{%
2286 %

```

User hooks.

```

2287 \do@insidelinehook%
2288 \csuse{insidethis@the\absline@num @the\section@num}%
2289 \global\csundef{insidethis@the\absline@num @the\section@num}%
2290 %

```

Left line number

```

2291 \l@dld@ta%
2292 %

```

Prepare text to be inserted before notes.

```

2293 \if@firstlineofpage%
2294 \set@Xtxtbeforenotes%
2295 \set@txtbeforenotesX%
2296 \global\@firstlineofpagefalse%
2297 \fi%
2298 %

```

Insert footnotes made of manuscripts data and critical footnotes.

```

2299 \ifdefstring{\ms@data@position}{msdata-regular}{%
2300 \insert@msdata%
2301 \add@inserts%
2302 \add@Xgroupbyline%
2303 }{%
2304 \add@inserts%
2305 \add@Xgroupbyline%
2306 \insert@msdata%
2307 }%
2308 %

```

Insert marginal notes.

```

2309 \affixside@note%
2310 %

```

Print left notes.

```

2311 \l@dlsn@te
2312 %

```

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

```

2313 {\ledllfill\hb@xt@ \wd\one@line{%
2314 \new@line%
2315 %

```

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

```

2316 \continuousnumberingwithcolumns@sync@linenumber@singletext%
2317 %

```

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

```

2318 \ifluatex%
2319 \texdir\l@luatextextdir@L%
2320 \fi%
2321 %

```

Insert, if needed, the hanging symbol.

```

2322 \inserthangingsymbol%
2323 %

```

And so, print the line.

```

2324 \l@dunhbox@line{\one@line}}%
2325 %

```

Right line number

```
2326 \ledrlfill\l@drd@ta%
2327 %
```

Print right notes.

```
2328 \l@drsn@te%
2329 }}%
2330 %
```

And reinsert penalties (for page breaking)...

```
2331 \add@penalties%
2332 }
2333 %
```

### 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.

```
2334 \def\print@eledsection{%
2335   \if@firstlineofpage%
2336     \set@Xtxtbeforenotes%
2337     \set@txtbeforenotesX%
2338     \global\@firstlineofpagefalse%
2339   \fi%
2340   \ifdefstring{\ms@data@position}{msdata-regular}{%
2341     \insert@msdata%
2342     \add@inserts%
2343     \add@Xgroupbyline%
2344   }{%
2345     \add@inserts%
2346     \add@Xgroupbyline%
2347     \insert@msdata%
2348   }%
2349   \affixside@note%
2350   \numdef{\temp@}{\pstarts@typeset@L-1}%
2351   \xifinlist{\temp@}{\eled@sections@@}{\@nobreaktrue}{\@nobreakfalse}%
2352   \@eled@sectioningtrue%
2353   \csuse{eled@sectioning@the\pstarts@typeset@L}%
2354   \@eled@sectioningfalse%
2355   \global\csundef{eled@sectioning@the\pstarts@typeset@L}%
2356   \if@RTL%
2357     \hspace{-3\paperwidth}%
2358     {\hbox{\l@dunhbox@line{\one@line}} \new@line}%
2359   \else%
2360     \hspace{3\paperwidth}%
2361     {\new@line \hbox{\l@dunhbox@line{\one@line}}}%

```



```

2362 \fi%
2363 \vskip-\baselineskip%
2364 \continuousnumberingwithcolumns@sync@linenumber@singletext%
2365 }
2366 %

```

### VII.2.4 Hooks

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

```

2367 \newcommand*\do@linehook{}
2368 \newcommand*\do@insidelinehook{}
2369 %

```

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

```

2370 \newcommand*\dolinehook[1]{\gdef\do@linehook{#1}}%
2371 \newcommand*\doinsidelinehook[1]{\gdef\do@insidelinehook{#1}}%
2372
2373 %

```

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

```

2374 \newcommand{\doinsidethislinehook}[1]{%
2375 \leavevmode%In case it begins with a \pstart, ensure the \@insidethisline
is written after \@nl
2376 \ifledRcol%
2377 \write\linenum@outR{\string\@insidethisline[\unexpanded{#1}]}%
2378 \else%
2379 \write\linenum@out{\string\@insidethisline[\unexpanded{#1}]}%
2380 \fi%
2381 }%
2382 %

```

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

```

2383 \newcommand{\@insidethisline}[1]{%
2384 \ifledRcol%
2385 \csgappto{insidethisR@the\absline@numR @the\section@numR}{#1}%
2386 \else%
2387 \csgappto{insidethis@the\absline@num @the\section@num}{#1}%
2388 \fi%
2389 }%
2390 %

```

### 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 notes.

```

\l@dcsnotetext
\l@dcsnotetext@l
\l@dcsnotetext@r
2391 \newcommand*{\l@emptyd@ta}{%
2392   \gdef\l@dld@ta{}%
2393   \gdef\l@drd@ta{}%
2394   \gdef\l@dcsnotetext@l{}%
2395   \gdef\l@dcsnotetext@r{}%
2396   \gdef\l@dcsnotetext{}%
2397
2398 %

```

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

`\l@drsn@te`

```

2399 \newcommand{\l@dlsn@te}{%
2400   \ifboolexpr{%
2401     bool {l@dprintingcolumns}%
2402     and bool {ledRcol}%
2403   }{% If we are on a right column
2404     \@tempdima=\@morespace@leftnote@rightcolumn%
2405   }{%
2406     \@tempdima=\z@%
2407   }%
2408   \hb@xt@ \z@{\hss\box\l@dlp@rbox\kern\ledlsnotesep\hskip\@tempdima}%
2409 }%
2410 \newcommand{\l@drsn@te}{%
2411   \ifboolexpr{%
2412     bool {l@dprintingcolumns}%
2413     and not bool {ledRcol}%
2414   }{% If we are on a left column
2415     \@tempdima=\@morespace@rightnote@leftcolumn%
2416   }{%
2417     \@tempdima=\z@%
2418   }%
2419   \hb@xt@ \z@{\hskip\@tempdima\kern\ledrsnotesep\box\l@drp@rbox\hss}%
2420 }%
2421
2422 %

```

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

`\ledrlfill`

```

2423 \newcommand*{\ledllfill}{\hfil}
2424 \newcommand*{\ledrlfill}{}
2425
2426 %

```

## 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.

```

2427 \newcommand*{\getline@num}{%
2428   \global\advance\absline@num \@ne%
2429   \do@actions
2430   \do@ballast
2431   \ifnumberline
2432     \ifsublines@
2433       \ifnum\sub@lock<\tw@
2434         \global\advance\subline@num \@ne
2435       \fi
2436     \else
2437       \ifnum\@lock<\tw@
2438         \global\advance\line@num \@ne
2439         \global\subline@num \z@
2440       \fi
2441     \fi
2442   \fi
2443 }
2444 %

```

`\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. 176).

`\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`

```

2445 \newcount\ballast@count
2446 \newcounter{ballast}
2447 \setcounter{ballast}{0}
2448 %

```

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.

```

2449 \newcommand*{\do@ballast}{\global\ballast@count \z@
2450   \begingroup
2451   \advance\absline@num \@ne
2452   \ifnum\next@actionline=\absline@num
2453     \ifnum\next@action>-1001\relax
2454     \global\advance\ballast@count by -\c@ballast
2455   \fi
2456 \fi

```

```

2457 \endgroup}
2458 %

```

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

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

```

2459 \newcommand*{\do@actions}{%
2460   \global\let\do@actions@next=\relax
2461   \ifnum\absline@num<\next@actionline\else
2462 %

```

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.

```

2463   \ifnum\next@action>-1001
2464     \global\page@num=\next@action
2465     \ifresumenumbering@start%
2466       \setbox0=\hbox{}%Required to get the correct page number, when the
\resumenumbering is just after a \newpage
2467       \ifnum\pausenumbering@page@num<\page@num%
2468         \global\resumenumbering@startfalse%
2469       \fi%
2470     \fi%
2471     \ifboolexpr{%
2472       bool{resumenumbering@start}%
2473       and test {\ifdimgreater{\pagedepth}{\z@}}%
2474     }%
2475     {}%
2476     {\global\@firstlineofpagetrue}%
2477     \ifcsdef{reset@line\the\absline@num @\the\section@num}%
2478     {%
2479       \global\line@num=\z@%
2480       \global\subline@num=\z@%
2481       \resetprevline@%
2482     }%
2483     {}%
2484     \global\resumenumbering@startfalse%
2485     \add@msdata@firstlineofpage%
2486 %

```

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`.)

```

2487   \else
2488     \ifnum\next@action<-4999
2489       \@l@dttempcnta=-\next@action

```

```

2490     \advance\@l@dttempcnta by -5001
2491     \ifsublines@
2492         \global\subline@num=\@l@dttempcnta
2493     \else
2494         \global\line@num=\@l@dttempcnta
2495     \fi
2496 %

```

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

```

2497     \else
2498         \@l@dttempcnta=-\next@action
2499         \advance\@l@dttempcnta by -1000
2500         \do@actions@fixedcode
2501     \fi
2502 \fi
2503 %

```

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.

```

2504     \ifx\actionlines@list\empty
2505         \gdef\next@actionline{1000000}%
2506     \else
2507         \gl@p\actionlines@list\to\next@actionline
2508         \gl@p\actions@list\to\next@action
2509         \global\let\do@actions@next=\do@actions
2510     \fi
2511 \fi
2512 %

```

Make the recursive call, if necessary.

```

2513 \do@actions@next}
2514
2515 %

```

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

```

2516 \newcommand*{\do@actions@fixedcode}{%
2517     \ifcase\@l@dttempcnta
2518     \or%                % 1001 = starting sublineation
2519         \global\sublines@true
2520     \or%                % 1002 = ending sublineation
2521         \global\sublines@false
2522     \or%                % 1003 = starting locking number
2523         \global\@lock=\@one
2524     \or%                % 1004 = ending locking number
2525         \ifnum\@lock=\tw@
2526             \global\@lock=\thr@@

```

```

2527 \else
2528   \global\@lock=\z@
2529 \fi
2530 \or%           % 1005 = starting locking subnumber
2531   \global\sub@lock=\@ne
2532 \or%           % 1006 = ending locking subnumber
2533   \ifnum\sub@lock=\tw@
2534     \global\sub@lock=\thr@@
2535   \else
2536     \global\sub@lock=\z@
2537   \fi
2538 \or%           % 1007 = skipping numbering
2539   \l@dskipnumbertrue
2540 \or%           % 1008 = skipping numbering in stanza
2541   \l@dskipversenumbertrue%
2542 \or%           % 1009 = hiding number
2543   \l@dhidenumbertrue
2544 \or%           % 1010 = inserting msdata
2545   \add@msdata%
2546 \else
2547   \led@warn@BadAction
2548 \fi}
2549
2550
2551 %

```

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

`\continuousnumberingwithcolumns@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.

```

2552 \newcommand{\continuousnumberingwithcolumns@sync@linenumber@singletext}{%
2553   \ifcontinuousnumberingwithcolumns%
2554     \unless\ifafterendnumberingR%
2555       \new@lineR%
2556       \xappto\next@line@list@stuffR{%
2557         \unexpanded{\global\line@numR=}\the\line@num%
2558       }%
2559     \fi%

```

```

2560 \fi%
2561 }%
2562 %

```

## 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.

```

2563 \newcommand{\linenumannotation}[1]{%
2564 \leavevmode%In case it begins with a \pstart, ensure the \@annot is
written after \@nl
2565 \ifledRcol%
2566 \write\linenum@outR{\string\@annot[#1]}%
2567 \else%
2568 \write\linenum@out{\string\@annot[#1]}%
2569 \fi%
2570 }%
2571 %

```

`\linenumannotationposition@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.

```

2572 \def\Xlinenumannotationposition@side{after}%
2573 %

```

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

```

2574 \def\Xwraplinenumannotation@ref{\textsuperscript}%
2575 \def\Xwraplinenumannotation@side{\textsuperscript}%
2576 %

```

`\@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.

```

2577 \newcommand{\@annot}[1]{%
2578   \store@annot@to@absline{#1}%
2579   \def\current@annot{#1}%
2580 }%
2581 \newcommand{\store@annot@to@absline}[1]{%
2582   \ifledRcol%
2583     \csgappto{annotR@the\absline@numR @the\section@numR}{#1}%
2584   \else%
2585     \csgappto{annot@the\absline@num @the\section@num}{#1}%
2586   \fi%
2587 }%
2588 %

```

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

```

2589 \let\current@annot=\empty%
2590 %

```

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

```

2591 \newcommand{\reset@current@annot}{%
2592   \let\current@annot\empty%
2593 }%
2594 %

```

`\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.

```

2595 \def\parse@annot#1|#2|{%
2596   \gdef\annot@start{#1}%
2597   \gdef\annot@end{#2}%
2598 }%
2599 %

```



## 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:

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

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

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

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

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

```
2600 \newcommand*{\affixline@num}{%
2601 %
```

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).

```
2602 \ifledgroupnotesL@else
2603 \ifnumberline
2604 \ifl@dskipnumber
2605 \global\l@dskipnumberfalse
2606 \else
2607 \ifsublines@
2608 \l@tempcntb=\subline@num
2609 \ifnum\subline@num>\c@firstsublinenum
2610 \l@tempcnta=\subline@num
2611 \advance\l@tempcnta by-\c@firstsublinenum
2612 \divide\l@tempcnta by\c@sublinenumincrement
2613 \multiply\l@tempcnta by\c@sublinenumincrement
2614 \advance\l@tempcnta by\c@firstsublinenum
2615 \else
2616 \l@tempcnta=\c@firstsublinenum
2617 \fi
2618 %
```

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.

```
2619 \ch@cksubl@ck
2620 %
```

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

```

2621     \else
2622         \@l@tempcntb=\line@num
2623     %

```

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

```

2624         \ifx\linenumberlist\empty
2625             \ifnum\line@num>\c@firstlinenum
2626                 \@l@tempcnta=\line@num
2627                 \advance\@l@tempcnta by-\c@firstlinenum
2628                 \divide\@l@tempcnta by\c@linenumincrement
2629                 \multiply\@l@tempcnta by\c@linenumincrement
2630                 \advance\@l@tempcnta by\c@firstlinenum
2631             \else
2632                 \@l@tempcnta=\c@firstlinenum
2633             \fi
2634         \else
2635     %

```

The `\linenumberlist` was not `\empty`, so here is Wayne's numbering mechanism. This takes place in `TEX`'s mouth.

```

2636         \@l@tempcnta=\line@num
2637         \edef\rem@inder{\linenumberlist,\number\line@num,}%
2638         \edef\sc@n@list{\def\noexpand\sc@n@list
2639             ###1,\number\@l@tempcnta,###2|{\def\noexpand\rem@inder
2640 {####2}}}%
2641         \sc@n@list\expandafter\sc@n@list\rem@inder|
2642         \ifx\rem@inder\empty%
2643             \advance\@l@tempcnta\@ne
2644         \fi
2645     %

```

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

```

2646         \ch@ck@l@ck
2647     \fi
2648 %

```

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

```

2649         \ifnum\@l@tempcnta=\@l@tempcntb
2650             \ifl@dskipversenumber\else
2651     %

```

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`

```

2652         \if@twocolumn
2653             \if@firstcolumn
2654                 \gdef\l@dld@ta{\llap{\leftlinenum}}}%
2655             \else
2656                 \gdef\l@drd@ta{\rlap{\rightlinenum}}}%
2657             \fi
2658         \else
2659             \ifboolexpr{bool {l@dprintingcolumns} and test {\
ifnumgreater{\line@margin@columns}{\m@ne}}}%
2660                 {\@l@tempcntb=\line@margin@columns}%
2661                 {\@l@tempcntb=\line@margin}%
2662                 \ifnum\@l@tempcntb>\@ne
2663                     \advance\@l@tempcntb \page@num
2664                 \fi
2665                 \ifboolexpr{%
2666                     bool {l@dprintingcolumns}%
2667                     and (%
2668                         (test {\ifdefstring{\linenum@OnlyPages@ForColumns}{left
2669
2670                         and test {\ifnumodd{\page@num}}}%
2671                         )%
2672                         or%
2673                         (test {\ifdefstring{\linenum@OnlyPages@ForColumns}{right
2674
2675                         and not test {\ifnumodd{\page@num}}}%
2676                         )%
2677                     )%
2678                     }%
2679                     {}%
2680                     {\ifodd\@l@tempcntb%
2681                         \gdef\l@drd@ta{\rlap{\rightlinenum}}}%
2682                     \else%
2683                         \gdef\l@dld@ta{\llap{\leftlinenum}}}%
2684                     \fi%
2685                 }%
2686             \fi
2687         \fi
2688     %

```

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

```

2689     \f@x@l@cks
2690     \fi
2691   \fi
2692 \fi
2693 }
2694
2695 %

```

`\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.

```

2696 \newcommand*{\ch@cksub@l@ck}{%
2697   \ifcase\sub@lock
2698   \or
2699     \ifnum\sublock@disp=\@ne
2700       \@l@tempcntb=\z@ \@l@tempcnta=\@ne
2701     \fi
2702   \or
2703     \ifnum\sublock@disp=\tw@ \else
2704       \@l@tempcntb=\z@ \@l@tempcnta=\@ne
2705     \fi
2706   \or
2707     \ifnum\sublock@disp=\z@
2708       \@l@tempcntb=\z@ \@l@tempcnta=\@ne
2709     \fi
2710   \fi}
2711 %

```

Similarly for line numbers.

```

2712 \newcommand*{\ch@ck@l@ck}{%
2713   \ifcase\@lock
2714   \or
2715     \ifnum\lock@disp=\@ne
2716       \@l@tempcntb=\z@ \@l@tempcnta=\@ne
2717     \fi
2718   \or
2719     \ifnum\lock@disp=\tw@ \else
2720       \@l@tempcntb=\z@ \@l@tempcnta=\@ne
2721     \fi
2722   \or
2723     \ifnum\lock@disp=\z@
2724       \@l@tempcntb=\z@ \@l@tempcnta=\@ne
2725     \fi
2726   \fi}
2727 %

```

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

```

2728 \newcommand*{\f@x@l@cks}{%
2729   \ifcase\@lock
2730   \or
2731     \global\@lock=\tw@
2732   \or \or
2733     \global\@lock=\z@
2734   \fi
2735   \ifcase\sub@lock
2736   \or
2737     \global\sub@lock=\tw@
2738   \or \or
2739     \global\sub@lock=\z@
2740   \fi}
2741
2742 %

```

## XI Pstart number printing inside

Inside, the printing of the pstart number runs like the printing of the line number. There are only a few differences:

- The pstarts counter is upgraded in the `\pend` command. Consequently, the `\affixpstart@num` command has not to upgrade it, unlike the `\affixline@num` which upgrades the lines counter.
- To print the pstart number only at the beginning of a pstart, and not in every line, a boolean test is made. The `\pstartnum` boolean is set to TRUE at every `\pend`. It is tried in the `\leftpstartnum` and `\rightpstartnum` commands. After the try, it is set to FALSE.

```

\leftpstartnum 43
\rightpstartnum 44 \newif\ifsidepstartnum
\ifsidepstartnum 45 \newcommand*{\affixpstart@num}{%
2746   \ifsidepstartnum
2747     \if@twocolumn
2748       \if@firstcolumn
2749         \gdef\l@dld@ta{\llap{\leftpstartnum}}}%
2750       \else
2751         \gdef\l@drd@ta{\rlap{\rightpstartnum}}}%
2752       \fi
2753     \else
2754       \@l@tempcntb=\line@margin%
2755       \ifnum\@l@tempcntb>\@ne
2756         \advance\@l@tempcntb \page@num
2757       \fi
2758       \ifodd\@l@tempcntb
2759         \gdef\l@drd@ta{\rlap{\rightpstartnum}}}%
2760       \else

```

```

2761 \gdef\l@dld@ta{\llap{{\leftpstartnum}}}%
2762 \fi
2763 \fi
2764 \fi
2765 }
2766 %
2767
2768 \newif\ifpstartnum
2769 \pstartnumtrue
2770 \newcommand*{\leftpstartnum}{
2771   \ifpstartnum\thepstart
2772   \kern\linenumsep\fi
2773   \global\pstartnumfalse
2774 }
2775 \newcommand*{\rightpstartnum}{
2776   \ifpstartnum
2777   \kern\linenumsep
2778   \thepstart
2779   \fi
2780   \global\pstartnumfalse
2781 }
2782 %
2783 %

```

## 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.

```

2784 \list@create{\inserts@list}
2785 %

```

`\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`.

```
2786 \newcommand*{\add@inserts}{%
2787   \global\let\add@inserts@next=\relax
2788   %
```

If `\inserts@list` is empty, there are not any more notes or insertions for this paragraph, and we need not waste our time.

```
2789   \ifx\inserts@list\empty \else
2790   %
```

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.

```
2791   \ifx\next@insert\empty
2792     \ifx\insertlines@list\empty
2793       \global\noteschanged@true
2794       \gdef\next@insert{100000}%
2795     \else
2796       \gl@p\insertlines@list\to\next@insert
2797     \fi
2798   \fi
2799   %
```

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.

```
2800   \ifnum\next@insert=\absline@num
2801     \gl@p\inserts@list\to\@insert
2802     \@insert
2803     \global\let\@insert=\undefined
2804     \global\let\next@insert=\empty
2805     \global\let\add@inserts@next=\add@inserts
2806   \fi
2807 \fi
2808 %
```

Make the recursive call, if necessary.

```
2809 \add@inserts@next}
2810
2811 %
```

**`\add@Xgroupbyline`** If you use `\Xgroupbyline`, the insertion of the critical footnotes are not made immediately in `\add@inserts`, but the content to be inserted is stored, to be inserted in one block. This insertion in one block is made by `\add@Xgroupbyline`.

```
2812 \newcommand{\add@Xgroupbyline}{%
2813   \unless\ifnocritical%
```

```

2814 \def\do##1{%Looping on the series
2815 \def\do####1{%Looping on the ##1@forinserting command
2816 \ifcsdef{##1@forinserting@####1}{%
2817 \X@beforeinsertion{##1}%
2818 \if@ledgroup%
2819 \global\setbox\@nameuse{mp##1footins}=\vbox%
2820 \else%
2821 \insert\csname ##1footins\endcsname%
2822 \fi%
2823 {%
2824 \ifcsdef{Xhsize\csuse{series@display##1}@##1}%
2825 {\hsize \csuse{Xhsize\csuse{series@display##1}@##1}}%
2826 }%
2827 \if@ledgroup%
2828 \unvbox\@nameuse{mp##1footins}%
2829 \fi%
2830 \X@atbegininsertion{##1}%
2831 \ifcsstring{series@display##1}%
2832 {%
2833 \Xledsetnormalparstuff{##1}%
2834 \rule\z@\splittopskip%
2835 }%
2836 }%
2837 \csuse{##1@forinserting@####1}%
2838 \strut\par%
2839 }%
2840 \global\csundef{##1@forinserting@####1}%
2841 }%
2842 {%
2843 }%
2844 \ifcsdef{##1@forinserting}{%
2845 \dolistcslloop{##1@forinserting}%
2846 }{%
2847 \global\csundef{##1@forinserting}%
2848 }%
2849 \dolistloop{\@series}%
2850 \fi%
2851 }%
2852
2853
2854 %

```

## XII.2 Penalties

**\add@penalties** \add@penalties is the last macro used by \do@line. It adds up the club, widow, and interline penalties, and puts a single penalty of the appropriate size back into the paragraph; these penalties get removed by the \vsplit operation. \displaywidowpenalty and \brokenpenalty are not restored, since we have no easy way to find out where we should insert them.



In this code, `\num@lines` is the number of lines in the whole paragraph, and `\par@line` is the line we are working on at the moment. The count `\@l@dttempcnta` is used to calculate and accumulate the penalty; it is initially set to the value of `\ballast@count`, which has been worked out in `\do@ballast` above (VIII p. 163). Finally, the penalty is checked to see that it does not go below  $-10000$ .

```

2855 \newcommand*{\add@penalties}{\@l@dttempcnta=\ballast@count
2856 \ifnum\num@lines>\@ne
2857 \global\advance\par@line \@ne
2858 \ifnum\par@line=\@ne
2859 \advance\@l@dttempcnta \clubpenalty
2860 \fi
2861 \@l@dttempcntb=\par@line \advance\@l@dttempcntb \@ne
2862 \ifnum\@l@dttempcntb=\num@lines
2863 \advance\@l@dttempcnta \widowpenalty
2864 \fi
2865 \ifnum\par@line<\num@lines
2866 \advance\@l@dttempcnta \interlinepenalty
2867 \fi
2868 \fi
2869 \ifnum\@l@dttempcnta=\z@
2870 \relax
2871 \else
2872 \ifnum\@l@dttempcnta>-10000
2873 \penalty\@l@dttempcnta
2874 \else
2875 \penalty -10000
2876 \fi
2877 \fi}
2878
2879 %

```

### XII.3 Printing leftover notes

**`\flush@notes`** The `\flush@notes` macro is called after the entire paragraph has been sliced up and sent on to the vertical list. If the number of notes to this paragraph has increased since the previous run of  $\text{\TeX}$ , then there can be leftover notes that have not yet been printed. An appropriate error message will be printed elsewhere; but it is best to go ahead and print these notes somewhere, even if it is not in quite the right place. What we do is dump them all out here, so that they should be printed on the same page as the last line of the paragraph. We can hope that is not too far from the proper location, to which they will move on the next run. For the first run, we do not flush the notes, as that means all the notes will be added at the end of numbered section, and so, very far of the expected position

```

2880 \newcommand*{\flush@notes}{%
2881 \iftoggle{notfirstrun@jobname.\extensionchars\the\section@num}{%
2882 \@xloop%
2883 \ifx\inserts@list\empty \else%

```

```

2884 \gl@p\inserts@list\to\@insert%
2885 \@insert%
2886 \global\let\@insert=\undefined%
2887 \repeat%
2888 }{}%
2889 }%
2890
2891
2892 %

```

**\@xloop** `\@xloop` is a variant of the PLAIN  $\text{\TeX}$  `\loop` macro, useful when it's hard to construct a positive test using the  $\text{\TeX}$  `\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  $\text{\TeX}$  `\loop` is used, too, so we could just call it `\loop`; but it seems preferable not to change the definitions of any of the standard macros.)

This variant of `\loop` was introduced by Alois Kabelschacht in *TUGboat* **8** (1987), pp. 184–5.

```

2893 \def\@xloop#1\repeat{%
2894 \def\body{#1\expandafter\body\fi}%
2895 \body}
2896
2897 %

```

## 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.

```

2898 \newcommand{\set@Xtxtbeforenotes}{%
2899 \unless\ifnocritical@%
2900 \def\do##1{%
2901 \nottoggle{Xtxtbeforenotesonlyonce@##1}{%
2902 \global\togglefalse{Xtxtbeforenotes@##1@typeset}%
2903 }{}%
2904 }%
2905 \dolistloop{\@series}%
2906 \fi%
2907 }%
2908 %

```

**\set@txtbeforenotesX** The `\set@txtbeforenotesX` does the same for the `\textbeforenotesX`.

```

2909 \newcommand{\set@txtbeforenotesX}{%
2910 \unless\ifnofamiliar@%
2911 \def\do##1{%
2912 \nottoggle{txtbeforenotesonlyonceX@##1}{%

```

```

2913 \global\togglefalse{txtbeforenotesX@##1@typeset}%
2914 }{}%
2915 }%
2916 \dolistloop{\@series}%
2917 \fi%
2918 }%
2919 %

```

`\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.

```

2920 \newcommand{\insert@Xtxtbeforenotes}[1]{%
2921 \nottoggle{Xtxtbeforenotes@#1@typeset}{%
2922 \global\toggletrue{Xtxtbeforenotes@#1@typeset}%
2923 \ifcsvoid{Xtxtbeforenotes@#1}{%
2924 \ifcsstring{series@display#1}{paragraph}%
2925 {\noindent\csuse{Xtxtbeforenotes@#1}}%
2926 {\expandafter\insert\csname#1footins\endcsname%
2927 \bgroup%
2928 \noindent%
2929 \ifcsdef{\csuse{series@display#1}@begin@insert}{%
2930 \csuse{\csuse{series@display#1}@begin@insert}{#1}%
2931 }{}%
2932 \strut\csuse{Xnotefontsize@#1}%
2933 \csuse{Xtxtbeforenotes@#1}%
2934 \egroup%
2935 }%
2936 }%
2937 }%
2938 {}%
2939 }%
2940
2941 \newcommand{\insert@txtbeforenotesX}[1]{%
2942 \nottoggle{txtbeforenotesX@#1@typeset}{%
2943 \global\toggletrue{txtbeforenotesX@#1@typeset}%
2944 \ifcsvoid{txtbeforenotesX@#1}{%
2945 \ifcsstring{series@displayX#1}{paragraph}%
2946 {\noindent\csuse{txtbeforenotesX@#1}}%
2947 {\expandafter\insert\csname footins#1\endcsname%
2948 \bgroup%
2949 \noindent%
2950 \ifcsdef{\csuse{series@displayX#1}@begin@insert}{%
2951 \csuse{\csuse{series@displayX#1}@begin@insert}{#1}%
2952 }{}%
2953 \strut\csuse{notefontsizeX@#1}\csuse{txtbeforenotesX@#1}%
2954 \egroup%
2955 }%

```

```

2956 }%
2957 }%
2958 }%
2959 }%
2960 }%
2961
2962
2963 %

```

## XIII Critical footnotes

The footnote macros are adapted from those in PLAIN T<sub>E</sub>X, but they differ in these respects: the outer-level commands must add other commands to a list macro rather than doing insertions immediately; there are many separate levels of the footnotes, not just one; and there are options to reformat footnotes into paragraphs or into multiple columns.

### XIII.1 Fonts

Before getting into the details of formatting the notes, we set up some font macros. It is the notes that present the greatest challenge for our font-handling mechanism, because we need to be able to take fragments of our main text and print them in different forms: it is common to reduce the size, for example, without otherwise changing the fonts used.

`\select@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.

```

2964 \def\select@lemmafont#1|#2|#3|#4|#5|#6|#7|{\select@lemmafont#7|}
2965 \def\select@lemmafont#1/#2/#3/#4|{%
2966   {\fontencoding{#1}\fontfamily{#2}\fontseries{#3}\fontshape{#4}%
2967   \selectfont}
2968
2969 %

```

### 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.

```

2970 \newcommand*{\footnoteoptions@}[3]{%
2971   \def\do##1{%
2972     \ifstrequal{#1}{L}{% On the left side
2973       \xright@appenditem{\noexpand\setkeys[mac]{#3footnoteoption}{\
2974         unexpanded{##1}}}{\to\inserts@list%
2975       \global\advance\insert@count \@ne% Increment the left insert
2976       counter.

```

```

2975     }%
2976     {%
2977         \xright@appenditem{\noexpand\setkeys[mac]{#3footnoteoption}{\
unexpanded{##1}}}\to\inserts@listR%
2978         \global\advance\insert@countR \@ne% Increment the right insert
counter insert.
2979     }%
2980     }%
2981     \notblank{#2}{\docsvlist{#2}}{}% Parsing all options
2982 }
2983 %

```

### 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.

```

2984 \newcommand*{\footnotelang@lua}[1][1=L,usedefault]{%
2985     \ifstrequal{#1}{L}{%
2986         \xright@appenditem{\csxdef{footnote@luatextextdir}{\the\textdir}}\to\
inserts@list%Know the dir of lemma
2987         \global\advance\insert@count \@ne%
2988         \xright@appenditem{\csxdef{footnote@luatexpardir}{\the\pardir}}\to\
inserts@list%Know the dir of lemma
2989         \global\advance\insert@count \@ne%
2990     }%
2991     {%
2992         \xright@appenditem{\csxdef{footnote@luatextextdir}{\the\textdir}}\to\
inserts@listR%Know the dir of lemma
2993         \global\advance\insert@countR \@ne%
2994         \xright@appenditem{\csxdef{footnote@luatexpardir}{\the\pardir}}\to\
inserts@listR%Know the dir of lemma
2995         \global\advance\insert@countR \@ne%
2996     }%
2997 }
2998 %

```

`\footnotelang@poly` `\footnotelang@poly` is called to remember the information about the language of a lemma when polyglossia is used.

```

2999 \newcommand*{\footnotelang@poly}[1][1=L,usedefault]{%
3000     \ifstrequal{#1}{L}{%
3001         \if@RTL%
3002             \xright@appenditem{\csxdef{footnote@dir}{@RTLtrue}}\to\
inserts@list%Know the language used in the lemma
3003             \global\advance\insert@count \@ne%
3004         \else
3005             \xright@appenditem{\csxdef{footnote@dir}{@RTLfalse}}\to\
inserts@list%Know the language of lemma

```

```

3006     \global\advance\insert@count \@ne%
3007     \fi%
3008     \xright@appenditem{{\csxdef{footnote@lang}{\expandonce\language\language}}}\
to\inserts@listR%Know the language of lemma
3009     \global\advance\insert@count \@ne%
3010     }%
3011     {%
3012     \if@RTL
3013         \xright@appenditem{{\csxdef{footnote@dir}{@RTLtrue}}}\to\
inserts@listR%Know the language of lemma
3014         \global\advance\insert@countR \@ne%
3015     \else
3016         \xright@appenditem{{\csxdef{footnote@dir}{@RTLfalse}}}\to\
inserts@listR%Know the language of lemma
3017         \global\advance\insert@countR \@ne%
3018     \fi
3019     \xright@appenditem{{\csxdef{footnote@lang}{\expandonce\language\language}}}\
to\inserts@listR%Know the language of lemma
3020     \global\advance\insert@countR \@ne%
3021     }%
3022 }
3023 %

```

### 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`.

```

3024 \newcommand*{\footplitskips}{%
3025   \interlinepenalty=\interfootnotelinepenalty
3026   \unless\ifl@dprintingpages%
3027     \floatingpenalty=\@MM%
3028   \fi%
3029   \splittopskip=\ht\strutbox \splitmaxdepth=\dp\strutbox
3030   \leftskip=\z@skip \rightskip=\z@skip}
3031
3032 %

```

**`\normalfootnoterule`** `\normalfootnoterule` is a standard footnote-rule macro, for use by a `footstart` macro: just the same as the `PLAIN`  $\TeX$  footnote rule.

```

3033 \let\normalfootnoterule=\footnoterule
3034 %

```

## 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.

```

3035 \newcommandx{\Xarrangement}[2][1,usedefault]{%
3036   \def\do##1{%
3037     \csname Xarrangement@#2\endcsname{##1}%
3038   }%
3039   \ifstrepty{#1}%
3040     {%
3041       \dolistloop{\@series}%
3042     }%
3043   {
3044     \docsvlist{#1}%
3045   }%
3046 }%
3047 %

```

### 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.

```
3048 \newcommand*{\Xarrangement@normal}[1]{%
3049   \csgdef{series@display#1}{normal}
3050   \expandafter\let\csname #1footstart\endcsname=\normalfootstart
3051   \expandafter\let\csname v#1footnote\endcsname=\normalvfootnote
3052   \expandafter\let\csname #1footfmt\endcsname=\normalfootfmt
3053   \expandafter\let\csname #1footgroup\endcsname=\normalfootgroup
3054   \expandafter\let\csname #1footnoterule\endcsname=%
3055                                   \normalfootnoterule
3056   \count\csname #1footins\endcsname=1000
3057   \dimen\csname #1footins\endcsname=\csuse{Xmaxhnotes@#1}
3058   \skip\csname #1footins\endcsname=\csuse{Xbeforenotes@#1}%
3059   \advance\skip\csname #1footins\endcsname by\csuse{Xafterrule@#1}%
3060   %
```

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.

```
3061   \csxdef{default@#1footins}{1000}%Use this to confine the notes to one
3062   side only
3063   %
```

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).

```
3063   \ifnoledgroup@else%
3064     \expandafter\let\csname mpv#1footnote\endcsname=\mpnormalvfootnote
3065     \expandafter\let\csname mp#1footgroup\endcsname=\mpnormalfootgroup
3066     \count\csname mp#1footins\endcsname=1000
```



```

3067 \dimen\csname mp#1footins\endcsname=\csuse{Xmaxhnotes@#1}
3068 \skip\csname mp#1footins\endcsname=\csuse{Xbeforenotes@#1}%
3069 \advance\skip\csname mp#1footins\endcsname by\csuse{Xafterrule@#1}%
3070 \fi
3071 }
3072
3073 %

```

**\normalvfootnote** We now begin a series of commands that do ‘normal’ footnote formatting: a format much like that implemented in PLAIN T<sub>E</sub>X, in which each footnote is a separate paragraph.

\normalvfootnote takes the series letter as #1 and the entire text of the footnote is #2. It does the \insert for this note, calling on the \footfmt macro for this note series to format the text of the note.

```

3074 \notbool{parapparatus}{\newcommand*{\newcommand}{\normalvfootnote}[2]{%
3075 \iftoggle{Xgroupbyline@#1}{%In the case we use \Xgroupbyline, the
insertion is done later, in \add@Xgroupbyline.
3076 \prepare@Xgroupbyline{#1}{#2}{\normalvfootnote@inserted}%
3077 }{%In the case we don't use \Xgroupbyline, the insertion is made directly
3078 \X@beforeinsertion{#1}%
3079 \insert\csname #1footins\endcsname{%
3080 \X@atbegininsertion{#1}%
3081 \normalvfootnote@inserted{#1}{#2}%
3082 }%
3083 }%
3084 }%
3085 %

```

**\normalvfootnote@inserted** The \normalvfootnote@inserted macro is expanded to the content to be add to a \insert for normal critical footnote.

```

3086 \notbool{parapparatus}{\newcommand*{\newcommand}{\normalvfootnote@inserted}[2]{%
3087 \nottoggle{Xgroupbyline@#1}{\noindent}{\csuse{Xhooknote@#1}%
3088 \csuse{Xnotefontsize@#1}%
3089 \footsplitskips
3090 \ifl@dpairing\ifl@dpaging\else%
3091 \setXnoteswidthliketwocolumns@#1}%
3092 \fi\fi%
3093 \setXnotespositionliketwocolumns@#1}%
3094 \spaceskip=\z@skip \xspaceskip=\z@skip%
3095 \csname #1footfmt\endcsname #2{#1}%
3096 }%
3097 %

```

**\X@beforeinsertion** \newcommand{\X@beforeinsertion}[1]{%

```

3099 \if@ledgroup\else%
3100 \insert@Xtxtbeforenotes{#1}%

```

```

3101 \fi%
3102 \csuse{Xbeforeinserting@#1}%
3103 }%
3104 %

```

```

\beforeinsertion@X05 \newcommand{\beforeinsertion@X}[1]{%
3106 \if@ledgroup\else%
3107 \insert@txtbeforenotesX{#1}%
3108 \fi%
3109 \csuse{beforeinsertingX@#1}%
3110 }%
3111 %

```

```

\X@atbegininsertion12 \newcommand{\X@atbegininsertion}[1]{%
3113 \hsize=\expandafter\dimexpr\csuse{Xwidth@#1}\relax%
3114 }%
3115 %

```

And somewhat different versions of `\normalvfootnote` and `\normalvfootnote@inserted` for minipages.

```

\mpnormalvfootnote16 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\mpnormalvfootnote}[2]{%
3117 \iftoggle{Xgroupbyline@#1}{%
3118 \prepare@Xgroupbyline{#1}{#2}{\mpnormalvfootnote@inserted}%
3119 }%
3120 {%
3121 \global\setbox\@nameuse{mp#1footins}%
3122 \vbox{%
3123 \unvbox\@nameuse{mp#1footins}%
3124 \mpnormalvfootnote@inserted{#1}{#2}%
3125 }%
3126 }%
3127 }%
3128
3129 %

```

```

\mpnormalvfootnote@inserted30 \newcommand{\mpnormalvfootnote@inserted}[2]{%
3131 \noindent\csuse{Xhooknote@#1}%
3132 \csuse{Xnotefontsize@#1}%
3133 \hsize\columnwidth%
3134 \@parboxrestore%
3135 \color@begingroup%
3136 \csname #1footfmt\endcsname #2{#1}\color@endgroup%
3137 }%
3138 %

```

`\normalfootfmt` `\normalfootfmt` is a ‘normal’ macro to take the footnote line and page number information (see V.9 p. 106), 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.

```

3139 \notbool{parapparatus@}{\newcommand*{\newcommand}{\normalfootfmt}[4]{%
3140   \xstorelineinfo{#1}{#4}%
3141   \nottoggle{Xgroupbyline@#4}{\xledsetnormalparstuff{#4}}{}%
3142   \hangindent=\csuse{Xhangindent@#4}%
3143   \everypar{\hangindent=\csuse{Xhangindent@#4}}%
3144   \nottoggle{Xgroupbyline@#4}{\rule{z@}{\splittopskip}}{}%
3145   {\printlinefootnote{#1}{#4}}%
3146   \print@lemma{#1}{#2}{#4}%
3147   \csuse{Xwrapcontent@#4}{#3}%
3148   \nottoggle{Xgroupbyline@#4}{\strut\par}{}%
3149 }%
3150 %

```

`\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. T<sub>E</sub>X 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`.

```

3151 \newcommand*{\normalfootstart}[1]{%
3152 %

```

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. 239. Here is part of this algorithm, when the block of notes are ready to be printed.

```

3153 \ifdimequal{0pt}{\Xprenotes@}{}%
3154   {%
3155     \iftoggle{Xprenotes@}{%
3156       \togglefalse{Xprenotes@}%

```

```

3157         \skip\csname #1footins\endcsname=%
3158         \glueexpr\csuse{Xprenotes@}+\csuse{Xafterrule@#1}\relax%
3159     }%
3160 }%
3161 }%
3162 \vskip\skip\csname #1footins\endcsname%
3163 %

```

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. 237 for the general description of the problem.

```

3164 \leftskipOpt \rightskipOpt
3165 \ifl@dpairing\else%
3166     \hsize=\old@hsize%
3167 \fi%
3168 \setXnoteswidthliketwocolumns@{#1}%
3169 \setXnotespositionliketwocolumns@{#1}%
3170 %

```

And now, print the footnote's rule to finish the footnote's introduction.

```

3171 \print@Xfootnoterule{#1}%
3172 }%
3173 %

```

`\normalfootgroup` `\normalfootgroup` is a standard footnote-grouping macro: it sends the contents of the footnote-insert box to the output page without alteration.

```

3174 \newcommand*{\normalfootgroup}[1]{%
3175     \csuse{Xhookgroup@#1}%
3176     \unvbox\csname #1footins\endcsname%
3177     \hsize=\old@hsize%
3178 }%
3179 %
3180 %

```

`\mpnormalfootgroup` A somewhat different version for minipages. Note that, in this case, we do not make distinctions between the `\Xfootgroup` and `\Xfootstarts` macros.

```

3181 \unless\ifnoledgroup@
3182 \newcommand*{\mpnormalfootgroup}[1]{%
3183     \vskip\skip\@nameuse{mp#1footins}
3184     \ifl@dpairing\ifparledgroup%
3185         \leavevmode\marks\parledgroup@{begin}%
3186         \marks\parledgroup@series{#1}%
3187         \marks\parledgroup@type{Xfootnote}%
3188     \fi\fi\normalcolor%
3189     \ifparledgroup%
3190         \ifl@dpairing%
3191         \else%

```

```

3192 \setXnoteswidthliketwocolumns@{#1}%
3193 \setXnotespositionliketwocolumns@{#1}%
3194 \print@Xfootnoterule{#1}%%
3195 \fi%
3196 \else%
3197 \setXnoteswidthliketwocolumns@{#1}%
3198 \setXnotespositionliketwocolumns@{#1}%
3199 \print@Xfootnoterule{#1}%%
3200 \fi%
3201 \setlength{\parindent}{0pt}
3202 \csuse{Xbhookgroup@#1}%
3203 \unvbox\csname mp#1footins\endcsname}}
3204 \fi
3205 %

```

### 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 T<sub>E</sub>X of ordinary size may not be able to handle more than about 100 notes of this kind on a page.

`\Xarrangement@paragraph` The `\Xarrangement@paragraph` macro sets up everything for one series of the footnotes so that they will be paragraphed; it takes the series letter as argument. We include the setting of `\count\footins` to 1000 for the footnote series just in case user is switching to paragraphed footnotes after having columnar ones, since they change this value (see below).

The argument of `\Xarrangement@footparagraph` is the letter denoting the series of notes to be paragraphed.

```

3206 \newcommand*{\Xarrangement@paragraph}[1]{%
3207 \csgdef{series@display#1}{paragraph}
3208 \expandafter\let\csname #1footstart\endcsname=\parafootstart
3209 \expandafter\let\csname v#1footnote\endcsname=\paravfootnote
3210 \expandafter\let\csname #1footfmt\endcsname=\parafootfmt
3211 \expandafter\let\csname #1footgroup\endcsname=\parafootgroup
3212 \count\csname #1footins\endcsname=1000
3213 \csxdef{default@#1footins}{1000}%Use this to confine the notes to one
side only
3214 \dimen\csname #1footins\endcsname=\csuse{Xmaxhnotes@#1}
3215 \skip\csname #1footins\endcsname=\csuse{Xbeforenotes@#1}%
3216 \advance\skip\csname #1footins\endcsname by\csuse{Xafterrule@#1}%
3217 \para@footsetup{#1}
3218 %

```

And the extra setup for minipages.

```

3219 \ifnoledgroup@{\else
3220 \expandafter\let\csname mpv#1footnote\endcsname=\mpparavfootnote

```

```

3221 \expandafter\let\csname mp#1footgroup\endcsname=\mpparafootgroup
3222 \count\csname mp#1footins\endcsname=1000
3223 \dimen\csname mp#1footins\endcsname=\csuse{Xmaxhnotes@#1}
3224 \skip\csname mp#1footins\endcsname=\csuse{Xbeforenotes@#1}%
3225 \advance\skip\csname mp#1footins\endcsname by\csuse{Xafterrule@#1}%
3226 \fi
3227 }
3228 %

```

**\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.

```

3229 \providecommand{\footfudgefiddle}{64}
3230 %

```

**\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  $\mathcal{E}\mathcal{T}\mathcal{X}$  not `\hsize`. Peter Wilson have also included `\footfudgefiddle`.

```

3231 \newcommand*{\para@footsetup}[1]{\csuse{Xhookgroup@#1}\csuse{
Xnotefontsize@#1}
3232 \setXnoteswidthliketwocolumns@{#1}%
3233 \ifcsempy{Xwidth@#1}%
3234 {}%
3235 {\columnwidth=\expandafter\dimexpr\csuse{Xwidth@#1}\relax}%
3236 \dimen0=\baselineskip
3237 \multiply\dimen0 by 1024
3238 \divide \dimen0 by \columnwidth \multiply\dimen0 by \footfudgefiddle\
relax
3239 \csxdef{#1footfudgefactor}{%
3240 \expandafter\strip@pt\dimen0 }}
3241
3242 %

```

`\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.

```

3243 \newcommand*{\parafootstart}[1]{%
3244   \rightskip=0pt \leftskip=0pt%
3245   \nottoggle{Xparindent@#1}{\parindent=\z@}{}%
3246   \ifdimequal{0pt}{\Xprenotes@}{}%
3247   {%
3248     \iftoggle{Xprenotes@}{%
3249       \togglefalse{Xprenotes@}%
3250       \skip\csname #1footins\endcsname=%
3251       \glueexpr\csuse{Xprenotes@}+\csuse{Xafterrule@#1}\relax%
3252     }%
3253   }%
3254 }%
3255 \vskip\skip\csname #1footins\endcsname%
3256 \setXnoteswidthliketwocolumns@{#1}%
3257 \setXnotespositionliketwocolumns@{#1}%
3258 \print@Xfootnoterule{#1}%
3259 \let\bidirTL@everypar\@empty%
3260 \noindent\leavevmode}
3261 %

```

**\parafootnote** `\parafootnote` 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

<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).

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. 187 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.

```

3262 \newcommand*{\paravfootnote}[2]{%
3263   \csuse{Xbeforeinserting@#1}%
3264   \insert\csname #1footins\endcsname
3265   \bgroup
3266     \csuse{Xnotefontsize@#1}
3267     \footplitskips
3268     \setbox0=\vbox{\hsize=\maxdimen%
3269       \let\bid@RTL@everypar\@empty%
3270       \insert@Xtxtbeforenotes{#1}%
3271       \noindent\csuse{Xhooknote@#1}%
3272       \csname #1footfmt\endcsname #2{#1}}%
3273     \setbox0=\hbox{\Xunvxh{0}{#1}}%
3274     \dp0=0pt
3275     \ht0=\csname #1footfudgefactor\endcsname\wd0
3276   %

```

Here we produce the contents of the footnote from box 0, and add a penalty of 0 between boxes in this insert.

```

3277   \if@RTL\noindent \leavevmode\fi\box0%
3278   \penalty0
3279 \egroup}
3280
3281 %

```

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

<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.



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.

```

3282 \newcommand*{\mpparavfootnote}[2]{%
3283   \global\setbox\@nameuse{mp#1footins}\vbox{%
3284     \unvbox\@nameuse{mp#1footins}%
3285     \csuse{Xnotefontsize@#1}
3286     \footplitskips
3287     \setbox0=\vbox{\hsize=\maxdimen%
3288       \let\bidir@RTL@everypar\@empty%
3289       \insert@Xtxtbeforenotes{#1}%
3290       \noindent\color@begingroup%
3291       \csuse{Xhooknote@#1}%
3292       \csname #1footfmt\endcsname #2{#1}\color@endgroup}%
3293     \setbox0=\hbox{\Xunvxh{0}{#1}}%
3294     \dp0=\z@
3295     \ht0=\csname #1footfudgefactor\endcsname\wd0
3296     \box0
3297     \penalty0
3298   }}
3299
3300 %

```

`\Xunvxh` Here is (modified) Michael's definition of `\unvxh`, used above. Michael's macro also takes care to remove some unwanted penalties and glue that TeX automatically attaches to the end of paragraphs. When TeX finishes a paragraph, it throws away any remaining glue, and then tacks on the following items: a `\penalty` of 10000, a `\parfillskip` and a `\rightskip` (*The TeXbook*, pp. 99–100). `\unvxh` cancels these unwanted paragraph-final items using `\unskip` and `\unpenalty`.

```

3301 \newcommand*{\Xunvxh}[2]{%
3302   \setbox0=\vbox{\unvbox#1%
3303     \global\setbox1=\lastbox}%
3304   \unhbox1
3305   \unskip           % remove \rightskip,
3306   \unskip           % remove \parfillskip,
3307   \unpenalty        % remove \penalty of 10000,
3308   \hskip\csuse{Xafternote@#2}\relax}% add the glue to go between the notes
3309
3310 %

```

`\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).

```

3311 \newcommand*{\parafootfmt}[4]{%
3312   \Xstorelineinfo{#1}{#4}%
3313   \Xinsertparafootsep{#4}%
3314   \ledsetnormalparstuff@common%
3315   \printlinefootnote{#1}{#4}%
3316   \print@lemma{#1}{#2}{#4}%
3317   \csuse{Xwrapcontent@#4}{#3}%
3318   \penalty-10 }
3319 %

```

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.

```

3320 \newcommand*{\parafootgroup}[1]{%
3321   \hsize=\expandafter\dimexpr\csuse{Xwidth@#1}\relax%
3322   \unvbox\csname #1footins\endcsname
3323   \ifcsstring{Xragged@#1}{L}{\RaggedLeft}{}%
3324   \ifcsstring{Xragged@#1}{R}{\RaggedRight}{}%
3325   \makehboxofhboxes
3326   \setbox0=\hbox{\unhbox0 \removehboxes}%
3327   \csuse{Xbhookgroup@#1}%
3328   \csuse{Xnotefontsize@#1}%
3329   \unhbox0\par%
3330   \global\hsize=\old@hsize%
3331   }%
3332
3333 %

```

`\mpparafootgroup` The minipage version.

```

3334 \newcommand*{\mpparafootgroup}[1]{%
3335   \setXnoteswidthliketwocolumns@{#1}%
3336   \vskip\skip\@nameuse{mp#1footins}
3337   \ifl@dpairing\ifparledgroup%
3338     \leavevmode\marks\parledgroup@{begin}%
3339     \marks\parledgroup@series{#1}%

```

```

3340 \marks\parledgroup@type{Xfootnote}%
3341 \fi\fi\normalcolor
3342 \ifparledgroup%
3343 \ifl@dpairing%
3344 \else%
3345 \setXnoteswidthliketwocolumns@{#1}%
3346 \setXnotespositionliketwocolumns@{#1}%
3347 \print@Xfootnoterule{#1}%
3348 \fi%
3349 \else%
3350 \setXnoteswidthliketwocolumns@{#1}%
3351 \setXnotespositionliketwocolumns@{#1}%
3352 \print@Xfootnoterule{#1}%
3353 \fi%
3354 \unvbox\csname mp#1footins\endcsname
3355 \ifcsstring{Xragged@#1}{L}{\RaggedLeft}{}%
3356 \ifcsstring{Xragged@#1}{R}{\RaggedRight}{}%
3357 \makehboxofhboxes
3358 \setbox0=\hbox{\unhbox0 \removehboxes}%
3359 \csuse{Xhookgroup@#1}%
3360 \csuse{Xnotefontsize@#1}%
3361 \nottoggle{Xparindent@#1}{\parindent=\z@}{}%
3362 \unhbox0\par}}
3363
3364 %

```

And finally, the two macros which are required to transform the long horizontal box stored in the insert' box to a printable text.

```

\makehboxofhboxes65 \newcommand*{\makehboxofhboxes}{\setbox0=\hbox{}}%
\removehboxes66 \loop
3367 \unpenalty
3368 \setbox2=\lastbox
3369 \ifhbox2
3370 \setbox0=\hbox{\box2\unhbox0}%
3371 \repeat}
3372
3373 \newcommand*{\removehboxes}{\setbox0=\lastbox
3374 \ifhbox0{\removehboxes}\unhbox0 \fi}
3375
3376 %

```

**Insertion of the footnotes separator** The command `\Xinsertparafootsep{<series>}` must be called at the beginning of `\parafootftm`.

```

\prevpage@num77 \newcommand{\Xinsertparafootsep}[1]{%
\Xinsertparafootsep78 \ifledRcol@%
3379 \ifnumequal{\csuse{#1prevpage@numR}}{\page@numR}%

```

```

3380     {\ifcsdef{prevline#1}% Be sur \prevline#1 exists.
3381       {\ifcsequal{prevline#1}{lineinfo@}%
3382         {\ifcempty{Xsymlinenum@#1}{\csuse{Xparafootsep@#1}}{}}%
3383         {\csuse{Xparafootsep@#1}}%
3384       }%
3385       {\csuse{Xparafootsep@#1}}%
3386     }%
3387     {}%
3388     \global\csname #1prevpage@numR\endcsname=\page@numR%
3389 \else%
3390   \ifnumequal{\csuse{#1prevpage@num}}{\page@num}%
3391     {\ifcsdef{prevline#1}% Be sur \prevline#1 exists.
3392       {\ifcsequal{prevline#1}{lineinfo@}%
3393         {\ifcempty{Xsymlinenum@#1}{\csuse{Xparafootsep@#1}}{}}%
3394         {\csuse{Xparafootsep@#1}}%
3395       }%
3396       {\csuse{Xparafootsep@#1}}%
3397     }%
3398     {}%
3399     \global\csname #1prevpage@num\endcsname=\page@num%
3400 \fi%
3401 }
3402 %

```

### 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 `\rigidbalance` macro is taken from *The TeXbook*, p. 397, with a slight change to the syntax of the arguments so that they do not depend on white space. Note also the extra unboxing in `\splitoff`, which allows the new `\vbox` to have its natural height as it goes into the alignment.

The  $\text{\LaTeX}$  `\line` macro has no relationship to the TeX `\line`. The  $\text{\LaTeX}$  equivalent is `\@@line`.

We do not call directly `\rigidbalance`, but we call `\Xrigidbalance` for critical notes and `\rigidbalanceX` for familiar notes. Both of them call `\rigidbalance`.

```

3403 \newcount\@k \newdimen\@h
3404 \newcommand*{\Xrigidbalance}[3]{%
3405   \hsize=\expandafter\dimexpr\csuse{Xwidth@\@currentseries}\relax%
3406   \rigidbalance{#1}{#2}{#3}%
3407 }%
3408
3409 \newcommand*{\rigidbalanceX}[3]{%
3410   \hsize=\expandafter\dimexpr\csuse{widthX@\@currentseries}\relax%
3411   \rigidbalance{#1}{#2}{#3}%

```

```

3412 }%
3413
3414 \newcommand*{\rigidbalance}[3]{%
3415   \setbox0=\box#1 \@k=#2 \@h=#3%
3416   \@@line{\splittopskip=\@h \vbadness=\@M \hfilneg
3417   \valign{##\vfil\cr\dosplits}}}}
3418
3419 \newcommand*{\dosplits}{\ifnum\@k>0 \noalign{\hfil}\splitoff
3420   \global\advance\@k-1\cr\dosplits\fi}
3421
3422 \newcommand*{\splitoff}{\dimen0=\ht0
3423   \divide\dimen0 by\@k \advance\dimen0 by\@h
3424   \setbox2 \vsplit0 to \dimen0
3425   \unvbox2 }
3426
3427 %

```

### Three columns

```

\Xarrangement@threecol 28 \newcommand*{\Xarrangement@threecol}[1]{%
3429   \csgdef{series@display#1}{threecol}
3430   \expandafter\let\csname v#1footnote\endcsname=\threecolvfootnote
3431   \expandafter\let\csname #1footfmt\endcsname=\threecolfootfmt
3432   \expandafter\let\csname #1footgroup\endcsname=\threecolfootgroup
3433   \dimen\csname #1footins\endcsname=\csuse{Xmaxhnotes@#1}%
3434   \skip\csname #1footins\endcsname=\csuse{Xbeforenotes@#1}%
3435   \advance\skip\csname #1footins\endcsname by\csuse{Xafterrule@#1}%
3436   \threecolfootsetup{#1}
3437   %

```

The additional setup for minipages.

```

3438   \ifnoledgroup@else
3439     \expandafter\let\csname mpv#1footnote\endcsname=\mpnormalvfootnote
3440     \expandafter\let\csname mp#1footgroup\endcsname=\mpthreecolfootgroup
3441     \skip\csname mp#1footins\endcsname=\csuse{Xbeforenotes@#1}%
3442     \advance\skip\csname mp#1footins\endcsname by\csuse{Xafterrule@#1}%
3443     \mpthreecolfootsetup{#1}
3444   \fi
3445 }
3446
3447 %

```

The `\footstart` and `\footnoterule` macros for these notes assume the normal values (XIII.6.2 p. 187 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.

```

3448 \newcommand*{\threecolfootsetup}[1]{%
3449   \count\csname #1footins\endcsname 333
3450   \csxdef{default@#1footins}{333}%Use this to confine the notes to one
side only
3451   \multiply\dimen\csname #1footins\endcsname \thr@@}
3452   %

```

`\mpthreecolfootsetup` The setup for minipages.

```

3453 \newcommand*{\mpthreecolfootsetup}[1]{%
3454   \count\csname mp#1footins\endcsname 333
3455   \multiply\dimen\csname mp#1footins\endcsname \thr@@}
3456   %
3457   %

```

`\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).

```

3458 \notbool{parapparatus@}{\newcommand*}{\newcommand*}{\threecolvfootnote}[2]{%
3459   \iftoggle{Xgroupbyline@#1}{%
3460     \prepare@Xgroupbyline{#1}{#2}{\threecolvfootnote@inserted}%
3461   }%
3462   {%
3463     \X@beforeinsertion{#1}%
3464     \insert\csname #1footins\endcsname{%
3465       \threecolvfootnote@inserted{#1}{#2}%
3466     }%
3467   }%
3468   %
3469   %

```

```

\threecol@begin@insert \notbool{parapparatus@}{\newcommand*}{\newcommand}{\
threecol@begin@insert}[2]{%
3471 \hspace=\expandafter\dimexpr\csuse{Xwidth@#1}\relax%
3472 \noindent\csuse{Xhooknote@#1}%
3473 \csuse{Xnotefontsize@#1}%
3474 \footnoteskip%
3475 \csname #1footfmt\endcsname #2{#1}%
3476 }%
3477 %

```

**\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.

```

3478 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\threecolfootfmt}[4]{%
3479 \Xstorelineinfo{#1}{#4}%
3480 \threecol@begin@insert{#4}%
3481 \hspace{\parindent}%
3482 \printlinefootnote{#1}{#4}%
3483 \print@lemma{#1}{#2}{#4}%
3484 \csuse{Xwrapcontent@#4}{#3}%
3485 \nottoggle{Xgroupbyline@#4}%
3486 {\strut\par\allowbreak}%
3487 }%
3488 %
3489 %

```

**\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.

```

3490 \newcommand{\threecol@begin@insert}[1]{%
3491 \normal@pars%
3492 \nottoggle{Xgroupbyline@#1}%
3493 {\hspace \csuse{Xsizethreecol@#1}}%
3494 }%
3495 \nottoggle{Xparindent@#1}{\parindent=\z@}{%
3496 \tolerance=5000%
3497 \hangindent=\csuse{Xhangindent@#1}%
3498 \everypar{\hangindent=\csuse{Xhangindent@#1}}%
3499 \@tempdima=\parindent%
3500 \csuse{Xcolalign@#1}%
3501 \parindent=\@tempdima%
3502 \strut%
3503 }%
3504 %

```

**\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.

```

3505 \newcommand*\threecolfootgroup}[1]{%
3506   \csuse{Xbhookgroup@#1}\par%
3507   \splittopskip=\ht\strutbox
3508   \expandafter
3509   \Xrigidbalance\csname #1footins\endcsname \thr@@ \splittopskip}
3510 %

```

`\mpthreecolfootgroup` The setup for minipages.

```

3511 \newcommand*\mpthreecolfootgroup}[1]{%
3512   \vskip\skip\@nameuse{mp#1footins}
3513   \ifl@dpairing\ifparledgroup%
3514     \leavevmode\marks\parledgroup@{begin}%
3515     \marks\parledgroup@series{#1}%
3516     \marks\parledgroup@type{Xfootnote}%
3517   \fi\fi\normalcolor
3518   \ifparledgroup%
3519     \ifl@dpairing%
3520     \else%
3521       \setXnoteswidthliketwocolumns@{#1}%
3522       \setXnotespositionliketwocolumns@{#1}%
3523       \print@Xfootnoterule{#1}%
3524     \fi%
3525   \else%
3526     \setXnoteswidthliketwocolumns@{#1}%
3527     \setXnotespositionliketwocolumns@{#1}%
3528     \print@Xfootnoterule{#1}%
3529   \fi%
3530   \csuse{Xbhookgroup@#1}\par%
3531   \splittopskip=\ht\strutbox
3532   \expandafter
3533   \Xrigidbalance\csname mp#1footins\endcsname \thr@@ \splittopskip}}
3534 %
3535 %

```

## Two columns

```

\Xarrangement@twocol 36 \newcommand*\Xarrangement@twocol}[1]{%
3537   \csgdef{series@display#1}{twocol}
3538   \expandafter\let\csname v#1footnote\endcsname=\twocolvfootnote
3539   \expandafter\let\csname #1footfmt\endcsname=\twocolfootfmt

```



```

3540 \expandafter\let\csname #1footgroup\endcsname=\twocolfootgroup
3541 \dimen\csname #1footins\endcsname=\csuse{Xmaxhnotes@#1}%
3542 \skip\csname #1footins\endcsname=\csuse{Xbeforenotes@#1}%
3543 \advance\skip\csname #1footins\endcsname by\csuse{Xafterrule@#1}%
3544 \twocolfootsetup{#1}
3545 %

```

The additional setup for minipages.

```

3546 \ifnoledgroup@else
3547   \expandafter\let\csname mpv#1footnote\endcsname=\mpnormalvfootnote
3548   \expandafter\let\csname mp#1footgroup\endcsname=\mptwocolfootgroup
3549   \skip\csname mp#1footins\endcsname=\csuse{Xbeforenotes@#1}%
3550   \advance\skip\csname mp#1footins\endcsname by\csuse{Xafterrule@#1}%
3551   \mptwocolfootsetup{#1}
3552 \fi
3553 }
3554
3555 %

```

`\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 `\hspace`.

`\twocolvfootnote`  
`\twocolvfootnote@inserted`  
`\twocolfootfmt`  
`\twocolfootgroup`

```

3556 \newcommand*{\twocolfootsetup}[1]{%
3557   \count\csname #1footins\endcsname 500
3558   \csxdef{default@#1footins}{500}%
3559   \multiply\dimen\csname #1footins\endcsname \tw@}
3560 %

3561 \notbool{parapparatus@}{\newcommand*{\twocolvfootnote}[2]{%
3562   \iftoggle{Xgroupbyline@#1}{%
3563     \prepare@Xgroupbyline{#1}{#2}{\twocolvfootnote@inserted}%
3564   }{%
3565     \X@beforeinsertion{#1}%
3566     \insert\csname #1footins\endcsname{%
3567       \twocolvfootnote@inserted{#1}{#2}%
3568     }%
3569   }%
3570 }%

3571 \notbool{parapparatus@}{\newcommand*{\twocolvfootnote@inserted}[2]{%
3572   \hspace=\expandafter\dimexpr\csuse{Xwidth@#1}\relax%
3573   \noindent\csuse{Xhooknote@#1}%
3574   \csuse{Xnotefontsize@#1}%
3575   \footssplitsskip%
3576   \csname #1footfmt\endcsname #2{#1}%
3577 }%
3578 %

```

```

3579 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\twocolfootfmt}[4]{% 4th
    arg is optional, for backward compatibility
3580 \Xstorelineinfo{#1}{#4}%
3581 \twocol@begin@insert{#4}%
3582 \hspace{\parindent}%
3583 \printlinefootnote{#1}{#4}%
3584 \print@lemma{#1}{#2}{#4}%
3585 \csuse{Xwrapcontent@#4}{#3}%
3586 \nottoggle{Xgroupbyline@#4}%
3587 {\strut\par\allowbreak}%
3588 {}%
3589 }%
3590 %

3591 \newcommand{\twocol@begin@insert}[1]{%
3592 \normal@pars%
3593 \hsize \csuse{Xhsizetwocol@#1}%
3594 \nottoggle{Xparindent@#1}{\parindent=\z@}{}%
3595 \tolerance=5000%
3596 \hangindent=\csuse{Xhangindent@#1}%
3597 \everypar{\hangindent=\csuse{Xhangindent@#1}}%
3598 \@tempdima=\parindent%
3599 \csuse{Xcolalign@#1}%
3600 \parindent=\@tempdima%
3601 \strut%
3602 }%
3603
3604 \newcommand*{\twocolfootgroup}[1]{%
3605 \csuse{Xbhookgroup@#1}\par%
3606 \splittopskip=\ht\strutbox
3607 \expandafter
3608 \Xrigidbalance\csname #1footins\endcsname \tw@ \splittopskip}
3609
3610 %

```

`\mptwocolfootsetup` The versions for minipages.

`\mptwocolfootgroup`

```

3611 \newcommand*{\mptwocolfootsetup}[1]{%
3612 \count\csname mp#1footins\endcsname 500
3613 \multiply\dimen\csname mp#1footins\endcsname \tw@}
3614 %

3615 \newcommand*{\mptwocolfootgroup}[1]{%
3616 \vskip\skip\@nameuse{mp#1footins}
3617 \ifl@dpairing\ifparledgroup%
3618 \leavevmode\marks\parledgroup@{begin}%
3619 \marks\parledgroup@series{#1}%
3620 \marks\parledgroup@type{Xfootnote}%
3621 \fi\fi\normalcolor
3622 \ifparledgroup%

```

```

3623 \ifl@dpairing%
3624 \else%
3625   \setXnoteswidthliketwocolumns@{#1}%
3626   \setXnotespositionliketwocolumns@{#1}%
3627   \print@Xfootnoterule{#1}%
3628 \fi%
3629 \else%
3630   \setXnoteswidthliketwocolumns@{#1}%
3631   \setXnotespositionliketwocolumns@{#1}%
3632   \print@Xfootnoterule{#1}%
3633 \fi%
3634 \csuse{Xbhookgroup@#1}\par%
3635 \splittopskip=\ht\strutbox
3636 \expandafter
3637 \Xrigidbalance\csname mp#1footins\endcsname \tw@ \splittopskip}}
3638 %
3639 %

```

## XIII.7 Critical notes presentation

Here, we define some commons macro which are used in order to print a critical notes, that is a note with 1) line number 2) lemma 3) lemma separator 4) text associated to the lemma.

### XIII.7.1 Font tools

`\endashchar` The fonts that are used for printing notes might not have the character mapping we expect: for example, the Computer Modern font that contains old-style numerals does not contain an en-dash or square brackets, and its period and comma are in odd locations. To allow use of the standard footnote macros with such fonts, we use the following macros for certain characters.

The `\endashchar` macro is simply an en-dash from the normal font and is immune to changes in the surrounding font. The same goes for the full stop. These two are used in `\printlines`. The right bracket macro is the same again; it crops up in `\normalfootfmt` and the other footnote macros for controlling the format of the footnotes.

Note that these commands are not directly called by `reledmac`, but are enclosed as default value of specific hooks. Consequently, people should not redefine them, but use instead the `\Xlinrangeseparator`, `\Xendlinrangeseparator`, `\Xsublinesep`, `\Xendsublinesep` and `\Xlemmaseparator` macros.

With `polyglossia`, each critical note has a `\footnote@lang` which shows the language of the lemma, and which can be used to switch the bracket from right to left.

```

3640 \def\endashchar{\textnormal{--}}
3641
3642 \newcommand*{\fullstop}{\textnormal{.}}
3643 \def\Xsublinesep@side{\fullstop}
3644

```

```

3645 \newcommand*{\rbracket}{\textnormal{%
3646   \csuse{text\csuse{footnote@lang}}{%
3647     \ifluatex%
3648       \ifdefstring{\footnote@luatextextdir}{TRT}{\thinspace[]{\thinspace
3649     }%
3650     \else%
3651       \thinspace}%
3652     \fi}%
3653 }%
3654 %
3655 %

```

### XIII.7.2 Pstart number in footnote

**\printpstart** The `\printpstart` macro prints the pstart number for a note.

```

3656 \newcommand{\printpstart}[0]{%
3657   \ifboolexpr{bool{!@dpairing} or bool{!@dprintingpages} or bool{
3658     !@dprintingcolumns}}{%
3659     \ifledRcol%
3660       \thepstartR%
3661     \else%
3662       \thepstartL%
3663     \fi%
3664   }%
3665   \thepstart%
3666 }%
3667 %

```

### XIII.7.3 Lemma printing

**\print@lemma** `\print@lemma` is called inside critical footnotes to print the lemma and the lemma separator (#1: line number and font information, #2: lemma, #3: series).

```

3668 %
3669 \newcommand{\print@lemma}[3]{%
3670   \bgroup%
3671   \nottoggle{Xlemmadisablefontselection@#3}%
3672   {\select@lemmafont#1|}%
3673   }%
3674   \bgroup%
3675   \csuse{Xlemmafont@#3}%Deprecated
3676   \csuse{Xwraplemma@#3}{#2}%
3677   \egroup%
3678   \egroup%
3679   \iftoggle{nosep@}{%
3680     \hskip\csuse{Xinplaceoflemmaseparator@#3}%

```

```

3681 \relax%
3682 }%
3683 {\ifcempty{Xlemmaseparator@#3}%
3684   {%
3685     \hskip\csuse{Xinplaceoflemmaseparator@#3}%
3686     \relax%
3687   }%
3688   {%
3689     \nobreak%
3690     \hskip\csuse{Xbeforelemmaseparator@#3}%
3691     \csuse{Xlemmaseparator@#3}%
3692     \hskip\csuse{Xafterlemmaseparator@#3}%
3693     \relax%
3694   }%
3695 }%
3696 }%
3697 %

```

#### XIII.7.4 Line number printing

**\Xstorelineinfo** The `\Xstorelineinfo` macro is used to store some data about line number of the current critical footnote, data which will be reused later for the `\Xnumberonlyfirstinline` and related setting.

#1 footnote specification for the current footnote ; #2 footnote series.

```

3698 \newcommand{\Xstorelineinfo}[2]{%
3699   \l@dp@rsefootspec#1|}%
3700   \iftoggle{Xnumberonlyfirstintwolines@#2}{%
3701     \xdef\lineinfo@{\l@dparsedstartline - \l@dparsedstartsub - \
3702       \l@dparsedendline - \l@dparsedendsub}%
3703     {%
3704       \xdef\lineinfo@{\l@dparsedstartline - \l@dparsedstartsub}%
3705     }%
3706   }%
3707   %

```

**\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`.

```

3708 \newcommand{\printlinefootnote}[2]{%
3709   \iftoggle{nonum@}{%Try if the line number must printed for this specific
3710     not (by default, yes)
3711     \hspace{\csuse{Xinplaceofnumber@#2}}}%
3712   {%
3713     {%

```

```

3714 \iftoggle{Xnonumber@#2}%Try if the line number must printed (by
default, yes)
3715 {%
3716 \hspace{\csuse{Xinplaceofnumber@#2}}%
3717 }%
3718 {%
3719 \iftoggle{Xnumberonlyfirstinline@#2}% If for this series the
line number must be printed only in the first time.
3720 {%
3721 \ifcsdef{prevline#2}%
3722 {%Be sure the \prevline exists.
3723 \ifcsequal{prevline#2}{\lineinfo@}%Try it
3724 {%
3725 \ifcsequal{Xsymlinenumber@#2}%
3726 {%
3727 \hspace{\csuse{Xinplaceofnumber@#2}}%
3728 }%
3729 {\printsymlinefootnotearea{#2}}%
3730 }%
3731 {%
3732 \printlinefootnotearea{#1}{#2}%
3733 }%
3734 }%
3735 {%
3736 \printlinefootnotearea{#1}{#2}%
3737 }%
3738 }%
3739 {%
3740 \printlinefootnotearea{#1}{#2}%
3741 }%
3742 \csxdef{prevline#2}{\lineinfo@}%
3743 }%
3744 }%
3745 }%
3746 }%
3747 }
3748 %

```

**\printsymlinefootnotearea** This macro prints the space before the line symbol, changes the font, when prints the line symbol and the space after it.

```

3749 \newcommand{\printsymlinefootnotearea}[1]{%
3750 \hspace{\csuse{Xbeforesymlinenumber@#1}}%
3751 \csuse{Xnotenumfont@#1}%
3752 \ifdimequal{\csuse{Xboxsymlinenumber@#1}}{\z@}%
3753 {\csuse{Xsymlinenumber@#1}}%
3754 {\hbox to \csuse{Xboxsymlinenumber@#1}%
3755 {\csuse{Xsymlinenumber@#1}\hfill}%
3756 }%
3757 \hspace{\csuse{Xaftersymlinenumber@#1}}%

```

```

3758 }%
3759 %

```

**\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.)

```

3760 \newcommand{\printlinefootnotearea}[2]{%
3761   \printXbeforenumber{#2}%
3762   \csuse{Xnotenumfont@#2}%
3763   \boxfootnotenumbers{#1}{#2}%
3764   \printXafternumber{#2}%
3765 }%
3766 %

```

**\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.

```

3767 \newcommand{\boxfootnotenumbers}[2]{%
3768   \ifdimequal{\csuse{Xboxlinenum@#2}}{0pt}{%
3769     \printlinefootnotenumbers{#1}{#2}%
3770   }%
3771   {%
3772     \hbox to \csuse{Xboxlinenum@#2}%
3773     {%
3774       \IfSubStr{RC}{\csuse{Xboxlinenumalign@#2}}{\hfill}{}%
3775       \printlinefootnotenumbers{#1}{#2}%
3776       \IfSubStr{LC}{\csuse{Xboxlinenumalign@#2}}{\hfill}{}%
3777     }%
3778   }%
3779 }%
3780 %

```

**\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.

```

3781 \newcommand{\printlinefootnotenumbers}[2]{%
3782   \xdef\@currentseries{#2}%
3783   \ifboolexpr{%
3784     (togl{Xpstart@#2} and bool{numberpstart})%
3785     or togl{Xpstarteverytime@#2}}%
3786   {\printpstart}{}%
3787   \iftoggle{Xstanza@#2}{%
3788     \ifnumberstanza%
3789     \printstanza%
3790     \csuse{Xstanzaseparator@#2}%

```

```

3791 \fi%
3792 }{}%
3793 \iftoggle{Xonlypstart@#2}{%
3794 \csuse{Xtxtbeforenumber@#2}%
3795 \printlines#1|\ifledRcol@{\Rlineflag\fi}}%
3796 }%
3797 %

```

**\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.)

```

3798 \newcommand{\printXbeforenumber}[1]{%
3799 \hspace{\csuse{Xbeforenumber@#1}}%
3800 }%
3801 %

```

**\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

```

3802 \newcommand{\printXafternumber}[1]{%
3803 \iftoggle{Xnonbreakableafternumber@#1}{\nobreak}{}%
3804 \hspace{\csuse{Xafternumber@#1}}%
3805 }%
3806 %

```

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. 106: the starting page, line, and sub-line numbers, followed by the ending page, line, and sub-line numbers, and then the font specifier for the lemma.

edmac’ creator have defined six boolean in order to know which component of line number description we have to print:

- `\ifl@d@pnum` for page numbers;
- `\ifl@d@ssub` for starting sub-line;
- `\ifl@d@elin` for ending line;
- `\ifl@d@esl` for ending sub-line; and
- `\ifl@d@dash` for the dash between the starting and ending groups.

There is no boolean for the line number because it is always printed.

Maieul Rouquette has added `\ifl@d@Xtwolines` and `\ifl@d@Xmorethantwolines` to print a symbol which stands for “and subsequent” when there are two, three or more lines. 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@pnum3807 \newif\ifl@d@pnum
\ifl@d@ssub3808 \newif\ifl@d@ssub
\ifl@d@elin3809 \newif\ifl@d@elin
\ifl@d@esl3810 \newif\ifl@d@esl
\ifl@d@dash3811 \newif\ifl@d@dash
\ifl@d@Xtwolines3812 \newif\ifl@d@Xtwolines%
\ifl@d@Xmorethantwolines3813 \newif\ifl@d@Xmorethantwolines%
\@annot@start@print3814 \let\@annot@start@print\relax%
\@annot@end@print3815 \let\@annot@end@print\relax%
3816 %

\l@d@parsefootspec \l@d@parsefootspec{<spec>}{<lemma>}{<text>} parses a footnote specification. <lemma>
\l@d@p@rsefootspec and <text> are the lemma and text respectively. <spec> is the line and page num-
\l@d@p@rsedstartpage ber and lemma font specifier in \l@d@num style format. The real work is done by
\l@d@p@rsedstartline \l@d@p@rsefootspec which defines macros holding the numeric values. In many cases,
\l@d@p@rsedstartsub this last command is called directly. Just a reminder of the arguments:
\l@d@p@rsedendpage \printlines #1 | #2 | #3 | #4 | #5 | #6 | #7
\l@d@p@rsedendline \printlines start-page | line | subline | end-page | line | subline | fontflag
\l@d@p@rsedendsub3817 \newcommand*{\l@d@parsefootspec}[3]{\l@d@p@rsefootspec#1|}
3818 \def\l@d@p@rsefootspec#1|#2|#3|#4|#5|#6|#7|{%
3819 \gdef\l@d@p@rsedstartpage{#1}%
3820 \gdef\l@d@p@rsedstartline{#2}%
3821 \gdef\l@d@p@rsedstartsub{#3}%
3822 \gdef\l@d@p@rsedendpage{#4}%
3823 \gdef\l@d@p@rsedendline{#5}%
3824 \gdef\l@d@p@rsedendsub{#6}%
3825 }
3826 %

Initialise the several number value macros.

3827 \def\l@d@p@rsedstartpage{0}%
3828 \def\l@d@p@rsedstartline{0}%
3829 \def\l@d@p@rsedstartsub{0}%
3830 \def\l@d@p@rsedendpage{0}%
3831 \def\l@d@p@rsedendline{0}%
3832 \def\l@d@p@rsedendsub{0}%
3833
3834 %

```

**\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.

```

3835 \newcommand*{\setprintlines}[6]{%
3836 \let\@annot@start@print\relax%
3837 \let\@annot@end@print\relax%
3838 \l@d@pnumfalse%
3839 \l@d@dashfalse%
3840 \l@d@elinfalse%
3841 %

```

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

```

3842 \ifbypage@
3843   \ifnum#4=#1 \else
3844     \l@d@pnumtrue
3845     \l@d@dashtrue
3846   \fi
3847 \fi
3848 %

```

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.

```

3849 \ifboolexpr{%
3850   bool{\l@d@pnum}%
3851   or not test{\ifnumequal{#2}{#5}}}%
3852 }{%
3853   \l@d@elintrue%
3854   \l@d@dashtrue%
3855   \unless\ifx\relax\annot@end%
3856     \def\@annot@end@print{%
3857       \l@wrapcs@ifnotemptybox{Xwraplinenumannotation@\@currentseries}{\
annot@end}%
3858     }%
3859   \fi%
3860   }{%
3861 %

```

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.

```

3862 \ifl@d@elin%
3863   \def\@annot@start@print{%
3864     \l@wrapcs@ifnotemptybox{Xwraplinenumannotation@\@currentseries}{\
annot@start}%
3865   }%
3866   \else%
3867     \ifx\annot@start\annot@end%
3868       \unless\ifx\@annot@start\relax%
3869         \def\@annot@start@print{%
3870           \l@wrapcs@ifnotemptybox{Xwraplinenumannotation@\@currentseries}{\
annot@start}%
3871         }%
3872       \fi%
3873     \else%
3874       \ifx\@annot@end@print\relax%
3875         \def\@annot@start@print{%
3876           \l@wrapcs@ifnotemptybox{Xwraplinenumannotation@\@currentseries}{%

```

```

3877         \ifx\annot@start\empty\else%
3878         \annot@start%
3879         \ifdefined\linangesep@%
3880         \linangesep@%
3881         \else%
3882         \csuse{Xlinangeseparator@}\@currentseries}%
3883         \fi%
3884         \fi%
3885         \annot@end%
3886     }%
3887 }%
3888 \else%
3889 \let\@annot@start@print\@annot@end@print%
3890 \let\@annot@end@print\relax%
3891 \fi%
3892 \fi%
3893 \fi%
3894 %

```

We print the starting sub-line if it is nonzero.

```

3895 \l@d@ssubfalse
3896 \ifnum#3=0 \else
3897     \l@d@ssubtrue
3898 \fi
3899 %

```

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.

```

3900 \l@d@eslfalse
3901 \ifnum#6=0 \else
3902     \ifnum#6=#3
3903         \ifl@d@elin \l@d@esltrue \else \l@d@eslfalse \fi
3904     \else
3905         \l@d@esltrue
3906         \l@d@dashtrue
3907     \fi
3908 \fi%
3909 %

```

However, if the `\Xtwolines` is set for the current series, we do not print the last line number.

```

3910 \ifl@d@dash%
3911     \ifbool{expr{togl{fulllines@} or test{\ifcsempy{Xtwolines@}\@currentseries}}}%
3912     {}%
3913     {%
3914         \setistwofollowinglines{#1}{#2}{#4}{#5}%
3915         \ifbool{expr{%
3916             (%

```

```

3917         togl {Xtwolinesbutnotmore@\@currentseries}%
3918         and not%
3919         (%
3920             bool {istwofollowinglines@}%
3921         )%
3922     )%
3923     or%
3924     (%
3925         (not test{\ifnumequal{#1}{#4}})%
3926         and togl{Xtwolinesonlyinsamepage@\@currentseries}%
3927     )%
3928     }%
3929     {}%
3930     {%
3931         \l@d@dashfalse%
3932         \l@d@Xtwolinestrue%
3933         \l@d@elinfalse%
3934         \l@d@eslfalse%
3935         \ifcempty{Xmorethantwolines@\@currentseries}%
3936             {}%
3937             {\ifistwofollowinglines@\else%
3938                 \l@d@Xmorethantwolinestrue%
3939             }%
3940         }%
3941     }%
3942     }%
3943 \fi%
3944 %

```

End of \setprintlines.

```

3945 }%
3946 %

```

**\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.

```

3947 \newif\ifistwofollowinglines@%
3948 \newcommand{\setistwofollowinglines}[4]{%
3949   \ifcsdef{lastlinenumberon@#1}%
3950     {\numdef{\tmp}{\csuse{lastlinenumberon@#1}}}%
3951     {\numdef{\tmp}{0}}%
3952   \istwofollowinglines@false%
3953   \ifnumequal{#4-#2}{1}%
3954     {\istwofollowinglines@true}%
3955     {\ifbypage@%
3956       \ifnumequal{#3-#1}{1}%
3957       {%
3958         \ifnumequal{#2}{\tmp}%
3959         {\ifnumequal{#4}{1}{\istwofollowinglines@true}{}}%
3960         {}%
3961       }%
3962       {}%
3963     \fi%
3964   }%
3965 }%
3966 %

```

`\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

```

3967 \def\printlines#1|#2|#3|#4|#5|#6|#7|#8|{%
3968   \begingroup%
3969   %

```

If we use Lua<sub>T</sub><sub>E</sub>X, ensure we use good text's direction.

```

3970 \ifluatex%
3971   \edef\@tmp{\the\textdir}%
3972   \ifdefstring{\@tmp}{TLT}{\textdir TLT}%Test in order to prevent
    spurious space (bug #397)
3973   \fi%
3974 %

```

Decide which part of line number components we will print.

```

3975   \setprintlines{#1}{#2}{#3}{#4}{#5}{#6}%
3976 %

```

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.

```

3977 \ifdimequal{\csuse{Xboxstartlinenum@\@currentseries}}{0pt}%
3978 {\bgroup}%
3979 {\leavevmode\hbox to \csuse{Xboxstartlinenum@\@currentseries}\bgroup\
hfill}%
3980 \ifcsstring{Xlinenumannotationposition@\@currentseries}{before}%
3981 {\@annot@start@print}%
3982 {}%
3983 \ifl@d@pnum%
3984 \wrap@edcrossref{\@this@crossref@start}{#1}%
3985 \csuse{Xpagelinesep@\@currentseries}%
3986 \fi%
3987 \wrap@edcrossref{\@this@crossref@start}{%
3988 \linenumrep{#2}%
3989 \iftoggle{Xlineflag@\@currentseries}{#8}{}%
3990 }%
3991 \ifl@d@ssub%
3992 \csuse{Xsublinesep@\@currentseries}%
3993 \wrap@edcrossref{\@this@crossref@start}{\sublinenumrep{#3}}%
3994 \fi
3995 \ifcsstring{Xlinenumannotationposition@\@currentseries}{after}%
3996 {\@annot@end@print}%
3997 {}%
3998 \egroup%
3999 %

```

Then print the dash + end line number, or the range symbol.

```

4000 \ifdimequal{\csuse{Xboxendlinenum@\@currentseries}}{0pt}%
4001 {\bgroup}%
4002 {\hbox to \csuse{Xboxendlinenum@\@currentseries}\bgroup}%
4003 \ifl@d@Xtwolines%
4004 \ifl@d@Xmorethantwolines%
4005 \csuse{Xmorethantwolines@\@currentseries}%
4006 \else%
4007 \csuse{Xtwolines@\@currentseries}%
4008 \fi%
4009 \else%
4010 \ifl@d@dash%
4011 \ifdefined\linerrangesep%
4012 \linerrangesep%
4013 \else%
4014 \csuse{Xlinerrangeseparator@\@currentseries}%
4015 \fi%
4016 \fi%
4017 \ifcsstring{Xlinenumannotationposition@\@currentseries}{before}%
4018 {\@annot@end@print}%
4019 {}%
4020 \ifl@d@pnum%
4021 \wrap@edcrossref{\@this@crossref@end}{#4}%
4022 \csuse{Xpagelinesep@\@currentseries}%
4023 \fi%

```

```

4024 \ifl@d@elin%
4025 \wrap@edcrossref{\@this@crossref@end}{%
4026 \linenumrep{#5}%
4027 \iftoggle{Xlineflag@\@currentseries}{#8}{}%
4028 }%
4029 \fi%
4030 \ifl@d@esl%
4031 \ifl@d@elin%
4032 \csuse{Xsublinesep@\@currentseries}%
4033 \fi%
4034 \wrap@edcrossref{\@this@crossref@end}{\sublinenumrep{#6}}%
4035 \fi%
4036 \ifcsstring{Xlinenumannotationposition@\@currentseries}{after}%
4037 {\@annot@end@print}%
4038 }%
4039 \fi%
4040 \ifdimequal{\csuse{Xboxendlinenum@\@currentseries}}{0pt}%
4041 {}%
4042 {\hfill}%Prevent underfull hbox
4043 \egroup%
4044 \endgroup%
4045 }%
4046 %

```

### XIII.7.5 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.

```

4047 \newcommand{\prepare@Xgroupbyline}[3]{%
4048 \iftoggle{Xgroupbylineseparatetwolines@#1}{%
4049 \l@dparsfootspec#2%
4050 \ifcsdef{#1@forinserting@\l@dparseendpage-\l@dparseendline-\l@dparseendsub}%
4051 {%
4052 \csgappto%
4053 {#1@forinserting@\l@dparseendpage-\l@dparseendline-\l@dparseendsub}%
4054 {%
4055 \ifcsemt{Xsymlinenum@#1}%
4056 {\csuse{Xparafootsep@#1}}%
4057 }%
4058 #3{#1}{#2}%
4059 \hskip\csuse{Xafternote@#1}\relax%
4060 }%
4061 }%
4062 {%
4063 \csdef%
4064 {#1@forinserting@\l@dparseendpage-\l@dparseendline-\l@dparseendsub}%

```

```

4065     {%
4066         #3{#1}{#2}%
4067         \hskip\csuse{Xafternote@#1}\relax%
4068     }%
4069 }%
4070 \listcsxadd{#1@forinserting}{\l@dparsedendpage-\l@dparsedendline-\l@dparsedendsub}%
4071 }{%
4072     \ifcsdef{#1@forinserting@all}{%
4073         \csgappto%
4074         {#1@forinserting@all}%
4075     {%
4076         \ifcsemt{Xsymlinenum@#1}%
4077             {\csuse{Xparafootsep@#1}}%
4078             {}%
4079         #3{#1}{#2}%
4080         \hskip\csuse{Xafternote@#1}\relax%
4081     }%
4082 }%
4083 {%
4084     \csdef%
4085     {#1@forinserting@all}%
4086     {%
4087         #3{#1}{#2}%
4088         \hskip\csuse{Xafternote@#1}\relax%
4089     }%
4090 }%
4091 \listcsgadd{#1@forinserting}{all}%
4092 }%
4093 }%
4094 %

```

## 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`

`\multfootsep`



and not `\newcommand`.

```
4095 \providecommand*\multiplefootnotemarker}{3sp}
4096 \providecommand*\multfootsep}{\textsuperscript{\normalfont,}}
4097
4098 %
```

`\m@mmf@prepare` A pair of self-cancelling kerns. This may have been defined in the memoir class.

```
4099 \providecommand*\m@mmf@prepare}{%
4100   \kern-\multiplefootnotemarker
4101   \kern\multiplefootnotemarker\relax}
4102 %
```

`\m@mmf@check` This may have been defined in the memoir class. If it recognises the last kern as `\multiplefootnotemarker` it typesets `\multfootsep`.

```
4103 \providecommand*\m@mmf@check}{%
4104   \ifdim\lastkern=\multiplefootnotemarker\relax
4105     \edef\@x@sf{\the\spacefactor}%
4106     \unkern
4107     \multfootsep
4108     \spacefactor\@x@sf\relax
4109   \fi}
4110
4111 %
```

We have to modify `\@footnotetext` and `\@footnotemark`. However, if memoir is used the modifications have already been made.

```
4112 \ifclassloaded{memoir}{}%
4113 %
```

`\@footnotetext` Add `\m@mmf@prepare` at the end of `\@footnotetext`.

```
4114 \apptocmd{\@footnotetext}{\m@mmf@prepare}{}{}
4115 %
```

`\@footnotemark` Modify `\@footnotemark` to cater for adjacent footnotes.

```
4116
4117 \patchcmd{\@footnotemark}
4118   {\nobreak}
4119   {\m@mmf@check
4120    \nobreak
4121   }
4122   {}{}
4123 \patchcmd{\@footnotemark}
4124   {\@makefnmark}
```

```

4125 {\@makefnmark
4126 \m@mmf@prepare
4127 }
4128 {}{}
4129 %

```

Finished the modifications for the non-memoir case.

```

4130 }
4131
4132 %

```

## 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.

```

4133 \pretocmd{\@footnotetext}{%
4134 \ifnumberedpar@
4135 \edtext{}{\l@dbfnote{#1}}}%
4136 \else
4137 }{}{}
4138 \apptocmd{\@footnotetext}{\fi}{}{}%
4139 %

```

`\l@dbfnote` `\l@dbfnote` adds the footnote to the insert list, and `\vl@dbfnote` calls the original `\@footnotetext`. We also patch `\footnote` in order to get the correct footnote numbers when typesetting parallel texts. This is moved into a `\get@fnmark` command.

```

\footnote
\get@fnmark
\get@thisfootnote
4140
4141 \patchcmd%
4142 {\footnote}%
4143 {\stepcounter\@mpfn}%
4144 {%
4145 \ifboolexpr{bool{l@dpairing} or bool{l@dprintingpages} or bool{
l@dprintingcolumns}}{%
4146 \global\advance\footnote@reading by \@ne%
4147 \get@thisfootnote%
4148 \get@fnmark{\thisc@footnote}%
4149 \ifcsdef{footnotereading\the\footnote@reading=typeset}%
4150 {\setcounter{\@mpfn}{\csuse{footnotereading\the\footnote@reading=
typeset}}}%
4151 {\setcounter{\@mpfn}{\footnote@reading}}%
4152 }{}%
4153 \stepcounter\@mpfn%
4154 }%
4155 }%

```

```

4156 {}
4157 {}
4158
4159 \newcommand{\get@thisfootnote}{%
4160   \ifbool{bool{1@dpairing}} or bool{1@dprintingpages} or bool{
4161     1@dprintingcolumns}}{%
4162     \protected@xdef\thisc@footnote{\the\footnote@reading}%
4163   }{%
4164     \protected@xdef\thisc@footnote{\the\c@footnote}%
4165   }%
4166
4167 \newcommand{\l@dbfnote}[1]{%
4168   \get@thisfootnote%
4169   \gdef\@tag{#1\relax}%
4170   \ifledRcol%
4171     \xright@appenditem{%
4172       \ifdefined\Hy@footnote@currentHref%
4173         \noexpand\def\noexpand\Hy@footnote@currentHref{\
4174         Hy@footnote@currentHref}%
4175       \fi%
4176       \noexpand\l@dbfnote{\expandonce\@tag}\thisc@footnote}%
4177     }%
4178     \to\inserts@listR
4179     \global\advance\insert@countR \@ne%
4180   \else%
4181     \xright@appenditem{%
4182       \ifdefined\Hy@footnote@currentHref%
4183         \noexpand\def\noexpand\Hy@footnote@currentHref{\
4184         Hy@footnote@currentHref}%
4185       \fi%
4186       \noexpand\l@dbfnote{\expandonce\@tag}\thisc@footnote}%
4187     }%
4188     \to\inserts@list
4189     \global\advance\insert@count \@ne%
4190   \fi
4191   \ignorespaces%
4192 }%
4193
4194 \newcommand{\get@fnmark}[1]{%
4195   \ifbool{bool{1@dpairing}} or bool{1@dprintingpages} or bool{
4196     1@dprintingcolumns}}{%
4197     \stepcounter{footnote@typeset}%
4198     \setcounter{footnote}{\c@footnote@typeset}%
4199     \immediate\write\@mainaux{%
4200       \csgdef{footnotereading#1=typeset}{\the\c@footnote@typeset}%
4201     }%
4202   \def\@thefnmark{\thefootnote}%
4203 }%

```

```

4202     {%
4203         \setcounter{footnote}{#1}%
4204         \def\@thefnmark{\thefootnote}%
4205     }%
4206 }%
4207
4208 \newcommand{\vl@dbfnote}[2]{%
4209     \get@fnmark{#2}%
4210     \@footnotetext{#1}%
4211 }%
4212 %

```

### 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`

```

4213 \newcommand*{\prebodyfootmark}{%
4214     \leavevmode
4215     \ifhmode
4216         \edef\@x@sf{\the\spacefactor}%
4217         \m@mmf@check
4218         \nobreak
4219         \fi}
4220 \newcommand{\postbodyfootmark}{%
4221     \m@mmf@prepare
4222     \ifhmode\spacefactor\@x@sf\fi\relax}
4223
4224 %

```

### 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.

```

4225 \newcommandx{\arrangementX}[2][1,usedefault]{%
4226     \def\do##1{%
4227         \csname arrangementX@#2\endcsname{##1}%
4228     }%
4229     \ifstrempy{#1}%
4230     {%
4231         \dolistloop{\@series}%
4232     }%
4233     {
4234         \docsvlist{#1}%

```

```

4235 }%
4236 }%
4237 %

```

#### 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.

```

4238 \newcommand*{\normal@footnotemarkX}[1]{%
4239   \prebodyfootmark
4240   \wrapped@bodyfootmarkX{#1}%
4241   \postbodyfootmark}
4242
4243 %

```

`\normalbodyfootmarkX` The `\normalbodyfootmarkX{<series>}` *really* typesets the in-text marker. The style is the normal superscript.

```

4244 \newcommand*{\normalbodyfootmarkX}[1]{%
4245   \hbox{\textsuperscript{\normalfont\@nameuse{@thefnmark#1}}}%
4246 %

```

`\normalvfootnoteX` `\normalvfootnoteX{<series>}{<text>}` does the `\insert` for the `<series>` and calls the series' `\footfmt...` to format the `<text>`.

```

4247 \notbool{parapparatus@}{\newcommand*{\newcommand}{\normalvfootnoteX}[2]{%
4248   \beforeinsertionX{#1}%
4249   \insert\@nameuse{footins#1}\bgroup
4250     \reset@font%
4251     \hsize=\expandafter\dimexpr\csuse{widthX@#1}\relax%
4252     \noindent\csuse{bhooknoteX@#1}%
4253     \csuse{notefontsizeX@#1}%
4254     \footsplitskips
4255     \ifl@dpairing\ifl@dpaging\else%
4256       \setnoteswidthliketwocolumnsX@{#1}%
4257     \fi\fi%
4258     \setnotesXpositionliketwocolumns@{#1}%
4259     \spaceskip=\z@skip \xspaceskip=\z@skip
4260     \csuse{\csuse{footnote@dir}}\@nameuse{footfmt#1}{#1}{#2}\egroup}
4261
4262 %

```

`\mpnormalvfootnoteX` The minipage version.

```

4263 \newcommand*{\mpnormalvfootnoteX}[2]{%
4264   \get@thisfootnoteX{#1}%
4265   \get@fnmarkX{#1}{\thisc@footnote}%
4266   \edef\thisc@footnoteX@reading{\the\csname footnote#1@reading\endcsname}%
4267   \global\setbox\@nameuse{mpfootins#1}\vbox{%

```

```

4268 \unvbox\@nameuse{mpfootins#1}
4269 \noindent\csuse{bhooknoteX@#1}%
4270 \csuse{notefontsizeX@#1}%
4271 \hsize\columnwidth
4272 \@parboxrestore
4273 \color@begingroup
4274 \@nameuse{footfmt#1}{#1}{#2}\color@endgroup}}
4275
4276 %

```

`\normalfootfmtX` `\normalfootfmtX{series}{text}` typesets the footnote text, prepended by the marker.

```

4277 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\normalfootfmtX}[2]{%
4278 \ifluatex%
4279 \textdir\footnote@luatextextdir%
4280 \pardir\footnote@luatexpardir%
4281 \par%
4282 \fi%
4283 \protected@edef\@currentlabel{%
4284 \@nameuse{@thefnmark#1}%
4285 }%
4286 \ledsetnormalparstuffX{#1}%
4287 \hangindent=\csuse{hangindentX@#1}%
4288 \everypar{\hangindent=\csuse{hangindentX@#1}}%
4289 \rule{z@splittopskip}%
4290 {\csuse{notenumfontX@#1}\wrapped@footfootmarkX{#1}}%
4291 \csuse{wrapcontentX@#1}{#2}%
4292 \strut\par}}
4293
4294 %

```

`\normalfootfootmarkX` `\normalfootfootmarkX{series}` is called by `\normalfootfmtX` to typeset the footnote marker in the footer before the footnote text.

```

4295 \newcommand*{\normalfootfootmarkX}[1]{%
4296 \textsuperscript{\@nameuse{@thefnmark#1}}}
4297
4298 %

```

`\normalfootstartX` `\normalfootstartX{series}` is the *series* footnote starting macro used in the output routine.

```

4299 \newcommand*{\normalfootstartX}[1]{%
4300 \ifdimequal{Opt}{\prenotesX@}{}%
4301 {%
4302 \iftoggle{prenotesX@}{%
4303 \togglefalse{prenotesX@}%
4304 \skip\cname footins#1\endcname=%
4305 \glueexpr\csuse{prenotesX@}+\csuse{afterruleX@#1}\relax%
4306 }%

```

```

4307     {}%
4308   }%
4309   \vskip\skip\csname footins#1\endcsname%
4310   \leftskip=\z@
4311   \rightskip=\z@
4312   \ifl@dpairing\else%
4313     \hsize=\old@hsize%
4314   \fi%
4315   \setnoteswidthliketwocolumnsX@{#1}%
4316   \setnotesXpositionliketwocolumns@{#1}%
4317   \print@footnoteXrule{#1}%
4318 }%
4319
4320 %

```

**\normalfootnoteruleX** The rule drawn before the footnote series group.

```

4321 \let\normalfootnoteruleX=\footnoterule
4322
4323 %

```

**\normalfootgroupX** `\normalfootgroupX{<series>}` sends the contents of the `<series>` insert box to the output page without alteration.

```

4324 \newcommand*{\normalfootgroupX}[1]{%
4325   \csuse{bhookgroupX@#1}%
4326   \unvbox\@nameuse{footins#1}%
4327   \hsize=\old@hsize%
4328 }%
4329
4330 %

```

**\mpnormalfootgroupX** The minipage version.

```

4331 \newcommand*{\mpnormalfootgroupX}[1]{%
4332   \vskip\skip\@nameuse{mpfootins#1}
4333   \ifl@dpairing\ifparledgroup%
4334     \leavevmode\marks\parledgroup@{begin}%
4335     \marks\parledgroup@series{#1}%
4336     \marks\parledgroup@type{footnoteX}%
4337   \fi\fi\normalcolor
4338   \ifparledgroup%
4339     \ifl@dpairing%
4340     \else%
4341       \setnoteswidthliketwocolumnsX@{#1}%
4342       \setnotesXpositionliketwocolumns@{#1}%
4343       \print@footnoteXrule{#1}%
4344     \fi%
4345   \else%
4346     \setnoteswidthliketwocolumnsX@{#1}%

```

```

4347 \setnotesXpositionliketwocolumns@{#1}%
4348 \print@footnoteXrule{#1}%
4349 \fi%
4350 \csuse{bhookgroupX@#1}%
4351 \unvbox\@nameuse{mpfootins#1}}
4352
4353 %

```

**\normalbfnoteX**<sub>54</sub>

```

4355 \newcommand{\normalbfnoteX}[2]{%
4356 \get@thisfootnoteX{#1}%
4357 \ifledRcol%
4358 \ifluatex
4359 \footnotelang@lua[R]%
4360 \fi
4361 \@ifundefined{xpg@main@language}%if polyglossia
4362 {}%
4363 {\footnotelang@poly[R]}%
4364 \xright@appenditem{%
4365 \noexpand\led@set@index@fornote{#1}%
4366 \noexpand\prepare@edindex@fornote{\led@nums}%
4367 \unexpanded{\def\this@footnoteX@reading}{\the\csname footnote#1
@reading\endcsname}%
4368 \noexpand\vbfnoteX{#1}{#2}{\thisc@footnote}%
4369 \noexpand\led@reinit@index@fornote%
4370 \unexpanded{\advance\@edindex@fornote@m@ne}%
4371 }%
4372 \to\inserts@listR
4373 \global\advance\insert@countR \@ne%
4374 \else%
4375 \ifluatex
4376 \footnotelang@lua%
4377 \fi
4378 \@ifundefined{xpg@main@language}%if polyglossia
4379 {}%
4380 {\footnotelang@poly}%
4381 \xright@appenditem{%
4382 \noexpand\led@set@index@fornote{#1}%
4383 \noexpand\prepare@edindex@fornote{\led@nums}%
4384 \unexpanded{\def\this@footnoteX@reading}{\the\csname footnote#1
@reading\endcsname}%
4385 \noexpand\vbfnoteX{#1}{#2}{\thisc@footnote}%
4386 \noexpand\led@reinit@index@fornote%
4387 \unexpanded{\advance\@edindex@fornote@m@ne}%
4388 }%
4389 \to\inserts@list
4390 \global\advance\insert@count \@ne%
4391 \fi
4392 \ignorespaces}

```



```
4393
4394 %
```

**\get@thisfootnoteX** The macro `\get@thisfootnote` command just saves the footnote number in the `\thisfootnote` macro, depending on the use of pairing environments.

```
4395 \newcommand{\get@thisfootnoteX}[1]{%
4396   \ifbool{bool{1@dpairing}} or bool{1@dprintingpages} or bool{
4397     1@dprintingcolumns}}{%
4398     \protected@xdef\thisc@footnote{\the\csname footnote#1@reading\
4399       endcsname}%
4400   }{%
4401     \protected@xdef\thisc@footnote{\the\csname c@footnote#1\endcsname}%
4402   }%
4403 }
```

**\vbfnoteX** This command calls the correct footnote-inserting commands.

```
4403 \newcommand{\vbfnoteX}[3]{%
4404   \get@fnmarkX{#1}{#3}%
4405   \@nameuse{regvfootnote#1}{#1}{#2}%
4406 }%
4407
4408 %
```

**\get@fnmarkX** This command gets the correct footnote number when typesetting parallel texts.

```
4409 \newcommand{\get@fnmarkX}[2]{%
4410   \ifbool{bool{1@dpairing}} or bool{1@dprintingpages} or bool{
4411     1@dprintingcolumns}}{%
4412     {%
4413       \stepcounter{footnote#1@typeset}%
4414       \setcounter{footnote#1}{\value{footnote#1@typeset}}%
4415       \@namedef{@thefnmark#1}{\csuse{thefootnote#1}}%
4416       \immediate\write\@mainaux{%
4417         \csgdef{footnote#1@reading#2=typeset}{\the\csname c@footnote#1
4418           @typeset\endcsname}%
4419       }%
4420     }%
4421     {%
4422       \setcounter{footnote#1}{#2}%
4423       \@namedef{@thefnmark#1}{\csuse{thefootnote#1}}%
4424     }%
4425 }
```

```

\newcommand{\vnumfootnoteX}[2]{%
4427 \ifnumberedpar@
4428 \edtext{}{\normalbfnoteX{#1}{#2}}%
4429 \else
4430 \def\this@footnoteX@reading{\the\csname footnote#1@reading\endcsname}%
4431 \get@thisfootnoteX{#1}%
4432 \get@fnmarkX{#1}{\expandonce\thisc@footnote}%
4433 \@nameuse{regvfootnote#1}{#1}{#2}%
4434 \fi}
4435
4436 %

```

**arrangementX@normal** `\arrangementX@normal{<series>}` initialises the settings for the <series> footnotes. This should always be called for each series.

```

4437 \newcommand*{\arrangementX@normal}[1]{%
4438 \csgdef{series@displayX#1}{normal}
4439 \expandafter\let\csname footstart#1\endcsname=\normalfootstartX
4440 \@namedef{@footnotemark#1}{\normal@footnotemarkX{#1}}
4441 \@namedef{bodyfootmark#1}{\normalbodyfootmarkX{#1}}
4442 \expandafter\let\csname regvfootnote#1\endcsname=\normalvfootnoteX
4443 \expandafter\let\csname vfootnote#1\endcsname=\vnumfootnoteX
4444 \expandafter\let\csname footfmt#1\endcsname=\normalfootfmtX
4445 \@namedef{footfootmark#1}{\normalfootfootmarkX{#1}}
4446 \expandafter\let\csname footgroup#1\endcsname=\normalfootgroupX
4447 \expandafter\let\csname footnoterule#1\endcsname=\normalfootnoteruleX
4448 \count\csname footins#1\endcsname=1000
4449 \csxdef{default@footins#1}{1000}%Use to have note only for one side
4450 \dimen\csname footins#1\endcsname=\csuse{maxhnotesX@#1}
4451 \skip\csname footins#1\endcsname=\csuse{beforenotesX@#1}%
4452 \advance\skip\csname footins#1\endcsname by\csuse{afterruleX@#1}%
4453 %

```

Additions for minipages.

```

4454 \ifnoledgroup@else%
4455 \expandafter\let\csname mpvfootnote#1\endcsname=\mpnormalvfootnoteX
4456 \expandafter\let\csname mpfootgroup#1\endcsname=\mpnormalfootgroupX
4457 \count\csname mpfootins#1\endcsname=1000
4458 \dimen\csname mpfootins#1\endcsname=\csuse{maxhnotesX@#1}
4459 \skip\csname mpfootins#1\endcsname=\csuse{beforenotesX@#1}%
4460 \advance\skip\csname mpfootins#1\endcsname by\csuse{afterruleX@#1}%
4461 \fi
4462 }
4463
4464 %

```

#### 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  \newcommand*{\arrangementX@twocol}[1]{%
4466 \csgdef{series@displayX#1}{twocol}
4467 \expandafter\let\csname regvfootnote#1\endcsname=\twocolvfootnoteX
4468 \expandafter\let\csname footfmt#1\endcsname=\twocolfootfmtX
4469 \expandafter\let\csname footgroup#1\endcsname=\twocolfootgroupX
4470 \dimen\csname footins#1\endcsname=\csuse{maxhnotesX@#1}%
4471 \skip\csname footins#1\endcsname=\csuse{beforenotesX@#1}%
4472 \advance\skip\csname footins#1\endcsname by \csuse{afterruleX@#1}\relax%
4473 \twocolfootsetupX{#1}
4474 \ifnoledgroup@ \else%
4475 \expandafter\let\csname mpvfootnote#1\endcsname=\mpnormalvfootnoteX
4476 \expandafter\let\csname mpfootgroup#1\endcsname=\mptwocolfootgroupX
4477 \skip\csname mpfootins#1\endcsname=\csuse{beforenotesX@#1}%
4478 \advance\skip\csname mpfootins#1\endcsname by \csuse{afterruleX@#1}
4479 \mptwocolfootsetupX{#1}
4480 \fi%
4481 }
4482
4483 %

```

```

\twocolfootsetupX \twocolfootsetupX{<series>}
\mptwocolfootsetupX
4484 \newcommand*{\twocolfootsetupX}[1]{%
4485 \count\csname footins#1\endcsname 500
4486 \csxdef{default@footins#1}{500}%Use this to confine the notes to one
side only
4487 \multiply\dimen\csname footins#1\endcsname by \tw@}
4488 \newcommand*{\mptwocolfootsetupX}[1]{%
4489 \count\csname mpfootins#1\endcsname 500
4490 \multiply\dimen\csname mpfootins#1\endcsname by \tw@}
4491
4492 %

```

```

\twocolvfootnoteX \twocolvfootnoteX{<series>}
4493 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\twocolvfootnoteX}[2]{%
4494 \beforeinsertion@X{#1}%
4495 \insert\csname footins#1\endcsname\bgroup%
4496 \hsize=\expandafter\dimexpr\csuse{widthX@#1}\relax%
4497 \noindent\csuse{bhooknoteX@#1}%
4498 \csuse{notefontsizeX@#1}%
4499 \footsplitskips%
4500 \spaceskip=\z@skip \xspaceskip=\z@skip%
4501 \@nameuse{footfmt#1}{#1}{#2}\egroup}
4502
4503 %

```

```

\twocolfootfmtX \twocolfootfmtX{<series>}

```

```

4504 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\twocolfootfmtX}[2]{%
4505   \protected@edef\@currentlabel{%
4506     \@nameuse{@thefnmark#1}%
4507   }%
4508   \normal@pars%
4509   \hangindent=\csuse{hangindentX@#1}%
4510   \everypar{\hangindent=\csuse{hangindentX@#1}}%
4511   \hsize \csuse{hsizeX@#1}%
4512   \nottoggle{parindentX@#1}{\parindent=\z@}{}%
4513   \tolerance=5000\relax%
4514   \par%
4515   \@tempdima=\parindent%
4516   \csuse{colalignX@#1}%
4517   \parindent=\@tempdima%
4518   {\hspace{\parindent}}%
4519   \csuse{notenunfontX@#1}\wrapped@footfootmarkX{#1}\strut%
4520   \csuse{wrapcontentX@#1}{#2}%
4521   \strut\par}%
4522 \allowbreak%
4523 }%
4524
4525 %

```

```

\twocolfootgroupX \twocolfootgroupX{<series>}
\mptwocolfootgroupX
4526 \newcommand*{\twocolfootgroupX}[1]{\csuse{bhookgroupX@#1}\csuse{
notefontsizeX@#1}
4527   \splittopskip=\ht\strutbox
4528   \expandafter
4529   \rigidbalanceX\csname footins#1\endcsname \tw@ \splittopskip}}
4530
4531 \newcommand*{\mptwocolfootgroupX}[1]{%
4532   \vskip\skip\@nameuse{mpfootins#1}
4533   \ifl@dpairing\ifparledgroup%
4534     \leavevmode\marks\parledgroup@{begin}%
4535     \marks\parledgroup@series{#1}%
4536     \marks\parledgroup@type{footnoteX}%
4537   \fi\fi\normalcolor
4538   \ifparledgroup%
4539     \ifl@dpairing%
4540     \else%
4541       \setnoteswidthliketwocolumnsX@{#1}%
4542       \setnotesXpositionliketwocolumns@{#1}%
4543       \print@footnoteXrule{#1}%
4544     \fi%
4545   \else%
4546     \setnoteswidthliketwocolumnsX@{#1}%
4547     \setnotesXpositionliketwocolumns@{#1}%
4548     \print@footnoteXrule{#1}%
4549   \fi%

```

```

4550 \csuse{bhookgroupX@#1}%
4551 \splittopskip=\ht\strutbox
4552 \expandafter
4553 \rigidbalanceX\csname mpfootins#1\endcsname \tw@ \splittopskip}}
4554
4555 %

```

#### 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 4556 \newcommand*{\arrangementX@threecol}[1]{%
4557 \csgdef{series@displayX#1}{threecol}
4558 \expandafter\let\csname regvfootnote#1\endcsname=\threecolvfootnoteX
4559 \expandafter\let\csname footfmt#1\endcsname=\threecolfootfmtX
4560 \expandafter\let\csname footgroup#1\endcsname=\threecolfootgroupX
4561 \dimen\csname footins#1\endcsname=\csuse{maxhnotesX@#1}%
4562 \skip\csname footins#1\endcsname=\csuse{beforenotesX@#1}%
4563 \advance\skip\csname footins#1\endcsname by \csuse{afterruleX@#1}\relax%
4564 \threecolfootsetupX{#1}
4565 \ifnoledgroup@else%
4566 \expandafter\let\csname mpvfootnote#1\endcsname=\mpnormalvfootnoteX
4567 \expandafter\let\csname mpfootgroup#1\endcsname=\mpthreecolfootgroupX
4568 \skip\csname mpfootins#1\endcsname=\csuse{beforenotesX@#1}%
4569 \advance\skip\csname mpfootins#1\endcsname by \csuse{afterruleX@#1}
4570 \mpthreecolfootsetupX{#1}
4571 \fi%
4572 }
4573
4574 %

```

```

\threecolfootsetupX \threecolfootsetupX{<series>}
\mpthreecolfootsetupX
4575 \newcommand*{\threecolfootsetupX}[1]{%
4576 \count\csname footins#1\endcsname 333
4577 \csxdef{default@footins#1}{333}%Use this to confine the notes to one
side only
4578 \multiply\dimen\csname footins#1\endcsname by \thr@@}
4579 \newcommand*{\mpthreecolfootsetupX}[1]{%
4580 \count\csname mpfootins#1\endcsname 333
4581 \multiply\dimen\csname mpfootins#1\endcsname by \thr@@}
4582
4583 %

```

```

\threecolvfootnoteX \threecolvfootnoteX{<series>}{<text>}
4584 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\threecolvfootnoteX}[2]{
%

```

```

4585 \beforeinsertionX{#1}%
4586 \insert\csname footins#1\endcsname\bgroup%
4587   \hsize=\expandafter\dimexpr\csuse{widthX@#1}\relax%
4588   \noindent\csuse{bhooknoteX@#1}%
4589   \csuse{notefontsizeX@#1}%
4590   \footsplitskips%
4591   \@nameuse{footfmt#1}{#1}{#2}\egroup
4592
4593 %

```

`\threecolfootfmtX` `\threecolfootfmtX{<series>}`

```

4594 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\threecolfootfmtX}[2]{%
4595   \protected@edef\@currentlabel{%
4596     \@nameuse{@thefnmark#1}%
4597   }%
4598   \hangindent=\csuse{hangindentX@#1}%
4599   \everypar{\hangindent=\csuse{hangindentX@#1}}%
4600   \normal@pars%
4601   \hsize \csuse{hsizethreecolX@#1}%
4602   \nottoggle{parindentX@#1}{\parindent=\z@}{}%
4603   \tolerance=5000\relax%
4604   \@tempdima=\parindent%
4605   \csuse{colalignX@#1}%
4606   \parindent=\@tempdima%
4607   {\hspace{\parindent}}%
4608   \csuse{notenunfontX@#1}\wrapped@footfootmarkX{#1}\strut%
4609   \csuse{wrapcontentX@#1}{#2}%
4610   \strut\par}\allowbreak
4611
4612 %

```

`\threecolfootgroupX` `\threecolfootgroupX{<series>}`  
`\mpthreecolfootgroupX`

```

4613 \newcommand*{\threecolfootgroupX}[1]{\csuse{bhookgroupX@#1}\csuse{
notefontsizeX@#1}
4614   \splittopskip=\ht\strutbox
4615   \expandafter
4616   \rigidbalanceX\csname footins#1\endcsname \thr@@ \splittopskip}}
4617
4618 \newcommand*{\mpthreecolfootgroupX}[1]{%
4619   \vskip\skip\@nameuse{mpfootins#1}
4620   \ifl@dpairing\ifparledgroup
4621     \leavevmode\marks\parledgroup@{begin}%
4622     \marks\parledgroup@series{#1}%
4623     \marks\parledgroup@type{footnoteX}%
4624     \fi\fi\normalcolor
4625     \ifparledgroup%
4626     \ifl@dpairing%
4627     \else%

```

```

4628 \setnoteswidthliketwocolumnsX@{#1}%
4629 \setnotesXpositionliketwocolumns@{#1}%
4630 \print@footnotexrule{#1}%
4631 \fi%
4632 \else%
4633 \setnoteswidthliketwocolumnsX@{#1}%
4634 \setnotesXpositionliketwocolumns@{#1}%
4635 \print@footnotexrule{#1}%
4636 \fi%
4637 \csuse{bhookgroupX@#1}%
4638 \splittopskip=\ht\strutbox
4639 \expandafter
4640 \rigidbalanceX\csname mpfootins#1\endcsname \thr@@ \splittopskip}}
4641
4642 %

```

#### XIV.4.5 Paragraphed footnotes

The following macros set footnotes as one paragraph.

`\arrangementX@threecol` `\footparagraphX{<series>}`

```

4643 \newcommand*{\arrangementX@paragraph}[1]{%
4644 \csgdef{series@displayX#1}{paragraph}%
4645 \expandafter\let\csname footstart#1\endcsname=\parafootstartX
4646 \expandafter\let\csname regvfootnote#1\endcsname=\para@vfootnoteX
4647 \expandafter\let\csname footfmt#1\endcsname=\parafootfmtX
4648 \expandafter\let\csname footgroup#1\endcsname=\para@footgroupX
4649 \expandafter\let\csname footnoterule#1\endcsname=\normalfootnoteruleX
4650 \count\csname footins#1\endcsname=1000
4651 \csxdef{default@footins#1}{1000}%Use this to confine the notes to one
side only
4652 \dimen\csname footins#1\endcsname=\csuse{maxhnotesX@#1}
4653 \skip\csname footins#1\endcsname=\csuse{beforenotesX@#1}%
4654 \advance\skip\csname footins#1\endcsname by\csuse{afterruleX@#1}%
4655 \para@footsetupX{#1}
4656 \ifnoledgroup@else
4657 \expandafter\let\csname mpvfootnote#1\endcsname=\mppara@vfootnoteX
4658 \expandafter\let\csname mpfootgroup#1\endcsname=\mppara@footgroupX
4659 \count\csname mpfootins#1\endcsname=1000
4660 \dimen\csname mpfootins#1\endcsname=\csuse{maxhnotesX@#1}
4661 \skip\csname mpfootins#1\endcsname=\csuse{beforenotesX@#1}%
4662 \advance\skip\csname mpfootins#1\endcsname by\csuse{afterruleX@#1}%
4663 \fi
4664 }
4665
4666 %

```

`\para@footsetupX` `\para@footsetupX{<series>}`

```

4667 \newcommand*\para@footsetupX}[1]{\csuse{bhookgroupX@#1}\csuse{
notefontsizeX@#1}
4668 \setnoteswidthliketwocolumnsX@{#1}%
4669 \ifcempty{widthX@#1}%
4670 {}%
4671 {\columnwidth=\expandafter\dimexpr\csuse{widthX@#1}\relax}%
4672 \dimen0=\baselineskip
4673 \multiply\dimen0 by 1024
4674 \divide\dimen0 by \columnwidth \multiply\dimen0 by \footfudgefiddle\relax
4675 %
4676 \expandafter
4677 \xdef\csname footfudgefactor#1\endcsname{%
\expandafter\strip@pt\dimen0 }}
4678 %
4679 %

```

`\parafootstartX` `\parafootstartX{<series>}`

```

4680 \newcommand*\parafootstartX}[1]{%
4681 \ifdimequal{0pt}{\prenotesX@}{}%
4682 {%
4683 \iftoggle{prenotesX@}{%
4684 \togglefalse{prenotesX@}%
4685 \skip\csname footins#1\endcsname=%
4686 \glueexpr\csuse{prenotesX@}+\csuse{afterruleX@#1}\relax%
4687 }%
4688 }%
4689 }%
4690 \leftskip=\z@
4691 \rightskip=\z@
4692 \nottoggle{parindentX@#1}{\parindent=\z@}{}%
4693 \vskip\skip\@nameuse{footins#1}%
4694 \setnoteswidthliketwocolumnsX@{#1}%
4695 \setnotesXpositionliketwocolumns@{#1}%
4696 \print@footnoteXrule{#1}%
4697 }
4698 %
4699 %

```

`\para@vfootnoteX` `\para@vfootnoteX{<series>}{<text>}`  
`\mppara@vfootnoteX`

```

4700 \newcommand*\para@vfootnoteX}[2]{%
4701 \csuse{beforeinsertingX@#1}%
4702 \insert\csname footins#1\endcsname%
4703 \bgroup
4704 \csuse{notefontsizeX@#1}
4705 \footsplitskips
4706 \setbox0=\vbox{\hsize=\maxdimen%
4707 \let\bidirectional\everypar\@empty%
4708 \insert@txtbeforenotesX{#1}%

```



```

4709 \noindent\csuse{bhooknoteX@#1}%
4710 \@nameuse{footfmt#1}{#1}{#2}}%
4711 \setbox0=\hbox{\unvvhX{0}{#1}}%
4712 \dp0=\z@
4713 \ht0=\csname footfudgefactor#1\endcsname\wd0
4714 \box0
4715 \penalty0
4716 \egroup}
4717 \newcommand*{\mppara@vfootnoteX}[2]{%
4718 \get@thisfootnoteX{#1}%
4719 \get@fnmarkX{#1}{\thisc@footnote}%
4720 \edef\thisc@footnoteX@reading{\the\csname footnote#1@reading\endcsname}%
4721 \global\setbox\@nameuse{mpfootins#1}\vbox{%
4722 \unvvhbox\@nameuse{mpfootins#1}
4723 \csuse{notefontsizeX@#1}
4724 \footsplitskips
4725 \setbox0=\vbox{\hsize=\maxdimen%
4726 \let\bidir@RTL@everypar\@empty%
4727 \noindent\color@begingroup%
4728 \csuse{bhooknoteX@#1}%
4729 \@nameuse{footfmt#1}{#1}{#2}\color@endgroup}%
4730 \setbox0=\hbox{\unvvhX{0}{#1}}%
4731 \dp0=\z@
4732 \ht0=\csname footfudgefactor#1\endcsname\wd0
4733 \box0
4734 \penalty0}}
4735
4736 %

```

```

\unvvhX 4737 \newcommand*{\unvvhX}[2]{% 2th is optional for retro-compatibility
4738 \setbox0=\vbox{\unvvhbox#1%
4739 \global\setbox1=\lastbox}%
4740 \unhbox1
4741 \unskip % remove \rightskip,
4742 \unskip % remove \parfillskip,
4743 \unpenalty % remove \penalty of 10000,
4744 \hskip\csuse{afternoteX@#2}%
4745 \relax}% but add the glue to go between the notes
4746
4747 %

```

`\parafootfmtX` `\parafootfmtX{<series>}`

```

4748 \newcommand*{\parafootfmtX}[2]{%
4749 \protected@edef\@currentlabel{%
4750 \@nameuse{@thefnmark#1}%
4751 }%
4752 \insertparafootsepX{#1}%
4753 \ledsetnormalparstuff@common%

```

```

4754 {\csuse{notenumfontX@#1}%
4755 \csuse{notenumfontX@#1}%
4756 \wrapped@footfootmarkX{#1}%
4757 \strut%
4758 \csuse{wrapcontentX@#1}{#2}%
4759 \penalty-10}}
4760
4761 %

```

```

\para@footgroupX \para@footgroupX{<series>}
\mppara@footgroupX
4762 \newcommand*{\para@footgroupX}[1]{%
4763 \hspace=\expandafter\dimexpr\csuse{widthX@#1}\relax%
4764 \unvbox\csname footins#1\endcsname
4765 \ifcsstring{raggedX@#1}{L}{\RaggedLeft}{}%
4766 \ifcsstring{raggedX@#1}{R}{\RaggedRight}{}%
4767 \makeboxofhboxes
4768 \setbox0=\hbox{\unhbox0 \removehboxes}%
4769 \csuse{bhookgroupX@#1}
4770 \csuse{notefontsizeX@#1}
4771 \unhbox0\par}
4772
4773 \newcommand*{\mppara@footgroupX}[1]{%
4774 \setnoteswidthliketwocolumnsX@{#1}%
4775 \vskip\skip@nameuse{mpfootins#1}
4776 \ifl@dpairing\ifparledgroup
4777 \leavevmode%
4778 \leavevmode\marks\parledgroup@{begin}%
4779 \marks\parledgroup@series{#1}%
4780 \marks\parledgroup@type{footnoteX}%
4781 \fi\fi\normalcolor
4782 \ifparledgroup%
4783 \ifl@dpairing%
4784 \else%
4785 \setnoteswidthliketwocolumnsX@{#1}%
4786 \setnotesXpositionliketwocolumns@{#1}%
4787 \print@footnoteXrule{#1}%
4788 \fi%
4789 \else%
4790 \setnoteswidthliketwocolumnsX@{#1}%
4791 \setnotesXpositionliketwocolumns@{#1}%
4792 \print@footnoteXrule{#1}%
4793 \fi%
4794 \unvbox\csname mpfootins#1\endcsname
4795 \ifcsstring{raggedX@#1}{L}{\RaggedLeft}{}%
4796 \ifcsstring{raggedX@#1}{R}{\RaggedRight}{}%
4797 \makeboxofhboxes
4798 \setbox0=\hbox{\unhbox0 \removehboxes}%
4799 \csuse{bhookgroupX@#1}%
4800 \csuse{notefontsizeX@#1}%

```

```

4801 \nottoggle{parindentX@#1}{\parindent=\z@}{}%
4802 \unhbox0\par}}
4803
4804 %

```

**Insertion of the footnotes separator** The command `\insertparafootsepX{<series>}` must be called at the beginning of `\parafootftmX`.

```

\insertparafootsepX \newcommand{\insertparafootsepX}[1]{%
4805
4806   \ifledRcol{%
4807     \ifnumequal{\csuse{prevpage#1@numR}}{\page@numR}%
4808       {\csuse{Xparafootsep@#1}}%
4809       {}%
4810     \global\csname prevpage#1@numR\endcsname=\page@numR%
4811   \else%
4812     \ifnumequal{\csuse{prevpage#1@num}}{\page@num}%
4813       {\csuse{Xparafootsep@#1}}%
4814       {}%
4815     \global\csname prevpage#1@num\endcsname=\page@num%
4816   \fi%
4817 }
4818 %

```

## 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.

```

4819 \newcommand{\wrapped@footfootmarkX}[1]{%
4820   \ifdefined\hypertarget%
4821     \hyperlink%
4822       {@bodyfootmark#1@this@footnoteX@reading}%
4823       {@@nameuse{footfootmark#1}}%
4824     \Hy@raisedlink{%
4825       \hypertarget%
4826       {@footnotemark#1@this@footnoteX@reading}%
4827       {}%
4828     }%
4829   \else%
4830     \@nameuse{footfootmark#1}%
4831   \fi%
4832 }%
4833 %

```

`\wrapped@bodyfootmarkX` `\wrapped@bodyfootmarkX` prints the footnote mark of the text body, wrapped in `hyperref` package's commands, if needed.

```

4834 \newcommand{\wrapped@bodyfootmarkX}[1]{%
4835   \ifdefined\hypertarget%
4836     \hyperlink%
4837       {@footnotemark#1@\expandafter\the\csname footnote#1@reading\
endcsname}%
4838     {@nameuse{bodyfootmark#1}}%
4839   \Hy@raisedlink{%
4840     \hypertarget%
4841       {@bodyfootmark#1@\expandafter\the\csname footnote#1@reading\
endcsname}%
4842     {}%
4843   }%
4844   \else%
4845     \@nameuse{bodyfootmark#1}%
4846   \fi%
4847 }%
4848 %

```

## 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<sub>TEX</sub>.

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 49 \newcommand*{\ledsetnormalparstuff@common}{%
\Xledsetnormalparstuff 50   \ifluatex%
\ledsetnormalparstuffX 51   \ifdefstring{\footnote@luatextextdir}{TLT}{}%
4852     {\textdir\footnote@luatextextdir}%
4853   \pardir\footnote@luatexpardir%
4854   \fi%
4855   \csuse{\csuse{footnote@dir}}%
4856   \normal@pars%
4857   \parfillskip \z@ \@plus 1fil}%
4858
4859 \newcommand*{\Xledsetnormalparstuff}[1]{%
4860   \ledsetnormalparstuff@common%
4861   \nottoggle{Xparindent@#1}{\parindent=\z@}{\hspace{\parindent}}%
4862 }%
4863
4864 \newcommand*{\ledsetnormalparstuffX}[1]{%
4865   \ledsetnormalparstuff@common%

```

```

4866 \nottoggle{parindentX@#1}{\parindent=\z@}{\hspace{\parindent}}}%
4867 }%
4868 %

```

## 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.

```

4869
4870 \newdimen\old@hsize%
4871 \AtBeginDocument{\old@hsize=\hsize}%
4872
4873 \newcommand{\setXnoteswidthliketwocolumns@}[1]{%
4874   \global\let\hsize@fornote=\hsize%
4875   \global\old@hsize=\hsize%
4876   \let\old@columnwidth=\columnwidth%
4877   \iftoggle{Xnoteswidthliketwocolumns@#1}%
4878     {%
4879       \setwidthliketwocolumns%
4880       \global\let\hsize@fornote=\hsize%
4881     }%
4882   {%}
4883   \let\hsize=\hsize@fornote%
4884   \let\columnwidth=\old@columnwidth%
4885 }%
4886
4887 \newcommand{\setnoteswidthliketwocolumnsX@}[1]{%
4888   \global\let\hsize@fornote=\hsize%
4889   \global\old@hsize=\hsize%
4890   \let\old@columnwidth=\columnwidth%
4891   \iftoggle{noteswidthliketwocolumnsX@#1}%
4892     {%
4893       \setwidthliketwocolumns%
4894       \global\let\hsize@fornote=\hsize%
4895     }%
4896   {%}
4897   \let\hsize=\hsize@fornote%
4898   \let\columnwidth=\old@columnwidth%
4899 }%
4900
4901 %

```

`\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 sens without `reledpar`.

`\setnotesXpositionliketwocolumns@`

```

4902 \newcommand{\setXnotespositionliketwocolumns@}[1]{%
4903   \iftoggle{Xnoteswidthliketwocolumns@#1}{%
4904     \csuse{setnotespositionliketwocolumns@\columns@position}%
4905   }{}%
4906 }%
4907
4908 \newcommand{\setnotesXpositionliketwocolumns@}[1]{%
4909   \iftoggle{noteswidthliketwocolumnsX@#1}{%
4910     \csuse{setnotespositionliketwocolumns@\columns@position}%
4911   }{}%
4912 }%
4913
4914 %

```

## 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.

`\mpfnpos`

`\@fnpos`

`\@mpfnpos`

```

4915 \def\@fnpos{familiar-critical}
4916 \def\@mpfnpos{critical-familiar}
4917 \newcommand{\fnpos}[1]{\xdef\@fnpos{#1}}
4918 \newcommand{\mpfnpos}[1]{\xdef\@mpfnpos{#1}}
4919 %

```

## 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@Xfootnoterule20 \newcommand{\print@Xfootnoterule}[1]{%
\print@footnoteXrule21   \vskip-\csuse{Xafterterrule@#1}%Because count in \dimen\csuse{#1footins}
4922   \nointerlineskip%
4923   \moveleft-\leftskip\vbox{\csuse{#1footnoterule}}%
4924   \nointerlineskip%
4925   \vskip\csuse{Xafterterrule@#1}%
4926 }%
4927
4928 \newcommand{\print@footnoteXrule}[1]{%
4929   \vskip-\csuse{afterterruleX@#1}%Because count in \dimen\csuse{footins#1}
4930   \nointerlineskip%
4931   \moveleft-\leftskip\vbox{\csuse{footnoterule#1}}%

```

```

4932 \nointerlineskip%
4933 \vskip\csuse{afterruleX@#1}%
4934 }%
4935
4936 %

```

## 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.

```

4937 \newtoggle{Xprenotes@}%
4938 \toggletrue{Xprenotes@}%

```

```

4939 \newcommand{\Xprenotes@}{Opt}%
4940 \newcommand*{\Xprenotes}[1]{\renewcommand{\Xprenotes@}{#1}}%
4941 \newcommand{\preXnotes}[1]{\led@warning@preXnotes@deprecated\Xprenotes{#1}}
    %For compatibility
4942 %

```

The same, but for familiar footnotes.

```

\Xprenotes 4943 \newtoggle{prenotesX@}
\Xprenotes 4944 \toggletrue{prenotesX@}
4945 \newcommand{\prenotesX@}{Opt}
4946 \newcommand*{\prenotesX}[1]{\renewcommand{\prenotesX@}{#1}}
4947 %

```

### XIX.3 Internal commands

```

firstXseries@ 4948 \gdef\firstXseries@{}
prepare@Xprenotes 4949 \newcommand{\prepare@Xprenotes}[1]{%
4950   \ifdimequal{Opt}{\Xprenotes@}%
4951   {}%
4952   {%
4953     \IfStrEq{\firstXseries@}{}{%
4954       \global\skip\csuse{#1footins}=\Xprenotes@%
4955       \global\advance\skip\csname #1footins\endcsname by\csuse{Xafterterrule@
#1}%
4956       \gdef\firstXseries@{#1}%
4957     }%
4958     {%
4959       \ifseriesbefore{#1}{\firstXseries@}%
4960       {%
4961         \global\skip\csuse{#1footins}=\csuse{Xbeforenotes@\firstXseries@}%
4962         \global\advance\skip\csname #1footins\endcsname by\csuse{Xafterterrule@
#1}%
4963         \gdef\firstXseries@{#1}%
4964       }%
4965       {%
4966       }%
4967     }%
4968   }
4969 %

```

The same thing is required for familiar notes and \prenotesX.

```

firstseriesX@ 4970 \gdef\firstseriesX@{}
prepare@prenotesX 4971 \newcommand{\prepare@prenotesX}[1]{%
4972   \ifdimequal{Opt}{\prenotesX@}%
4973   {}%

```



```

4974 {%
4975 \IfStrEq{\firstseriesX@}{}{%
4976 \global\skip\csuse{footins#1}=\prenotesX@%
4977 \global\advance\skip\csname footins#1\endcsname by\csuse{afterruleX@
#1}%
4978 \gdef\firstseriesX@{#1}%
4979 }%
4980 {%
4981 \ifseriesbefore{#1}{\firstseriesX@}%
4982 {%
4983 \global\skip\csuse{footins#1}=\csuse{beforenotesX@\firstseriesX@}%
4984 \global\advance\skip\csname footins#1\endcsname by\csuse{afterruleX@
#1}%
4985 \gdef\firstXseries@{#1}%
4986 }%
4987 {}%
4988 }%
4989 }%
4990 }
4991 %

```

## XX Endnotes

First, check the noend option.

```

4992 \ifbool{noend@}{}{%Used instead of \ifnoend@ to prevent expansion problem
4993 %

```

### XX.1 Internal commands

`\l@dend@open` and `\l@dend@close` are the macros that are used to open and close the endnote file. Note that all our writing to this file is `\immediate`: all page and line numbers for the endnotes are generated by the same mechanism we use for the footnotes, so that there is no need to defer any writing to catch information from the output routine. The argument of these two command is the series letter.

```

4994 \newcommand{\l@dend@open}[1]{%
4995 \global\booltrue{l@dend@#1}%
4996 \expandafter\immediate%
4997 \expandafter\openout%
4998 \csname l@d@#1end\endcsname%
4999 =\l@auxdir\jobname.#1end\relax%
5000 }%
5001 \newcommand{\l@dend@close}[1]{%
5002 \global\boolfalse{l@dend@#1}%
5003 \expandafter\immediate%
5004 \expandafter\closeout\csname l@d@#1end\endcsname%
5005 }%

```

```
5006
5007 %
```

**\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.

```
5008 \newcommand{\l@dend@stuff}{%
5009   \def\do##1{%
5010     \ifbool{l@dend@##1}{}%
5011     {\l@dend@open{##1}}%
5012     \expandafter\immediate\expandafter\write\csname l@d@##1end\endcsname{\
string\l@d@section{\the\section@num}\@percentchar}%
5013   }%
5014   \dolistloop{\@series}%
5015 }%
5016
5017 %
```

**\endprint** The \endprint here is nearly identical in its functioning to \normalfootfmt.  
**\l@d@section** The endnote file also contains \l@d@section commands, which supply the section numbers from the main text; standard reledmac does nothing with this information, but it is there if you want to write custom macros to do something with it. Arguments are:

- #1 Line numbers and font selection.
- #2 Lemma.
- #3 Note content.
- #4 Series.
- #5 Optional argument of \Xendnote.
- #6 Side (L or R).
- #7 Label for cross-referencing.

```
5018 \global\newcommand{\parapparatus@}{\long}\def\endprint#1#2#3#4#5#6#7{%
5019   \iftoggle{Xendfirstnote@#4}{%
5020     \global\togglefalse{Xendfirstnote@#4}%
5021     \csuse{Xendtxtbeforenotes@#4}%
5022   }{%
5023     \csuse{Xendbhooknote@#4}%
5024     \csuse{Xendnotefontsize@#4}%
5025     \hangindent=\csuse{Xendhangindent@#4}%
5026     \ifXendinsertsep@%
5027       \hskip\csuse{Xendafternote@#4}\relax%
5028     \csuse{Xendsep@#4}%
5029   }
```

```

5029 \else%
5030 \iftoggle{Xendparagraph@#4}%
5031 {\global\Xendinsertsep@true}%
5032 {}%
5033 \fi%
5034 \xdef\@currentseries{#4}%
5035 \def\do##1{%
5036 \setkeys[mac]{truefootnoteoption}{##1}%
5037 }%
5038 \notblank{#5}{\docsvlist{#5}}{}%
5039 \IfStrEq{#6}{R}{\ledRcol@true}{}%
5040 \def\@this@crossref@start{#7:start}%
5041 \def\@this@crossref@end{#7:end}%
5042 \printlineendnote{#1}{#4}%
5043 \IfStrEq{#6}{R}{\ledRcol@false}{}%
5044 \undef\@this@crossref@start%
5045 \undef\@this@crossref@end%
5046 \nottoggle{Xendlemmadisablefontselection@#4}%
5047 {\select@lemmafont#1|}%
5048 {}%
5049 \bgroup%
5050 \csuse{Xendlemmafont@#4}%
5051 \csuse{Xendwraplemma@#4}{#2}%
5052 \egroup%
5053 \ifboolexpr{%
5054 togl {nosep@}%
5055 or test{\ifcsempy{Xendlemmaseparator@#4}}%
5056 }%
5057 {\hskip\csuse{Xendinplaceoflemmaseparator@#4}\relax}%
5058 {\nobreak%
5059 \hskip\csuse{Xendbeforelemmaseparator@#4}%
5060 \csuse{Xendlemmaseparator@#4}%
5061 \hskip\csuse{Xendafterlemmaseparator@#4}%
5062 \relax%
5063 }%
5064 \csuse{Xendwrapcontent@#4}{#3}%
5065 \nottoggle{Xendparagraph@#4}{\par}{}%
5066 \def\do##1{%
5067 \setkeys[mac]{falsefootnoteoption}{##1}%
5068 }%
5069 \notblank{#5}{\docsvlist{#5}}{}%
5070 }}%
5071
5072 \let\l@d@section=\@gobble
5073
5074 %

```

**\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

```
\newcommand{\printlineendnote}[2]{%
\l@dp@rsefootspec#1|}%
\iftoggle{Xendnumberonlyfirstintwolines@#2}{%
\def\lineinfo@{\l@dparsedstartpage - \l@dparsedstartline - \
l@dparsedstartsub - \l@dparsedendpage - \l@dparsedendline - \
l@dparsedendsub}%
}%
}%
\def\lineinfo@{\l@dparsedstartpage - \l@dparsedstartline - \
l@dparsedstartsub}%
}%
\ifboolexpr{%
tog1 {nonum@}%
or tog1 {Xendnonumber@#2}%
}%
{\hspace{\csuse{Xendinplaceofnumber@#2}}}%
{%
\iftoggle{Xendnumberonlyfirstinline@#2}%
{\ifcsdef{prevendline#2}%
{\ifcsequal{prevendline#2}{\lineinfo@}%
}%
\csuse{Xendbhookinplaceofnumber@#2}%
\ifcsempy{Xendsymlinenum@#2}%
{\hspace{\csuse{Xendinplaceofnumber@#2}}}%
{\printsymlineendnotearea{#2}}%
\csuse{Xendahookinplaceofnumber@2}%
}%
{\printlineendnotearea{#1}{#2}}}%
{\printlineendnotearea{#1}{#2}}%
}%
{\printlineendnotearea{#1}{#2}}%We keep every time line
\csxdef{prevendline#2}{\lineinfo@}%
}%
}%
%
```

```
\printsymlineendnotearea07 \newcommand{\printsymlineendnotearea}[1]{%
5108 \hspace{\csuse{Xendbeforesymlinenum@#1}}%
5109 \csuse{Xendnotenumfont@#1}%
5110 \ifdimequal{\csuse{Xendboxsymlinenum@#1}}{\z@}%
5111 {\csuse{Xendsymlinenum@#1}}%
5112 {\hbox to \csuse{Xendboxsymlinenum@#1}%
5113 {\csuse{Xendsymlinenum@#1}\hfill}}%
5114 }%
5115 \hspace{\csuse{Xendaftersymlinenum@#1}}%
5116 }%
5117 %
```

`\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.)

```

5118 \newcommand{\printlineendnotearea}[2]{%
5119   \csuse{Xendbhooklinenumber@#2}%
5120   \hspace{\csuse{Xendbeforenumber@#2}}%
5121   \bgroup%
5122     \csuse{Xendnotenumfont@#2}%
5123     \ifdimequal{\csuse{Xendboxlinenum@#2}}{0pt}%
5124       {\printendlines#1||\ifledRcol@\@Rlineflag\fi}%
5125       {\leavevmode%
5126         \hbox to \csuse{Xendboxlinenum@#2}%
5127         {%
5128           \IfSubStr{RC}{\csuse{Xendboxlinenumalign@#2}}{\hfill}{}}%
5129           \printendlines#1||\ifledRcol@\@Rlineflag\fi%
5130           \IfSubStr{LC}{\csuse{Xendboxlinenumalign@#2}}{\hfill}{}}%
5131         }%
5132     \egroup%
5133     \hspace{\csuse{Xendafternumber@#2}}%
5134     \csuse{Xendahooklinenumber@#2}%
5135   }%
5136   %

```

## 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 5137 \newcommandx{\toendnotes}[2][1,usedefault]{%
\toendnotes* 5138   \ifboolexpr{bool{numbering} or bool{numberingR}}{%
5139     \def\do##1{%
5140       \expandafter\immediate\expandafter\write\csname l@d###1end\endcsname%
5141       {\unexpanded{#2}\@percentchar}%
5142     }%
5143     \ifstrempy{#1}%
5144       {\dolistloop{\@series}}%
5145       {\docsvlist{#1}}%
5146   }{\led@err@toendnotes@outsidenumbering}%
5147 }%
5148 \WithSuffix\newcommandx\toendnotes*[2][1,usedefault]{%
5149   \ifboolexpr{bool{numbering} or bool{numberingR}}{%
5150     \def\do##1{%
5151       \expandafter\immediate\expandafter\write\csname l@d###1end\endcsname%

```

```

5152      {#2\@percentchar}%
5153    }%
5154    \ifstrempy{#1}%
5155      {\dolistloop{\@series}}%
5156      {\docsvlist{#1}}%
5157    }\led@err@toendnotes@outsidenumbering}%
5158  }%
5159  %

```

### 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.

```

5160 \newif\ifXendinsertsep@
5161 \newcommand*{\doendnotes}[1]{%
5162   \l@dend@close{#1}%
5163   \begingroup
5164     \csxdef{prevpagenum@#1}{}%
5165     \csxdef{prevpagerange@#1}{}%
5166     \global\toggletrue{Xendfirstnote@#1}%
5167     \makeatletter
5168     \expandafter\let\csname #1end\endcsname=\endprint
5169     \input\l@auxdir\jobname.#1end%
5170     \global\Xendinsertsep@false%
5171   \endgroup}
5172 %

```

`\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.

```

5173 \newcommand*{\doendnotesbysection}[1]{%
5174   \l@dend@close{#1}%
5175   \csxdef{prevpagenum@#1}{}%
5176   \csxdef{prevpagerange@#1}{}%
5177   \global\expandafter\advance\csname #1end@bysection\endcsname by 1%
5178   \begingroup%
5179     \makeatletter%
5180     \def\l@d@section##1{%
5181       \global\toggletrue{Xendfirstnote@#1}%
5182       \ifnumequal{##1}{\csname #1end@bysection\endcsname}%
5183         {\cslet{#1end}{\endprint}}%
5184         {\cslet{#1end}{\@gobbleseven}}%
5185     }%
5186     \input\l@auxdir\jobname.#1end%

```

```

5187 \global\Xendinsertsep@false%
5188 \endgroup%
5189 }%
5190 %

```

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.

```

5191 }%
5192 %

```

**\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.

```

5193 \newcommand*{\setprintendlines}[6]{%
5194 \let\@annot@start@print\relax%
5195 \let\@annot@end@print\relax%
5196 \l@dpnumfalse%
5197 \l@ddashfalse%
5198 \l@d@elinfalse%
5199 %

```

First of all, we print the second page number only if the ending page number is different

```

5200 \ifnum#4=#1 \else
5201 \l@dpnumtrue
5202 \l@ddashtrue
5203 \fi
5204 %

```

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.

```

5205 \ifboolexpr{%
5206   bool{\l@dpnum}%
5207   or not test{\ifnumequal{#2}{#5}}}%
5208 }{%
5209   \l@d@elintrue%
5210   \l@ddashtrue%
5211   \l@d@elintrue
5212   \l@ddashtrue
5213   \unless\ifx\relax\annot@end%
5214     \def\@annot@end@print{%
5215       \l@wrapcs@ifnotemptybox{Xendwraplinenumannotation@{\@currentseries
5216 }{\@annot@end}%

```

```

5216     }%
5217     \fi%
5218 }%
5219 {}%
5220 %

```

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.

```

5221 \ifl@d@elin%
5222 \def\@annot@start@print{%
5223 \l@wrapcs@ifnotemptybox{Xendwraplinenumannotation@\@currentseries}{\
annot@start}%
5224 }%
5225 \else%
5226 \ifx\annot@start\annot@end%
5227 \unless\ifx\@annot@start\relax%
5228 \def\@annot@start@print{%
5229 \l@wrapcs@ifnotemptybox{Xendwraplinenumannotation@\@currentseries
}{\annot@start}%
5230 }%
5231 \fi%
5232 \else%
5233 \ifx\@annot@end@print\relax%
5234 \def\@annot@start@print{%
5235 \l@wrapcs@ifnotemptybox{Xendwraplinenumannotation@\@currentseries
}%
5236 \ifx\annot@start\empty\else%
5237 \annot@start%
5238 \ifdefined\linangesesep%
5239 \linangesesep%
5240 \else%
5241 \csuse{Xendlinangeseparator@\@currentseries}%
5242 \fi%
5243 \fi%
5244 \annot@end%
5245 }%
5246 }%
5247 \else%
5248 \let\@annot@start@print\@annot@end@print%
5249 \let\@annot@end@print\relax%
5250 \fi%
5251 \fi%
5252 \fi%
5253 %

```

We print the starting sub-line if it is nonzero.

```

5254 \l@d@ssubfalse
5255 \ifnum#3=0 \else

```



```

5256 \l@d@ssubtrue
5257 \fi
5258 %

```

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.

```

5259 \l@d@eslfalse
5260 \ifnum#6=0 \else
5261 \ifnum#6=#3
5262 \ifl@d@elin \l@d@esltrue \else \l@d@eslfalse \fi
5263 \else
5264 \l@d@esltrue
5265 \l@d@dashtrue
5266 \fi
5267 \fi%
5268 %

```

```

5269 \ifl@d@dash%
5270 \ifboolexpr{togl{fulllines@} or test{\ifcsemt{Xendtwolines@}\
@currentseries}}}%
5271 {}%
5272 {%
5273 \setistwofollowinglines{#1}{#2}{#4}{#5}%
5274 \ifboolexpr{%
5275 (%
5276 togl {Xendtwolinesbutnotmore@\@currentseries}%
5277 and not%
5278 (%
5279 bool {istwofollowinglines@}%
5280 )%
5281 )%
5282 or%
5283 (%
5284 (not test{\ifnumequal{#1}{#4}})%
5285 and togl{Xendtwolinesonlyinsamepage@\@currentseries}%
5286 )%
5287 }%
5288 {}%
5289 {%
5290 \l@d@dashfalse%
5291 \l@d@Xtwolinesttrue%
5292 \l@d@elinfalse%
5293 \l@d@eslfalse%
5294 \ifcsemt{Xendmorethantwolines@\@currentseries}%
5295 {}%
5296 {\ifistwofollowinglines@\else%
5297 \l@d@Xmorethantwolinesttrue%
5298 \fi%
5299 }%
5300 }%

```

```

5301 }%
5302 \fi%
5303 %

```

End of \setprintendlines.

```

5304 }%
5305 %

```

`\printendlines` Now we are ready to print it all.

```

5306 \def\printendlines#1|#2|#3|#4|#5|#6|#7|#8|{%
5307   \begingroup
5308   \setprintendlines{#1}-{#2}-{#3}-{#4}-{#5}-{#6}%
5309   %

```

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.

```

5310   \ifdimequal{\csuse{Xendboxstartlinenum@\@currentseries}}{0pt}%
5311   {\bgroup}%
5312   {\leavevmode\hbox to \csuse{Xendboxstartlinenum@\@currentseries}\bgroup
5313   \hfill}}%

```

Then, print the starting page number-

```

5314   \ifboolexpr{%
5315     (%
5316       test{\ifcsstring{prevpagenum@\@currentseries}{#1}}%
5317       and not%
5318       (togl{Xendpagenumberonlyfirstifsingle@\@currentseries} and bool{
5319         l@d@pnum})}%
5319     )%
5320     or%
5321     (%
5322       test {\ifcsstring{prevpagerange@\@currentseries}{#1-#4}}%
5323       )%
5324     )%
5325     {%
5326       \ifcsempy{Xendsympagenum@\@currentseries}%
5327       {\hspace{\csuse{Xendinplaceofpagenumber@\@currentseries}}}%
5328       {\csuse{Xendsympagenum@\@currentseries}}%
5329     }%
5330     {%
5331       \wrap@edcrossref{\@this@crossref@start}{\printpnum{#1}}%
5332     }%
5333   %

```

Then, determine what must be printed before the start line.

```

5334 \ifl@d@dash%
5335 \ifl@d@pnum%
5336 \csuse{Xendlineprefixsingle@ \@currentseries}%
5337 \else%
5338 \ifcsemtyp{Xendlineprefixmore@ \@currentseries}%
5339 {\csuse{Xendlineprefixsingle@ \@currentseries}}%
5340 {\csuse{Xendlineprefixmore@ \@currentseries}}%
5341 \fi%
5342 \else%
5343 \csuse{Xendlineprefixsingle@ \@currentseries}%
5344 \fi%
5345 %

```

Then print the starting line, followed, if needed, by the side flag and the starting sub line number, then the line number annotation.

```

5346 \ifcsstring{Xendlinenumannotationposition@ \@currentseries}{before}%
5347 {\@annot@start@print}%
5348 {}%
5349 \wrap@edcrossref{\@this@crossref@start}{%
5350 \ifledRcol@%
5351 \linenumrepR{#2}%
5352 \else%
5353 \linenumrep{#2}%
5354 \fi%
5355 }%
5356 \iftoggle{Xendlineflag@ \@currentseries}{\ifledRcol@ \@Rlineflag\fi}{}%
5357 \ifl@d@ssub%
5358 \csuse{Xendsublinesep@ \@currentseries}%
5359 \wrap@edcrossref{\@this@crossref@start}{%
5360 \ifledRcol@%
5361 \sublinenumrepR{#3}%
5362 \else%
5363 \sublinenumrep{#3}%
5364 \fi%
5365 }%
5366 \fi%
5367 \ifcsstring{Xendlinenumannotationposition@ \@currentseries}{after}%
5368 {\@annot@start@print}%
5369 {}%
5370 %

```

Close the box.

```

5371 \egroup%
5372 %

```

Open the box for the ending line number.

```

5373 \ifdimequal{\csuse{Xendboxendlinenum@ \@currentseries}}{0pt}%
5374 {\bgroup}%
5375 {\hbox to \csuse{Xendboxendlinenum@ \@currentseries}\bgroup}%
5376 %

```

Print the dash + the ending line number, or the line number range symbol.

```

5377 \ifl@d@Xtwolines%
5378 \ifl@d@Xmorethantwolines%
5379 \csuse{Xendmorethantwolines@ \@currentseries}%
5380 \else%
5381 \csuse{Xendtwolines@ \@currentseries}%
5382 \fi%
5383 \else%
5384 \ifl@d@dash%
5385 \ifdefined\linrangesep%
5386 \linrangesep%
5387 \else%
5388 \csuse{Xendlinrangeseparator@ \@currentseries}%
5389 \fi%
5390 \fi%
5391 %

```

Print the ending page number.

```

5392 \ifl@d@pnum%
5393 \ifcsstring{prevpagerange@ \@currentseries}{#1-#4}%
5394 {%
5395 \ifcempty{Xendsympagenum@ \@currentseries}%
5396 {\hspace{\csuse{Xendinplaceofpagenumber@ \@currentseries}}}%
5397 {\csuse{Xendsympagenum@ \@currentseries}}%
5398 }%
5399 {%
5400 \wrap@edcrossref{\@this@crossref@end}\printnpnum{#4}%
5401 }%
5402 \fi%
5403 %

```

Print the ending line number, with, if needed, the line prefix, and followed by the side flag, the subline number, and line number annotation.

```

5404 \ifcsstring{Xendlinenumannotationposition@ \@currentseries}{before}%
5405 {\@annot@end@print}%
5406 {%
5407 \ifl@d@elin%
5408 \ifl@d@pnum\csuse{Xendlineprefixsingle@ \@currentseries}\fi%
5409 \wrap@edcrossref{\@this@crossref@end}{%
5410 \ifledRcol%
5411 \linenumrepR{#5}%
5412 \else%
5413 \linenumrep{#5}%
5414 \fi%
5415 }%
5416 \iftoggle{Xendlineflag@ \@currentseries}{\ifledRcol\@Rlineflag\fi}{}%
5417 \fi%
5418 \ifl@d@esl%
5419 \ifl@d@elin%

```

```

5420 \csuse{Xendsublinesep@\@currentseries}%
5421 \fi%
5422 \wrap@edcrossref{\@this@crossref@end}{%
5423 \ifledRcol@%
5424 \sublinenumrepR{#6}%
5425 \else%
5426 \sublinenumrep{#6}%
5427 \fi%
5428 }%
5429 \fi%
5430 \ifcsstring{Xendlinenumannotationposition@\@currentseries}{after}%
5431 {\@annot@end@print}%
5432 {}%
5433 \fi%
5434 %

```

Close the ending line box.

```

5435 \ifdimequal{\csuse{Xendboxendlinenum@\@currentseries}}{0pt}%
5436 {}%
5437 {\hfill}%Prevent underfull hbox
5438 \egroup%
5439 %

```

And, finally, save, if needed, the current page number for the Xendpagenumberonlyfirst hooks.

```

5440 \iftoggle{Xendpagenumberonlyfirst@\@currentseries}%
5441 {\iftoggle{Xendpagenumberonlyfirstintwo@\@currentseries}%
5442 {\csxdef{prevpagerange@\@currentseries}{#1-#4}}%
5443 {\csxdef{prevpagenum@\@currentseries}{#4}}%
5444 }%
5445 {}%
5446 %

```

Now, the end of \printendlines macro.

```

5447 \endgroup%
5448 }%
5449
5450 %

```

**\printnpnum** A macro to print a page number in an endnote. Should not be override anymore

```

5451 \newcommand*{\printnpnum}[1]{\csuse{Xendbeforepagenumber@\@currentseries}
5452 }#1\csuse{Xendafterpagenumber@\@currentseries}}
5453 %

```

## 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@`.

```
5454 \newcommand{\newseries}[1]{%
5455     \def\do##1{\newseries@{##1}}%
5456     \docsvlist{#1}
5457 }
5458 %
```

**\@series** The `\series@` macro is an etoolbox list, which contains the name of all series.

```
5459 \newcommand{\@series}{}
5460 %
```

The command `\newseries@` creates a new series of the footnote.

```
\newseries@_61 \newcommand{\newseries@}[1]{
5462 %
```

### XXI.1 Test if series is still existing

```
5463 \xifinlist{#1}{\@series}{\led@warn@SeriesStillExist{#1}}%
5464 {%
5465 %
```

### XXI.2 Init specific to reledpar

When calling `\newseries@` after having loaded `reledpar`, we need to load specific setting.

```
5466 \ifdefined\newseries@par%
5467     \newseries@par{#1}%
5468 \fi%
5469 %
```

### XXI.3 For critical footnotes

Critical footnotes are those which start with letters. We look for the `\nocritical` option of `reledmac`.

```
5470 \unless\ifnocritical@
5471 %
```

## XXI.3.1 Options

```

5472 \newtoggle{Xlineflag@#1}
5473 \newtoggle{Xparindent@#1}
5474 \newtoggle{Xlemmadisablefontselection@#1}
5475 \csgdef{Xwrapcontent@#1}{}%
5476 \csgdef{Xbeforeinserting@#1}{}%
5477 \csgdef{Xhangindent@#1}{Opt}%
5478 \csgdef{Xragged@#1}{}%
5479 \csgdef{Xhsizetwocol@#1}{0.45 \hsize}%
5480 \csgdef{Xhsizethreecol@#1}{.3 \hsize}%
5481 \csgdef{Xcolalign@#1}{\raggedright}%
5482 \csgdef{Xnotenumfont@#1}{\normalfont}%
5483 \csgdef{Xnotefontsize@#1}{\footnotesize}%
5484 \csgdef{Xbhooknote@#1}{}%
5485 \csgdef{Xbhookgroup@#1}{}%
5486
5487 \csgdef{Xboxlinenum@#1}{Opt}%
5488 \csgdef{Xboxlinenumalign@#1}{L}%
5489
5490 \csgdef{Xboxstartlinenum@#1}{Opt}%
5491 \csgdef{Xboxendlinenum@#1}{Opt}%
5492
5493 \csgdef{Xboxsymlinenum@#1}{Opt}%
5494 \newtoggle{Xnumberonlyfirstinline@#1}%
5495 \newtoggle{Xgroupbyline@#1}%
5496 \newtoggle{Xgroupbylineseparatetwolines@#1}%
5497 \newtoggle{Xnumberonlyfirstintwolines@#1}%
5498 \csgdef{Xtwolines@#1}{}%
5499 \csgdef{Xmorethantwolines@#1}{}%
5500 \csgdef{Xsublinesep@#1}{\fullstop}%
5501 \csgdef{Xpagelinesep@#1}{\csname Xsublinesep@#1\endcsname}%for
backward compatibility, call Xsublinesep@#1
5502 \newtoggle{Xtwolinesbutnotmore@#1}%
5503 \newtoggle{Xtwolinesonlyinsamepage@#1}%
5504 \newtoggle{Xonlypstart@#1}%
5505 \newtoggle{Xpstarteverytime@#1}%
5506 \newtoggle{Xpstart@#1}%
5507 \newtoggle{Xstanza@#1}%
5508 \csgdef{Xstanzaseparator@#1}{}%
5509 \csgdef{Xsymlinenum@#1}{}%
5510 \newtoggle{Xnonumber@#1}%
5511 \csgdef{Xbeforenumber@#1}{Opt}%
5512 \csgdef{Xtxtbeforenumber@#1}{}%
5513 \csgdef{Xafternumber@#1}{0.5em}%
5514 \newtoggle{Xnonbreakableafternumber@#1}%
5515 \csgdef{Xbeforesymlinenum@#1}{\csuse{Xbeforenumber@#1}}%
5516 \csgdef{Xaftersymlinenum@#1}{\csuse{Xafternumber@#1}}%
5517 \csgdef{Xinplaceofnumber@#1}{1em}%
5518 \global\cslet{Xlemmaseparator@#1}{\rbracket}%

```

```

5519 \csgdef{Xbeforelemmaseparator@#1}{0em}%
5520 \csgdef{Xafterlemmaseparator@#1}{0.5em}%
5521 \csgdef{Xinplaceoflemmaseparator@#1}{1em}%
5522 \csgdef{Xbeforenotes@#1}{1.2em \@plus .6em \@minus .6em}
5523 \csgdef{Xafterrule@#1}{0pt}
5524
5525 \csgdef{Xtxtbeforenotes@#1}{}
5526 \newtoggle{Xtxtbeforenotes@#1@typeset}%Not directly used by user,
but internal
5527 \newtoggle{Xtxtbeforenotesonlyonce@#1}%
5528
5529 \csgdef{Xmaxhnotes@#1}{0.8\vsizel}
5530 \newtoggle{Xnoteswidthliketwocolumns@#1}%
5531 \csgdef{Xparafootsep@#1}{}%
5532 \csgdef{Xafternote@#1}{1em plus.4em minus.4em}
5533 \csgdef{Xlinrangeseparator@#1}{\endashchar}%
5534
5535 \csgdef{Xlemmafонт@#1}{}%
5536 \csgdef{Xwraplemma@#1}{}
5537 \csgdef{Xwidth@#1}{\hsizel}
5538 \csgdef{Xwraplinenumannotation@#1}{\textsuperscript}%
5539 %

```

### XXI.3.2 Create inserts, needed to add notes in foot

As regards inserts, see chapter 15 of *The TeXbook* by D. Knuth.

```

5540 \expandafter\newinsert\csname #1footins\endcsname%
5541 \unless\ifnoledgroup@%
5542 \expandafter\newinsert\csname mp#1footins\endcsname%
5543 \fi%
5544 %

```

### 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.

```

5545 \global\newbool{parapparatus@}{\expandafter\newcommand\expandafter
*}{\expandafter\newcommand}\csname #1footnote\endcsname[2] [] {}%
5546 \if@edtext@secondarg@%
5547 \ifledRcol%
5548 \ifcsstring{Xonlyside@#1}{L}{\
led@error@note@called@onrightside{#1footnote}}{}%
5549 \else%
5550 \ifcsstring{Xonlyside@#1}{R}{\
led@error@note@called@onleftside{#1footnote}}{}%
5551 \fi%
5552 \begingroup%
5553 \newcommand{\content}{##2}%
5554 \ifnumberedpar@%

```



```

5555         \ifledRcol%
5556             \ifluatex%
5557                 \footnotelang@lua[R]%
5558             \fi%
5559         \@ifundefined{xpg@main@language}%if polyglossia
5560             {}%
5561             {\footnotelang@poly[R]}%
5562         \footnoteoptions@{R}{##1}{true}%
5563         \xright@appenditem{%
5564             \ifbool{indtl@innote}%
5565                 {\unexpanded{\let\index\nindex}}%
5566             {}%
5567         \ifbool{indtl@notenumber}%
5568             {\unexpanded{\let\index\nindex}}%There is no note
...number so
5569             {}%
5570             \noexpand\Xnote@true%
5571             \noexpand\prepare@Xprenotes{#1}%
5572             \noexpand\prepare@edindex@fornote{\l@d@nums}%
5573             \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.
5574             \noexpand\setcounter{stanzaR}{\the\c@stanzaR}%Save
stanzaR counter for footnote
5575             \unexpanded{\def\@this@crossref@start}{\theedtext:
start}%
5576             \unexpanded{\def\@this@crossref@end}{\theedtext:end}%
5577             \expandonce{\@beforeinsertofthisedtext}% Internal for
now, no reason to make it public
5578             \noexpand\parse@annot\l@current@annotR|%
5579             \noexpand\csuse{v#1footnote}{#1}%
5580             {\l@d@nums}{\expandonce\@tag}{\expandonce\content}}
%
5581             \noexpand\Xnote@false%
5582             \unexpanded{\advance\@edindex@fornote@m@ne}%
5583             \unexpanded{\undef\@this@crossref@start}%
5584             \unexpanded{\undef\@this@crossref@end}%
5585             \ifbool{indtl@innote}%
5586                 {\unexpanded{\let\index\orig@@index}}%
5587             {}%
5588             \ifbool{indtl@notenumber}%
5589                 {\unexpanded{\let\index\orig@@index}}%
5590             {}%
5591             }\to\inserts@listR
5592             \footnoteoptions@{R}{##1}{false}%
5593             \global\advance\insert@countR \@ne%
5594         \else%
5595             \ifluatex%
5596                 \footnotelang@lua%

```

```

5597         \fi%
5598         \@ifundefined{xpg@main@language}%if polyglossia
5599             {}%
5600             {\footnotelang@poly}%
5601         \footnoteoptions@{L}{#1}{true}%
5602         \xright@appenditem{%
5603             \ifbool{indtl@innote}%
5604                 {\unexpanded{\let\index\nindex}}%
5605                 {}%
5606             \ifbool{indtl@notenumber}%
5607                 {\unexpanded{\let\index\nindex}}%There is no note
...number so
5608             {}%
5609             \noexpand\Xnote@true%
5610             \noexpand\prepare@Xprenotes{#1}%
5611             \noexpand\prepare@edindex@fornote{\l@d@nums}%
5612             \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.
5613             \ifl@dpairing%
5614             \noexpand\setcounter{stanzaL}{\the\c@stanzaL}%Save
stanzaR counter for footnote
5615             \fi%
5616             \unexpanded{\def\@this@crossref@start}{\theedtext:
start}%
5617             \unexpanded{\def\@this@crossref@end}{\theedtext:end}%
5618             \expandonce{\@beforeinsertofthisedtext}%Internal for
now, no reason to make it public
5619             \noexpand\parse@annot\l@current@annot|%
5620             \noexpand\csuse{v#1footnote}%
5621             {#1}%
5622             {\l@d@nums}{\expandonce\@tag}{\expandonce\content
}}%
5623             \unexpanded{\undef\@this@crossref@start}%
5624             \unexpanded{\undef\@this@crossref@end}%
5625             \noexpand\Xnote@false%
5626             \unexpanded{\advance\@edindex@fornote@m@ne}%
5627             \ifbool{indtl@innote}%
5628                 {\unexpanded{\let\index\orig@@index}}%
5629                 {}%
5630             \ifbool{indtl@notenumber}%
5631                 {\unexpanded{\let\index\orig@@index}}%
5632                 {}%
5633             }\to\inserts@list
5634             \global\advance\insert@count \@ne%
5635             \footnoteoptions@{L}{#1}{false}%
5636         \fi
5637     \else
5638         \csuse{v#1footnote}{#1}{\{0|0|0|0|0|0|0\}}{#1}%

```

```

5639         \fi%
5640     \endgroup%
5641     \else%
5642         \led@err@FootnoteNotInSecondArgEdtext{#1}%
5643     \fi%
5644     \ignorespaces%
5645     }
5646 %

```

Create counter used to determine on which page the previous note was called.

```

5647     \expandafter\newcount\csname #1prevpage@num\endcsname%
5648     \expandafter\newcount\csname #1prevpage@numR\endcsname%
5649 %

```

We need to be able to modify reledmac's footnote macros and restore their

```

5650     \global\csletcs{#1@@footnote}{#1footnote}
5651 %

```

### XXI.3.4 Set standard display

```

5652     \Xarrangement@normal{#1}%
5653 %

```

End of for critical footnotes.

```

5654     \fi
5655 %

```

## XXI.4 For familiar footnotes

Familiar footnotes are those which end with letters. We look for the `nofamiliar` option of `reledmac`.

```

5656     \unless\ifnofamiliar@
5657 %

```

### XXI.4.1 Options

```

5658     \newtoggle{parindentX@#1}
5659     \csgdef{wrapcontentX@#1}{}%
5660     \csgdef{hangindentX@#1}{Opt}%
5661     \csgdef{beforeinsertingX@#1}{}%
5662     \csgdef{raggedX@#1}{}%
5663     \csgdef{hsizetwocolX@#1}{0.45 \hsize}%
5664     \csgdef{hsizethreecolX@#1}{.3 \hsize}%
5665     \csgdef{colalignX@#1}{\raggedright}%
5666     \csgdef{notenumfontX@#1}{\normalfont}%
5667     \csgdef{notefontsizeX@#1}{\footnotesize}%
5668     \csgdef{bhooknoteX@#1}{}%

```

```

5669 \csgdef{bhookgroupX@#1}{}%
5670 \csgdef{afterruleX@#1}{Opt}
5671 \csgdef{beforenotesX@#1}{1.2em \@plus .6em \@minus .6em}
5672 \csgdef{maxhnotesX@#1}{0.8\vsizex}%
5673 \newtoggle{noteswidthliketwocolumnsX@#1}%
5674 \csgdef{parafootsepX@#1}{}%
5675 \csgdef{afternoteX@#1}{1em plus.4em minus.4em}
5676 \csgdef{widthX@#1}{\hsizex}%
5677 \csgdef{txtbeforenotesX@#1}{}%
5678 \newtoggle{txtbeforenotesX@#1@typeset}%Not directly used by user,
but internal
5679 \newtoggle{txtbeforenotesonlyonceX@#1}%
5680 \csgdef{Xlinenumannotationposition@#1}{after}%
5681 % End of for familiar footnotes.
5682 % \subsubsection{Create inserts, needed to add notes in foot}
5683 % As regards inserts, see chapter 15 of the TeXBook by D. Knuth.
5684 % \begin{macrocode}
5685 \expandafter\newinsert\csname footins#1\endcsname%
5686 \unless\ifnoledgroup%
5687 \expandafter\newinsert\csname mpfootins#1\endcsname%
5688 \fi%
5689 %

```

#### 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.

```

5690 \global\expandafter\newcommand\csname footnote#1\endcsname[1]{%
5691 \begingroup%
5692 \prepare@prenotesX{#1}%
5693 \newcommand{\content}{##1}%
5694 %
5695 %

```

If we use the `\csquotes` package, we reset the quotation level.

```

5696 \ifdefined\csq@qllevel%
5697 \csq@qllevel=0\relax%
5698 \fi%
5699 %

```

If we are preparing parallel typesetting, we cannot just increase the footnote counter. Read `reledpar`'s handbook about that (V.1.2 p. 53).

```

5700 \global\expandafter\advance\csname footnote#1@reading\
endcsname by \@ne%
5701 \ifboolexpr{bool{!@dpairing} or bool{!@dprintingpages} or
bool{!@dprintingcolumns}}{%
5702 \ifcsdef{footnote#1@reading\the\csname footnote#1@reading\
endcsname=typeset}%
5703 {\setcounter{footnote#1}{\csuse{footnote#1@reading\the\
csname footnote#1@reading\endcsname=typeset}}}%

```

```

5704         {\setcounter{footnote#1}{\the\csname footnote#1@reading
\endcsname}}}%
5705         }{%
5706         \stepcounter{footnote#1}%
5707         }%
5708 %

```

We also have to check consistency with `\onlysideX` setting.

```

5709         \ifledRcol%
5710         \ifcsstring{onlysideX@#1}{L}{\
led@error@note@called@onrightside{footnote#1}}{}%
5711         \else%
5712         \ifcsstring{onlysideX@#1}{R}{\
led@error@note@called@onleftside{footnote#1}}{}%
5713         \fi%
5714 %

```

And now, the feature not depending of wether we are preparing parallel typesetting

```

5715         \protected@csxdef{@thefnmark#1}{\csuse{thefootnote#1}}%
5716         \nottoggle{nomk@}%Nomk is set to true when using \
footnoteXnomk with \parpackage
5717         {\csuse{@footnotemark#1}}%
5718         {}%
5719         \ifluatex%
5720         \xdef\footnote@luatextextdir{\the\textdir}%
5721         \xdef\footnote@luatexpardir{\the\pardir}%
5722         \fi%
5723         \if@ledgroup%
5724         \led@set@index@fornote{#1}%
5725         \fi%
5726         \csuse{vfootnote#1}{#1}{\expandonce\content}\m@mmf@prepare%
5727         \ifbool{indtl@innote}%
5728         {\let\index\orig@@index}%
5729         {}%
5730         \ifbool{indtl@notenumber}%
5731         {\let\index\orig@@index}%
5732         {}%
5733         \endgroup%
5734     }
5735 %

```

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).

```

5736         \newcounter{footnote#1}
5737         \global\expandafter\renewcommand\csname thefootnote#1\endcsname{\
arabic{footnote#1}}
5738         \expandafter\newcount\csname footnote#1@reading\endcsname%
5739 %

```

Create counter used to determine on which page the previous note was called.

```
5740 \expandafter\newcount\csname prevpage#1@num\endcsname%
5741 \expandafter\newcount\csname prevpage#1@numR\endcsname%
5742 %
```

Add `\let\footnoteX\@gobble` to `\no@expands`.

```
5743 \expandafter\gappto\expandafter\no@expands\expandafter{\expandafter\
let\csname footnote#1\endcsname\@gobble}%
5744 %
```

And now, define `\footnoteXmark` and `\footnoteXtext`, equivalent to classical `\footnotemark` and `\footnotetext`.

```
5745 \expandafter\newcommand\csname footnote#1mark\endcsname{%
5746 \begingroup%
5747 \prepare@prenotesX{#1}%
5748 \stepcounter{footnote#1}%
5749 \protected@csxdef{@thefnmark#1}{\csuse{thefootnote#1}}%
5750 \csuse{@footnotemark#1}%
5751 \m@mmf@prepare%
5752 \endgroup%
5753 }%
5754 \expandafter\newcommand\csname footnote#1text\endcsname[1]{%
5755 \begingroup%
5756 \csuse{vfootnote#1}{#1}{\expandonce{##1}}%
5757 \endgroup%
5758 }%
5759 %
```

Do not forget to initialize the series.

```
5760 \arrangementX@normal{#1}%
5761 \fi
5762 %
```

## XXI.5 The endnotes

Endnotes are commands like `\Xendnote`, where `X` is a series letter. First, we check for the `noend` options.

```
5763 \unless\ifnoend@
5764 %
```

### 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`

```

5765 \expandafter\newwrite\csname l@d@#1end\endcsname%
5766 \expandafter\newif\csname ifl@dend@#1\endcsname%
5767 %

```

### 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.

```

5768
5769 \global\expandafter\newcommand\csname #1endnote\endcsname[2][1,
usedefault]{%
5770 \bgroup%
5771 \newlinechar='40%
5772 \global\@noneed@Footnotetrue%
5773 \newcommand{\content}{##2}%
5774 \stepcounter{labidx}%
5775 \expandafter\immediate\expandafter\write\csname l@d@#1end\
endcsname{%
5776 \unexpanded{\def\sw@list@inedtext}%
5777 {\expandafter\unexpanded\expandafter{\sw@inthisedtext}}}%
5778 \@percentchar\space%Explicit space, to add a linebreak in the
output file
5779 \noexpand\parse@annot\l@current@annot|\@percentchar\space%
5780 \expandafter\string\csname #1end\endcsname%
5781 {\ifnumberedpar@l@d@nums\fi}%
5782 {\ifnumberedpar@\expandonce\@tag\fi}%
5783 {\expandonce\content}%
5784 {\#1}%
5785 {\unexpanded{##1}}}%
5786 {\ifledRcol R\else L\fi}%
5787 {\theedtext}%
5788 \@percentchar%
5789 }%
5790 \egroup%
5791 \ignorespaces%
5792 }%
5793 %

```

### XXI.5.3 Tools

The `\Xtoendnotes` command inserts any arbitrary content into the endnote file. It is an alias of the more generalist `\addtoendnotes`

```

5794
5795 \global\expandafter\newcommand\csname #1toendnotes\endcsname[1]{%
5796 \toendnotes[#1]{##1}%

```

```

5797 }%
5798
5799 \expandafter\WithSuffix\expandafter\newcommand\csname #1toendnotes\
endcsname*[1]{%
5800 \toendnotes*[#1]{##1}%
5801 }%
5802
5803 %

```

#### 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.

```

5804
5805 \global\cslet{#1end}{\@gobbleseven}
5806 %

```

We need to store the number of times \doendnotesbysection is called for one series.

```

5807 \global\expandafter\newcount\csname #1end@bysection\endcsname%
5808 %

```

#### XXI.5.5 The options

```

5809 \csgdef{Xendwraplemma@#1}{%
5810 \csgdef{Xendwrapcontent@#1}{}%
5811 \csgdef{Xendtwolines@#1}{}%
5812 \csgdef{Xendmoreethantwolines@#1}{}%
5813 \newtoggle{Xendtwolinesbutnotmore@#1}{}%
5814 \newtoggle{Xendtwolinesonlyinsamepage@#1}{}%
5815 \newtoggle{Xendlemmadisablefontselection@#1}%
5816 \csgdef{Xendnotenumfont@#1}{\normalfont}%
5817 \csgdef{Xendnotefontsize@#1}{\footnotesize}%
5818 \csgdef{Xendbhooknote@#1}{}%
5819
5820 \csgdef{Xendsublinesep@#1}{\fullstop}%
5821
5822 \csgdef{Xendbeforenumber@#1}{0pt}
5823 \csgdef{Xendafternumber@#1}{0.5em}
5824
5825 \csgdef{Xendboxlinenum@#1}{0pt}%
5826 \csgdef{Xendboxlinenumalign@#1}{L}%
5827
5828 \csgdef{Xendboxstartlinenum@#1}{0pt}%
5829 \csgdef{Xendboxendlinenum@#1}{0pt}%

```



```

5830 \csgdef{Xendlemmaseparator@#1}{}%
5831 \csgdef{Xendbeforelemmaseparator@#1}{0em}%
5832 \csgdef{Xendafterlemmaseparator@#1}{0.5em}%
5833 \csgdef{Xendinplaceoflemmaseparator@#1}{0.5em}%
5834
5835
5836 \newtoggle{Xendparagraph@#1}%
5837 \csgdef{Xendafternote@#1}{1em plus.4em minus.4em}%
5838 \csgdef{Xendsep@#1}{}%
5839
5840 \csgdef{Xendinplaceofnumber@#1}{Opt}%
5841 \newtoggle{Xendnonnumber@#1}%
5842
5843 \csgdef{Xendhangindent@#1}{Opt}%
5844 \newtoggle{Xendnumberonlyfirstinline@#1}%
5845 \newtoggle{Xendnumberonlyfirstintwolines@#1}%
5846
5847 \csgdef{Xendbeforesymlinenum@#1}{\csuse{Xendbeforenumber@#1}}%
5848 \csgdef{Xendaftersymlinenum@#1}{\csuse{Xendafternumber@#1}}%
5849 \csgdef{Xendsymlinenum@#1}{}%
5850 \csgdef{Xendboxsymlinenum@#1}{Opt}%
5851
5852 \csgdef{Xendbhooklinenum@#1}{}%
5853 \csgdef{Xendehooklinenum@#1}{}%
5854 \csgdef{Xendbhookinplaceofnumber@#1}{}%
5855 \csgdef{Xendehookinplaceofnumber@#1}{}%
5856
5857 \csgdef{Xendlinangeseparator@#1}{\endashchar}%
5858
5859 \csgdef{Xendbeforepagenumber@#1}{p.}%
5860 \csgdef{Xendafterpagenumber@#1}{) }%
5861 \csgdef{Xendlineprefixsingle@#1}{}%
5862 \csgdef{Xendlineprefixmore@#1}{}%
5863
5864 \newtoggle{Xendlineflag@#1}
5865
5866 \csgdef{Xendlemmafont@#1}{}%
5867
5868 \csgdef{Xendlinenumannotationposition@#1}{after}%
5869 \csgdef{Xendwraplinenumannotation@#1}{\textsuperscript}%
5870
5871 \newtoggle{Xendpagenumberonlyfirst@#1}%
5872 \newtoggle{Xendpagenumberonlyfirstifsingle@#1}%
5873 \newtoggle{Xendpagenumberonlyfirstintwo@#1}%
5874 \csgdef{Xendsympagenum@#1}{}%
5875 \csgdef{Xendinplaceofpagenumber@#1}{Opt}%
5876
5877 \csgdef{Xendtxtbeforenotes@#1}{}%
5878 \newtoggle{Xendfirstnote@#1}%Not a hook, but used to apply
Xendtxtbeforenotes

```

```

5879 %
5880 %
End of endnotes declaration
5881 \fi%
5882 %
Dump series in \@series
5883 \listxadd{\@series}{#1}
5884 }
5885 }% End of \newseries
5886 %

```

## XXI.6 Init standards series (A,B,C,D,E)

```

5887 \expandafter\newseries\expandafter{\default@series}
5888 %

```

## 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.

```

5889 \newcommand{\seriesatbegin}[1]{%
5890 \StrDel{\@series}{#1}[\@series]%
5891 \edef\@new{}%
5892 \listead{\@new}{#1}%
5893 \listead{\@new}{\@series}%
5894 \xdef\@series{\@new}%
5895 }
5896 %

```

**\seriesatend** And `\seriesatend` moves the series to the end of the list.

```

5897 \newcommand{\seriesatend}[1]{%
5898 \StrDel{\@series}{#1}[\@series]%
5899 \edef\@new{}%
5900 \listead{\@new}{\@series}%
5901 \listead{\@new}{#1}%
5902 \xdef\@series{\@new}%
5903 }
5904 %

```

### XXII.2 Test series order

**\ifseriesbefore** `\ifseriesbefore{<seriesA>}{<seriesB>}{<true>}{<false>}` expands `<true>` if `<seriesA>` is printed before `<seriesB>`, expands `<false>` otherwise.

```

5905 \newcommand{\ifseriesbefore}[4]{%
5906   \StrPosition{\@series}{#1}[\@first]%
5907   \StrPosition{\@series}{#2}[\@second]%
5908   \ifnumgreater{\@second}{\@first}{#3}{#4}%
5909 }
5910 %

```

### XXII.2.1 Get the first series

In some specific case, we need to know the first series of the list of series.

```

\@getfirstseries:11 \newcommand{\@getfirstseries}{%
5912   \ifdefempty{\@series}%
5913   {\xdef\@firstseries{}}%
5914   {\StrChar{\@series}{1}[\@firstseries]}%
5915 }%
5916 %

```

## 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`.

```

5917 \newcommandx{\settoggle@series}[5][4,5,usedefault]{%
5918   \def\do##1{%
5919     \global\settoggle{#2@##1}{#3}%
5920     \ifstrequal{#4}{critical}{
5921       \csuse{Xarrangement@}\csuse{series@display##1}}{##1}%
5922     }{}
5923     \ifstrequal{#4}{familiar}{
5924       \csuse{arrangementX@}\csuse{series@displayX##1}}{##1}%
5925     }{}
5926   }%
5927   \ifstreempty{#1}{%
5928     \dolistloop{\@series}%
5929     \ifstreempty{#5}{}%
5930     \docsvlist{#5}%
5931   }
5932   }%
5933   {%
5934     \docsvlist{#1}%
5935   }%
5936 }
5937 %

```

`\setcommand@series` `\setcommand@series{<series>}{<command>}{<value>}` is a generic command to store hook's value into commands specific to some series. The arguments are:

- #1 (mandatory): the series for which the hooks should be set. If empty, all the series will be affected.
- #2 (mandatory): the name of the hook.
- #3 (mandatory): the new value of the hook/command.
- #4 (optional): if equal to `reload`, reload the footnote setting (call `\footnormal` or `\footparagraph` or ... depending of the footnote display).
- #5 (optional): if not empty, and if #1 is empty, change the hook setting for pseudo-series, as `appref`.

```

5938 \newcommandx{\setcommand@series}[5][4,5,usedefault]{%
5939   \def\do##1{
5940     \csgdef{#2@##1}{#3}
5941     \ifstrequal{#4}{critical}{%
5942       \csuse{Xarrangement@}\csuse{series@display##1}{##1}%
5943     }{}
5944     \ifstrequal{#4}{familiar}{%
5945       \csuse{arrangementX@}\csuse{series@displayX##1}{##1}%
5946     }{}%
5947   }%
5948   \ifstrepty{#1}{%
5949     \dolistloop{\@series}%
5950     \ifstrepty{#5}{}%
5951     \docsvlist{#5}
5952   }
5953 }%
5954 {%
5955   \docsvlist{#1}%
5956 }%
5957 }%
5958 %

```

### 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><pseudoseris>`, like `\Xtwolinesappref`.

```

5959 \newcommandx{\newhookcommand@series}[2][2,usedefault]{%
5960   \global\expandafter\newcommand\expandafter*\csname #1\endcsname[2][]{%
5961     \setcommand@series{##1}{#1}{##2}[] [#2]%
5962   }%
5963   \ifstrepty{#2}{}%
5964   \def\do##1{%
5965     \global\expandafter\newcommand\expandafter*\csname #1##1\endcsname
5966     [1]{%
5967       \csuse{#1}[##1]{####1}%
5968     }%
5969     \docsvlist{#2}%
5970   }%
5971 }
5972 %

```

**\newhooktoggle@series** `\newhooktoggle@series` command names is a generic command to add new commands for a new toggle hook, like `\Xnumberonlyfirstinline`. The second argu-

ment is also used to create commands named `\<hookname><pseudoseris>`, like `\Xtwolinesbutnotmoreappref`.

```

5973 \newcommandx{\newhooktoggle@series}[2][2,usedefault]{%
5974   \global\expandafter\newcommandx\expandafter*\csname #1\endcsname[2][1,2={
true},usedefault]{%
5975     \settoggle@series{##1}{#1}{##2}[][#2]%
5976   }%
5977   \ifstrempy{#2}{}{%
5978     \def\do##1{%
5979       \global\expandafter\newcommand\expandafter*\csname #1##1\endcsname{%
5980         \csuse{#1}[##1]%
5981       }%
5982     }%
5983     \docsvlist{#2}%
5984   }%
5985 }
5986 %

```

`\newhooktoggle@series@reload` `\newhookcommand@toggle@reload` does the same thing as `\newhooktoggle@series` but the commands created by this macro also reload the series arrangement, depending of type of notes

```

5987 \newcommand{\newhooktoggle@series@reload}[2]{%
5988   \global\expandafter\newcommandx\expandafter*\csname #1\endcsname[2][1,2={
true},usedefault]{%
5989     \settoggle@series{##1}{#1}{##2}[#2]%
5990   }%
5991 }%
5992 %

```

`\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.

```

5993 \newcommand{\newhookcommand@series@reload}[2]{%
5994   \global\expandafter\newcommand\expandafter*\csname #1\endcsname[2][]{%
5995     \setcommand@series{##1}{#1}{##2}[#2]%
5996   }%
5997 }
5998 %

```

### 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.

```

5999 \unless\ifnocritical@
6000   \newhookcommand@series{Xwrapcontent}%
6001   \newhookcommand@series{Xbeforeinserting}%

```

```

6002 \newhookcommand@series{Xlemmafont}%
6003 \newhookcommand@series{Xwraplemma}%
6004 \newhooktoggle@series{Xparindent}
6005 \newhookcommand@series{Xhangindent}
6006 \newhookcommand@series{Xragged}
6007 \newhookcommand@series{Xhsizetwocol}
6008 \newhookcommand@series{Xhsizethreecol}
6009 \newhookcommand@series{Xcolalign}%
6010 \newhookcommand@series{Xnotenumfont}
6011 \newhookcommand@series{Xbhooknote}
6012 \newhookcommand@series@reload{Xbhookgroup}{critical}
6013 \newhookcommand@series{Xboxsymlinenum}%
6014 \newhookcommand@series{Xsymlinenum}
6015 \newhookcommand@series{Xbeforenumber}
6016 \newhookcommand@series{Xtxtbeforenumber}
6017 \newhookcommand@series{Xafternumber}
6018 \newhookcommand@series{Xbeforesymlinenum}
6019 \newhookcommand@series{Xaftersymlinenum}
6020 \newhookcommand@series{Xinplaceofnumber}
6021 \newhookcommand@series{Xlemmaseparator}
6022 \newhookcommand@series{Xbeforelemmaseparator}
6023 \newhookcommand@series{Xafterlemmaseparator}
6024 \newhookcommand@series{Xinplaceoflemmaseparator}
6025 \newhookcommand@series{Xtxtbeforenotes}
6026 \newhooktoggle@series{Xtxtbeforenotesonlyonce}%
6027 \newhookcommand@series@reload{Xafterrule}{critical}
6028 \newhooktoggle@series{Xnumberonlyfirstinline}
6029 \newhooktoggle@series{Xnumberonlyfirstintwolines}
6030 \newhooktoggle@series{Xgroupbyline}%
6031 \newhooktoggle@series{Xgroupbylineseparatetwolines}%
6032 \newhooktoggle@series{Xnonumber}
6033 \newhooktoggle@series{Xpstart}
6034 \newhooktoggle@series{Xpstarteverytime}%
6035
6036 \newhooktoggle@series{Xstanza}%
6037 \newhookcommand@series{Xstanzaseparator}%
6038
6039 \newhooktoggle@series{Xonlypstart}
6040 \newhooktoggle@series{Xnonbreakableafternumber}
6041 \newhooktoggle@series{Xlemmadisablefontselection}
6042 \newhookcommand@series@reload{Xmaxhnotes}{critical}
6043 \newhookcommand@series@reload{Xbeforenotes}{critical}
6044 \newhooktoggle@series@reload{Xnoteswidthliketwocolumns}{critical}%
6045 \newhookcommand@series@reload{Xnotefontsize}{critical}
6046
6047 \newhookcommand@series{Xboxlinenum}%
6048 \newhookcommand@series{Xboxlinenumalign}%
6049
6050 \newhookcommand@series{Xboxstartlinenum}%
6051 \newhookcommand@series{Xboxendlinenum}%

```

```

6052 \newhookcommand@series{Xafternote}%
6053 \newhookcommand@series{Xparafootsep}
6054 \newhookcommand@series@reload{Xwidth}{critical}%
6055
6056 \ifundef{\Xhsize}%
6057 {%
6058   \newcommandx{\Xhsize}[2][1,usedefault]{%
6059     \led@warning@Xhsize@deprecated%
6060     \Xwidth[#1]{#2}%
6061   }%
6062 }%
6063 {}%
6064 {}%
6065 \fi
6066 %

```

Now, hooks for critical notes which also apply to crossreferencing and line numbering at the sides of the page.

```

6067 \newhooktoggle@series{Xlineflag}[appref,SEref]
6068 \newhookcommand@series{Xtwolines}[appref,SEref]
6069 \newhookcommand@series{Xmorethantwolines}[appref,SEref]
6070 \newhookcommand@series{Xsublinesep}[appref,SEref,side]%
6071 \newhookcommand@series{Xpagelinesep}[appref,SEref,side]%
6072 \newhooktoggle@series{Xtwolinesbutnotmore}[appref,SEref]
6073 \newhooktoggle@series{Xtwolinesonlyinsamepage}[appref,SEref]
6074 \newhookcommand@series{Xlinerrangeseparator}[appref,SEref]
6075 \newhookcommand@series{Xlinenumannotationposition}[side,appref,SEref]%
6076 \newhookcommand@series{Xwraplinenumannotation}[side,ref,appref,SEref]%
6077 %

```

### XXII.3.5 Options for familiar notes

Before generating the optional commands for familiar notes, we check the `\nofamiliar` option.

```

6078 \unless\ifnofamiliar@
6079   \newhookcommand@series{wrapcontentX}%
6080   \newhookcommand@series{beforeinsertingX}%
6081   \newhooktoggle@series{parindentX}
6082   \newhookcommand@series{hangindentX}
6083   \newhookcommand@series{raggedX}
6084   \newhookcommand@series{hsizetwocolX}
6085   \newhookcommand@series{hsizethreecolX}
6086   \newhookcommand@series{colalignX}%
6087   \newhookcommand@series{notenumfontX}
6088   \newhookcommand@series{bhooknoteX}
6089   \newhookcommand@series@reload{bhookgroupX}{familiar}
6090   \newhookcommand@series@reload{beforenotesX}{familiar}
6091   \newhookcommand@series@reload{maxhnotesX}{familiar}

```



```

6092 \newhooktoggle@series@reload{noteswidthliketwocolumnsX}{familiar}%
6093 \newhookcommand@series@reload{afterruleX}{familiar}
6094 \newhookcommand@series@reload{notefontsizeX}{familiar}
6095 \newhookcommand@series{afternoteX}
6096 \newhookcommand@series{parafootsepX}
6097 \newhookcommand@series{txtbeforenotesX}%
6098 \newhooktoggle@series{txtbeforenotesonlyonceX}%
6099 \newhookcommand@series@reload{widthX}{familiar}%
6100 \ifundef{\hsizeX}%
6101 {%
6102   \newcommandx{\hsizeX}[2][1,usedefault]{%
6103     \led@warning@hsizeX@deprecated%
6104     \widthX[#1]{#2}%
6105   }%
6106 }%
6107 {}%
6108 \fi
6109 %

```

### 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.

```

6110 \unless\ifnoend@
6111   \newhookcommand@series{Xendwraplemma}
6112   \newhookcommand@series{Xendwrapcontent}
6113   \newhookcommand@series{Xendnotenumfont}
6114   \newhookcommand@series{Xendlemmafont}%
6115   \newhookcommand@series{Xendbhooknote}
6116
6117   \newhookcommand@series{Xendboxlinenum}%
6118   \newhookcommand@series{Xendboxlinenumalign}%
6119
6120   \newhookcommand@series{Xendboxstartlinenum}%
6121   \newhookcommand@series{Xendboxendlinenum}%
6122
6123   \newhookcommand@series{Xendnotefontsize}
6124   \newhooktoggle@series{Xendlemmadisablefontselection}
6125   \newhookcommand@series{Xendlemmaseparator}
6126   \newhookcommand@series{Xendbeforelemmaseparator}
6127   \newhookcommand@series{Xendafterlemmaseparator}
6128   \newhookcommand@series{Xendinplaceoflemmaseparator}
6129
6130   \newhookcommand@series{Xendbeforenumber}%
6131   \newhookcommand@series{Xendafternumber}%
6132
6133   \newhooktoggle@series{Xendparagraph}
6134   \newhookcommand@series{Xendafternote}
6135   \newhookcommand@series{Xendsep}

```

```

6136 \newhookcommand@series{Xendinplaceofnumber}%
6137 \newhooktoggle@series{Xendnonumber}%
6138
6139
6140 \newhooktoggle@series{Xendnumberonlyfirstinline}%
6141 \newhooktoggle@series{Xendnumberonlyfirstintwolines}%
6142
6143 \newhookcommand@series{Xendsymlinenum}%
6144 \newhookcommand@series{Xendbeforesymlinenum}%
6145 \newhookcommand@series{Xendaftersymlinenum}%
6146 \newhookcommand@series{Xendboxsymlinenum}%
6147
6148 \newhookcommand@series{Xendbhooklinenum}%
6149 \newhookcommand@series{Xendahooklinenum}%
6150 \newhookcommand@series{Xendbhookinplaceofnumber}%
6151 \newhookcommand@series{Xendahookinplaceofnumber}%
6152
6153 \newhookcommand@series{Xendhangindent}%
6154
6155 \newhooktoggle@series{Xendpagenumberonlyfirst}%
6156 \newhooktoggle@series{Xendpagenumberonlyfirstifsingle}%
6157 \newhooktoggle@series{Xendpagenumberonlyfirstintwo}%
6158 \newhookcommand@series{Xendsympagenum}%
6159 \newhookcommand@series{Xendinplaceofpagenumber}%
6160
6161 \newhookcommand@series{Xendtxtbeforenotes}%
6162
6163 \fi
6164 \newhooktoggle@series{Xendlineflag}[apprefwithpage,SErefwithpage]
6165 \newhookcommand@series{Xendt看olines}[apprefwithpage,SErefwithpage]
6166 \newhookcommand@series{Xendmoreethant看olines}[apprefwithpage,SErefwithpage]
6167 \newhooktoggle@series{Xendtwolinesbutnotmore}[apprefwithpage,SErefwithpage]
6168 \newhooktoggle@series{Xendtwolinesonlyinsamepage}[apprefwithpage,
SErefwithpage]
6169 \newhookcommand@series{Xendlinerangeseparator}[apprefwithpage,SErefwithpage
]
6170 \newhookcommand@series{Xendbeforepagenumber}[apprefwithpage,SErefwithpage,
SErefonlypage]
6171 \newhookcommand@series{Xendafterpagenumber}[apprefwithpage,SErefwithpage]
6172 \newhookcommand@series{Xendlineprefixsingle}[apprefwithpage,SErefwithpage]
6173 \newhookcommand@series{Xendlineprefixmore}[apprefwithpage,SErefwithpage]
6174 \newhookcommand@series{Xendsublinesep}[apprefwithpage,SErefwithpage]
6175
6176 \newhookcommand@series{Xendlinenumannotationposition}[apprefwithpage,
SErefwithpage]%
6177 \newhookcommand@series{Xendwraplinenumannotation}[apprefwithpage,
SErefwithpage]%
6178 %

```

## XXII.4 Hooks for a particular footnote

`\newhooktoggle@specific` `\newhooktoggle@specific` is a generic command to create boolean hook specific to a note.

```

6179 \newcommand{\newhooktoggle@specific}[1]{%
6180   \newtoggle{#1}%
6181   \define@key[mac]{truefootnoteoption}{#1}[]{\global\settoggle{#1}{true}}%
When enabling footnote option
6182   \define@key[mac]{falsefootnoteoption}{#1}[]{\global\settoggle{#1}{false}
}}
6183 }
6184 %

```

`\newhookarg@specific` `\newhookarg@specific` is a generic command to create argumen hook specific to a note.

```

6185 \newcommand{\newhookarg@specific}[1]{%
6186   \define@key[mac]{truefootnoteoption}{#1}{\csgdef{#1@}{##1}}%When enabling
footnote option
6187   \define@key[mac]{falsefootnoteoption}{#1}{\global\csundef{#1@}}%When
disabling footnote option
6188 }
6189 %

```

And now, we define some hooks specific to a note.

```

6190 \newhooktoggle@specific{fulllines}%
6191 \newhooktoggle@specific{nonum}
6192 \newhooktoggle@specific{nosep}
6193 \newhookarg@specific{linrangesep}
6194 %

```

`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.

```

6195 %

```

`\nomk@` `\nomk@` toggle is used by `reledpar` to remove the footnote mark in the text when using `\footnoteXmk`. Read `reledpar` handbook.

```

6196 \newtoggle{nomk}%
6197 %

```

## XXII.5 Alias

`\Xnolemmaseparator` `\Xnolemmaseparator[⟨series⟩]` is just an alias for `\Xlemmaseparator[⟨series⟩]{}`.

```

6198 \newcommandx*{\Xnolemmaseparator}[1][1]{\Xlemmaseparator[#1]}
6199 %

```

## 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 the  $\TeX$ 's kernel. Since `reledmac`, we use `etoolbox`'s patching commands instead of overriding. It should provides better compatibility with other package which modify these commands

`\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.

```

6200 \newcommand*{\l@ddoxtrafeet}{%
6201   \IfStrEq{familiar-critical}{\@fnpos}
6202     {\do@feetX\do@Xfeet}%
6203   {%
6204     \IfStrEq{critical-familiar}{\@fnpos}%
6205       {\do@Xfeet\do@feetX}%
6206     {%
6207       \setbox\@outputbox \vbox{%
6208         \unvbox\@outputbox%
6209         \do@feet@custom@order}{\@fnpos}%
6210       }%
6211     }%
6212   }%
6213 }%
6214
6215 %

```

`\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.

```

6216 \newcommand{\do@feet@custom@order}[2]{%
6217   \def\do##1{%
6218     \edef\@notesseries{\@firstoftwo##1}%
6219     \edef\@notetype{\@secondoftwo##1}%
6220     \ifdefstring{\@notetype}{critical}%
6221       {\csuse{#1append@Xnotes}{\@notesseries}}%
6222     {\ifdefstring{\@notetype}{familiar}%
6223       {\csuse{#1append@notesX}{\@notesseries}}%
6224     }%
6225   }%
6226 }%
6227 \expandafter\docsvlist\expandafter{#2}%
6228 }%

```

```
6229 %
```

**\do@Xfeet** \do@Xfeet is the code extending \@makecol to cater to the extra critical feet.

```
6230 \newcommand*\do@Xfeet}{%
6231   \setbox\@outputbox \vbox{%
6232     \unvbox\@outputbox
6233     \op@Xfeet}}
6234 %
```

**\@opXfeet** The extra critical feet to be added to the output. . A macro which appends critical notes to the output's routine, also adding vertical space before notes

```
\append@Xnotes
\print@Xnotes
6235 \newcommand{\append@Xnotes}[1]{%
6236   \ifvoid\csuse{#1footins}\else%
6237     \global\skip\csuse{#1footins}=\csuse{Xbeforenotes@#1}%
6238     \global\advance\skip\csuse{#1footins} by\csuse{Xafterrule@#1}%
6239     \print@Xnotes{#1}%
6240   \fi%
6241 }%
6242 %
```

The normal way to add one series, \print@Xnotes, is replaced by reledpar when using \Pages.

```
6243 \newcommand\print@Xnotes[1]{%
6244   \xdef\@currentseries{#1}%
6245   \csuse{#1footstart}{#1}%
6246   \csuse{#1footgroup}{#1}%
6247 }%
6248 %
```

We print all series of notes by looping on them. We check before printing them that they are not voided.

```
6249 \newcommand*\@opXfeet}{%
6250   \unless\ifnocritical@%
6251     \gdef\firstXseries@{}%
6252     \def\do##1{%
6253       \append@Xnotes{##1}%
6254     }%
6255     \dolistloop{\@series}%
6256   \fi%
6257 }%
6258 %
```

**\l@ddodoreinextrafeet** \l@ddodoreinextrafeet is the code for catering for the extra footnotes within \@reinserts. We use the same category and ordering as in \l@ddoxtrafeet.

```
6259 \newcommand*\l@ddodoreinextrafeet}{%
6260   \IfStrEq{familiar-critical}{\@fnpos}
```

```

6261 {\@doreinfeetX\X@doreinfeet}%
6262 {%
6263 \IfStrEq{critical-familiar}{\@fnpos}%
6264   {\X@doreinfeet\@doreinfeetX}%
6265   {\@doreinfeetX\X@doreinfeet}%
6266 }%
6267 }
6268
6269 %

```

**\X@doreinfeet** \X@doreinfeet is the code for catering for the extra critical footnotes within \@reinserts.

```

6270 \newcommand*{\X@doreinfeet}{%
6271   \unless\ifnocritical@%
6272     \def\do##1{%
6273       \ifvoid\csuse{##1footins}\else%
6274         \insert\csuse{##1footins}{\unvbox\csuse{##1footins}}%
6275       \fi}%
6276   \dolistloop{\@series}
6277   \fi%
6278 }
6279
6280 %

```

**\print@notesX** We have to add all the new kinds of familiar footnotes to the output routine. A macro  
**\append@notesX** which appends the familiar footnotes of one series onto the output routine, also adding  
**\do@feetX** vertical skip before notes.

```

6281 \@doreinfeetX \newcommand{\append@notesX}[1]{%
6282   \ifvoid\csuse{footins#1}\else%
6283     \global\skip\csuse{footins#1}=\csuse{beforenotesX@#1}%
6284     \global\advance\skip\csuse{footins#1} by\csuse{afterruleX@#1}%
6285     \print@notesX{#1}%
6286   \fi%
6287 }%
6288 %

```

The normal way to print one series of notes. \print@Xnotes is replaced by reledpar when using \Pages.

```

6289 \newcommand\print@notesX[1]{%
6290   \xdef\@currentseries{#1}%
6291   \csuse{footstart#1}{#1}%
6292   \csuse{footgroup#1}{#1}%
6293 }%
6294 %

```

We print all the series of notes by looping on them. We check before printing them that they are not voided.

```

6295 \newcommand*{\do@feetX}{%

```

```

6296 \unless\ifnofamiliar@%
6297 \gdef\firstseriesX@{}%
6298 \setbox\@outputbox \vbox{%
6299 \unvbox\@outputbox%
6300 \def\do##1{%
6301 \append@notesX{##1}%
6302 }%
6303 \dolistloop{\@series}}%
6304 \fi%
6305 }%
6306
6307 \newcommand{\@doreinfeetX}{%
6308 \unless\ifnofamiliar@%
6309 \def\do##1{%
6310 \ifvoid\csuse{footins##1}\else
6311 \insert%
6312 \csuse{footins##1}
6313 {\unvbox\csuse{footins##1}}%
6314 \fi%
6315 }%
6316 \dolistloop{\@series}%
6317 \fi%
6318 }%
6319
6320 %

```

## 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.)

```

6321 \@ifclassloaded{memoir}{%
6322 %
6323
6324 memoir is loaded so we use memoir's built in hooks.
6325
6326 \g@addto@macro{\m@mdoextrafeet}{\l@ddoxtrafeet}%
6327 \g@addto@macro{\m@mdodoreinextrafeet}{\l@ddodoreinextrafeet}%
6328 }{%
6329 %

```

memoir has not been loaded, so patch `\@makecol` and `\@reinserts`. If the fancyhdr package < version 3.8 has been loaded, we patch the `\latex@makecol` command, because this package redefines the standard `\@makecol` in the preamble, to call `\latex@makecol` which have been `\let` to `\@makecol`. If this package is not loaded, we directly patch `\@makecol`. If the fancyhdr package  $\geq$  version 3.8, we also directly patch `\@makecol`, because fancyhdr does its own patch `\AtBeginDocument`.

```

6327 \ifbool{expr}%
6328   test{\@ifpackageloaded{fancyhdr}}%
6329   and test {\@ifdef{\latex@makecol}}%
6330 }{%
6331   \patchcmd%
6332     {\latex@makecol}%
6333     {\xdef\@freelist{\@freelist\@midlist}}%
6334     {\xdef\@freelist{\@freelist\@midlist}\l@ddoxtrafeet}%
6335     {}%
6336     {\led@error@fail@patch@@@makecol}%
6337 }{%
6338   \patchcmd%
6339     {\@makecol}%
6340     {\xdef\@freelist{\@freelist\@midlist}}%
6341     {\xdef\@freelist{\@freelist\@midlist}\l@ddoxtrafeet}%
6342     {}%
6343     {\led@error@fail@patch@@@makecol}%
6344 }%
6345
6346 \patchcmd%
6347   {\@reinserts}%
6348   {\ifvbox}%
6349   {\l@ddodoreinextrafeet\ifvbox}%
6350   {}%
6351   {\led@error@fail@patch@@@reinserts}%
6352 }
6353
6354 %

```

It turns out that `\@doclearpage` also needs modifying.

`\if@led@nofoot` We have to check if there are any leftover feet.

```

6355 \newif\if@led@nofoot
6356
6357 %

```

```

6358 \@ifclassloaded{memoir}{%
6359 %

```

If the memoir class is loaded we hook into its modified `\@doclearpage`.

```

\@mem@extranofeet \g@addto@macro{\@mem@extranofeet}{%%
6361   \def\do#1{%
6362     \unless\ifnocritical@%
6363       \ifvoid\csuse{#1footins}\else\@mem@nofootfalse\fi%
6364     \fi%
6365     \unless\ifnofamiliar@%
6366       \ifvoid\csuse{footins#1}\else\@mem@nofootfalse\fi%
6367     \fi%

```



```

6368 }
6369 \dolistloop{\@series}%
6370 }%
6371 }-%
6372 %

```

As memoir is not loaded we have patch \@doclearpage.

```

\@led@testifnofoot73 \newcommand*{\@led@testifnofoot}{%
\@doclearpage74 \@led@nofoottrue%
6375 \ifvoid\footins\else%
6376 \@led@nofootfalse%
6377 \fi%
6378 \def\do##1{%
6379 \unless\ifnocritical@%
6380 \ifvoid\csuse{##1footins}\else%
6381 \@led@nofootfalse%
6382 \fi%
6383 \fi%
6384 \unless\ifnofamiliar@%
6385 \ifvoid\csuse{footins##1}\else%
6386 \@led@nofootfalse%
6387 \fi%
6388 \fi%
6389 }%
6390 \dolistloop{\@series}%
6391 }-%
6392
6393 \pretocmd%
6394 {\@doclearpage}%
6395 {\@led@testifnofoot}%
6396 {}%
6397 {\led@error@fail@patch@@doclearpage}%
6398
6399 \patchcmd%
6400 {\@doclearpage}%
6401 {\ifvoid\footins}%
6402 {\if@led@nofoot}%
6403 {}%
6404 {\led@error@fail@patch@@doclearpage}%
6405
6406 }
6407
6408 %

```

## 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. 127). 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  $\text{\LaTeX}$  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  $\text{\LaTeX}$ 's `\@outputpage` macro, as is the `page` counter in this macro. However, this increase will take account of the options.

`\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  $\text{\LaTeX}$  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.

```

6409
6410 \newcommand{\par@patch@thepage}{%
6411   \ifboolexpr{%
6412     bool{sameparallelpagenumbers}%
6413     or bool{prevpgnotnumbered}%
6414   }%
6415   {%
6416     \patchcmd{\thepage}%
6417       {page}{par@page}%
6418       {}%
6419       {\led@error@fail@patch@thepage}%
6420   }{}%
6421 }%
6422
6423 \newcommand{\par@patch@pagenumbering}{%
6424   \setcounter{par@page}{1}%
6425   \par@patch@thepage%
6426 }%
6427
6428 \ifl@dmemoir%
6429   \apptocmd{\@mempnum}%
6430     {\par@patch@pagenumbering}%
6431     {}%

```

```

6432     {\led@error@fail@patch@memnum}%
6433
6434 \else%
6435   \apptocmd{\pagenumbering}%
6436     {\par@patch@pagenumbering}%
6437     {}%
6438     {\led@error@fail@patch@pagenumbering}%
6439 \fi%
6440
6441 \AtBeginDocument{\par@patch@thepage}%
6442 %

```

**\@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.

```

6443 \AtBeginDocument{%
6444   \apptocmd{\@outputpage}{%
6445     \ifsameparallelpagelapnumber%
6446       \ifl@dprintingpages%
6447         \ifodd\c@page\else%
6448           \stepcounter{par@page}%
6449         \fi%
6450       \else%
6451         \stepcounter{par@page}%
6452       \fi%
6453     \else%
6454       \stepcounter{par@page}%
6455     \fi%
6456   }%
6457   {}%
6458   {\led@error@fail@patch@@@outputpage}%
6459 }%
6460 %

```

**\thepar@page** The par@page counter.

```

6461 \newcounter{par@page}%
6462 \setcounter{par@page}{1}%
6463 %

```

## 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.

```
6464 \list@create{\labelref@list}
6465 %
```

**\zz@@@** Two convenience macros to zero three / four labeling counters in one go.

```
\zz@@@
6466 \newcommand*\zz@@@{-}{000|000|000}% Set three counters to zero in one go
6467 \newcommand*\zz@@@{-}{000|000|000|000}% Set four counters to zero in one go
6468 %
6469 %
```

**\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.

```
6470 \newcommand*\edlabel}[1]{%
6471   \leavevmode%
6472   \@bsphack%
6473   \ifboolexpr{bool{ledRcol} or bool{ledRcol@}}{%
6474     \ifXnote@%
6475     \protected@write\@auxout{%
6476       {\string\l@dmake@labelsR\space\thepage|\l@dparsedstartline|\l@dparsedstartsub||\annot@start|\the\c@pstartR|{#1}}%
6477       \ifdef{\hypertarget}%
6478       {\Hy@raisedlink{\hypertarget{#1}{}}}%
6479       {}%
6480     \else%
6481     \write\linenum@outR{\string\@lab}%
6482     \ifx\labelref@listR\empty%
6483     \xdef\label@refs{\zz@@@}%
6484     \else%
6485     \gl@p\labelref@listR\to\label@refs%
6486     \fi%
6487   }
```

<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.

```

6487 \ifvmode%
6488 \advancelabel@refs%
6489 \fi%
6490 %

```

Use code from the kernel `\label` command to write the correct page number. Also define an `hypertarget` if `hyperref` package is loaded.

```

6491 \protected@write\@auxout{}%
6492 {\string\l@dmake@labelsR\space\thepage|\label@refs|\the\c@pstartR
|{#1}}%
6493 \ifdef\hypertarget%
6494 {\Hy@raisedlink{\hypertarget{#1}{}}}%
6495 {}%
6496 \fi%
6497 }{%
6498 \ifXnote@%
6499 \ifl@dpairing%pstart or pstartL?
6500 \protected@write\@auxout{}%
6501 {\string\l@dmake@labels\space\thepage|\l@dparsedstartline|\
l@dparsedstartsub||\annot@start|\the\c@pstartL|{#1}}%
6502 \ifdef\hypertarget%
6503 {\Hy@raisedlink{\hypertarget{#1}{}}}%
6504 {}%
6505 \else%
6506 \protected@write\@auxout{}%
6507 {\string\l@dmake@labels\space\thepage|\l@dparsedstartline|\
l@dparsedstartsub||\annot@start|\the\c@pstart|{#1}}%
6508 \ifdef\hypertarget%
6509 {\Hy@raisedlink{\hypertarget{#1}{}}}%
6510 {}%
6511 \fi%
6512 \else%
6513 \write\linenum@out{\string\@lab}%
6514 \ifx\labelref@list\empty%
6515 \xdef\label@refs{\zz@@@}%
6516 \else%
6517 \gl@p\labelref@list\to\label@refs%
6518 \fi%
6519 \ifvmode%
6520 \advancelabel@refs%
6521 \fi%
6522 \ifl@dpairing%Pstart or PstartL?
6523 \protected@write\@auxout{}%
6524 {\string\l@dmake@labels\space\thepage|\label@refs|\the\c@pstartL
|{#1}}%
6525 \ifdef\hypertarget%
6526 {\Hy@raisedlink{\hypertarget{#1}{}}}%
6527 {}%
6528 \else%
6529 \protected@write\@auxout{}%

```

```

6530      {\string\l@dmake@labels\space\thepage|\label@refs|\the\c@pstart
|{#1}}}%
6531      \ifdef{\hypertarget}%
6532      {\Hy@raisedlink{\hypertarget{#1}{}}}%
6533      {}%
6534      \fi%
6535      \fi%
6536      }%
6537      \esphack}%
6538
6539      %

```

`\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 using `\advancelabel@refs` command.

```

6540 \newcounter{line}%
6541 \newcounter{subline}%
6542 \newcounter{absline}%
6543 \newcommand{\advancelabel@refs}{%
6544     \setcounter{line}{\expandafter\labelrefsparseline\label@refs}%
6545     \stepcounter{line}%
6546     \setcounter{absline}{\expandafter\labelrefsparseabsline\label@refs}%
6547     \stepcounter{absline}%
6548     \ifsublines@%
6549         \setcounter{subline}{\expandafter\labelrefsparsesubline\label@refs}%
6550         \stepcounter{subline}{1}%
6551         \def\label@refs{\theline|\thesubline|\theabsline}%
6552     \else%
6553         \def\label@refs{\theline|0|\theabsline}%
6554     \fi%
6555 }
6556 \def\labelrefsparseline#1|#2|#3{#1}%
6557 \def\labelrefsparsesubline#1|#2|#3{#2}%
6558 \def\labelrefsparseabsline#1|#2|#3{#3}%
6559 %

```

`\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.

```

6560 \newcommand*{\l@dmake@labels}{%
6561 \def\l@dmake@labels#1|#2|#3|#4|#5|#6|#7{%
6562 \expandafter\ifx\csname the@label\csuse{XR@prefix}#7\endcsname%
6563 \relax%
6564 \else%
6565 \led@warn@DuplicateLabel{\csuse{XR@prefix}#7}%
6566 \fi
6567 \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
6568 \csgdef{the@label\csuse{XR@prefix}#7}{#1|#2|#3|#4|#5|#6|\relax}%
6569 \ignorespaces}
6570
6571 %

```

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.

```

6572 \AtBeginDocument{%
6573 \def\l@dmake@labels#1|#2|#3|#4|#5|#6|#7{%
6574 }
6575
6576 %

```

**\@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.

```

6577
6578 \newcommand*{\@lab}{%
6579 \ifledRcol
6580 \xright@appenditem{\linenumr@p{\line@numR}}|%
6581 \ifsublines@ \sublinenumr@p{\subline@numR}\else 0\fi|%
6582 \the\absline@numR|%
6583 \current@annot%
6584 }%
6585 \to\labelref@listR
6586 \else
6587 \xright@appenditem{%
6588 \linenumr@p{\line@num}|%
6589 \ifsublines@ \sublinenumr@p{\subline@num}\else 0\fi|%
6590 \the\absline@num|%
6591 \current@annot%
6592 }%

```

```

6593 \to\labelref@list
6594 \fi}
6595 %

```

**\applabel** \applabel, if called in \edtext will insert automatically both a starting and an ending label for the current edtext lines.

```

6596 \newcommand*{\applabel}[1]{%
6597 \if@edtext@secondarg%
6598 %

```

Label should not be already defined.

```

6599 \ifcsundef{the@label#1}{%
6600 \csdef{the@label#1}{\applabel}%
6601 }%
6602 {%
6603 \led@warn@DuplicateLabel{#1 (\applabel)}%
6604 }%
6605 %

```

Parse the \edtext line numbers and annotations.

```

6606 \expandafter\l@dp@rsefootsspec\l@d@nums|%
6607 \expandafter\parse@annot\l@current@annot|%
6608 %

```

Use the  $\LaTeX$  standard hack for label.

```

6609 \@bsphack%
6610 %

```

And now, write the data in the auxiliary file.

```

6611 \ifl@edRcol%
6612 \protected@write\@auxout{%
6613 {\string\l@dmake@labelsR\space\l@dparsedstartpage|\l@dparsedstartline|\l@dparsedstartsub||\annot@start|\the\c@pstartR|{#1:start}}}%
6614 \ifdef{\hypertarget}%
6615 {\Hy@raisedlink{\hypertarget{#1:start}}}%
6616 {}%
6617 \protected@write\@auxout{%
6618 {\string\l@dmake@labelsR\space\l@dparsedendpage|\l@dparsedendline||\l@dparsedendsub||\annot@end|\the\c@pstartR|{#1:end}}}%
6619 \else%
6620 \ifl@dpairing%pstart or pstartL?
6621 \protected@write\@auxout{%
6622 {\string\l@dmake@labelsR\space\l@dparsedstartpage|\l@dparsedstartline|\l@dparsedstartsub||\annot@start|\the\c@pstartL|{#1:start}}}%
6623 \ifdef{\hypertarget}%
6624 {\Hy@raisedlink{\hypertarget{#1:start}}}%

```



```

6625     {}%
6626     \protected@write\@auxout{}%
6627     {\string\l@dmake@labels\space\l@dparsedendpage|\
l@dparsedendline|\l@dparsedendsub||\annot@end|\the\c@pstartL|{#1:end}}}%
6628     \else%
6629     \protected@write\@auxout{}%
6630     {\string\l@dmake@labels\space\l@dparsedstartpage|\
l@dparsedstartline|\l@dparsedstartsub||\annot@start|\the\c@pstart|{#1:start
}}}%
6631     \ifdef{\hypertarget}%
6632     {\Hy@raisedlink{\hypertarget{#1:start}}}%
6633     {}%
6634     \protected@write\@auxout{}%
6635     {\string\l@dmake@labels\space\l@dparsedendpage|\
l@dparsedendline|\l@dparsedendsub||\annot@end|\the\c@pstart|{#1:end}}}%
6636     \fi%
6637     \fi%
6638     %

```

Use the  $\TeX$  standard hack for label.

```

6639     \esphack%
6640     %

```

Warning if `\applabel` is called outside of `\edtext`.

```

6641     \else%
6642     \led@warn@AppLabelOutSecondArgEdtext{#1}%
6643     \fi%
6644     %

```

End of `\applabel`

```

6645 }%
6646 %

```

**`\edlabels`** `\edlabels` and `\edlabelE` are just used to mark the beginning and the end of a passage.

```

\edlabelE
\edlabelSE
6647 \newcommand{\edlabels}[1]{%
6648   \edlabel{#1:start}%
6649 }
6650 \newcommand{\edlabelE}[1]{%
6651   \edlabel{#1:end}%
6652 }
6653 \newcommand{\edlabelSE}[1]{%
6654   \edlabels{#1}%
6655   \edlabelE{#1}%
6656 }
6657 %

```

**`\wrap@edcrossref`** `\wrap@edcrossref` is called around all `reledmac` crossref commands, except those which start with `x`. It adds the hyperlink.

```

6658 \newrobustcmd{\wrap@edcrossref}[2]{%
6659   \ifdef{\hyperlink}%
6660     {\hyperlink{#1}{#2}}%
6661     {#2}%
6662 }
6663 %

```

**\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.

```

6664 \newcommand*\edpageref[1]{\l@dref@undefined{#1}\wrap@edcrossref{#1}{\l@dgetref@num{1}{#1}}}
6665 \newcommand*\xpageref[1]{\l@dgetref@num{1}{#1}}
6666
6667 %

```

**\edlineref** If the specified label exists, \lineref gives its line number.

**\xlineref**

```

6668 \newcommand*\edlineref[1]{%
6669   \l@dref@undefined{#1}%
6670   \wrap@edcrossref{#1}{%
6671     \providetoggle{label@#1@ledRcol}%Required for the first run, when the
label has not yet been parsed on the .aux file
6672     \iftoggle{label@#1@ledRcol}%
6673       {\linenumrepR{\l@dgetref@num{2}{#1}}}%
6674       {\linenumrep{\l@dgetref@num{2}{#1}}}%
6675     \xflagref{#1}%
6676   }%
6677 }%
6678 \newcommand*\xlineref[1]{\l@dgetref@num{2}{#1}}%
6679
6680 %

```

**\sublineref** If the specified label exists, \sublineref gives its sub-line number.

**\xsublineref**

```

6681 \newcommand*\sublineref[1]{%
6682   \l@dref@undefined{#1}%
6683   \wrap@edcrossref{#1}{%
6684     \providetoggle{label@#1@ledRcol}%Required for the first run, when the
label has not yet been parsed on the .aux file
6685     \iftoggle{label@#1@ledRcol}%
6686       {\sublinenumrepR{\l@dgetref@num{3}{#1}}}%
6687       {\sublinenumrep{\l@dgetref@num{3}{#1}}}%

```

```

6688 }%
6689 }%
6690 \newcommand*{\xsublineref}[1]{\l@getref@num{3}{#1}}
6691
6692 %

```

**\xabslineref** If the specified label exists, \xabslineref gives its absolute line number. This is generally used only by reledmac internal macros.

```

6693 \newcommand*{\xabslineref}[1]{\l@getref@num{4}{#1}}%
6694 %

```

**\annotationref** If the specified label exists, \edannotationref gives the line number annotation with which it is associated.

```

6695 \newcommand*{\annotationref}[1]{%
6696   \l@dref@undefined{#1}%
6697   \wrap@edcrossref{#1}{%
6698     \l@wrap@ifnotemptybox{\Xwraplinenumannotation@ref}{%
6699       \l@getref@num{5}{#1}%
6700     }%
6701   }%
6702 }%
6703 \newcommand*{\xannotationref}[1]{%
6704   \l@getref@num{5}{#1}%
6705 }%
6706 %

```

**\pstartref** If the specified label exists, \pstartref gives its pstart number.

```

6707 \newcommand*{\pstartref}[1]{\l@dref@undefined{#1}\wrap@edcrossref{#1}{\l@getref@num{6}{#1}}}%
6708 \newcommand*{\xpstartref}[1]{\l@getref@num{6}{#1}}%
6709
6710 %

```

**\xflagref** \xflagref finds the side flag of any ref defined with \edlabel.

```

6711 \newcommand*{\xflagref}[1]{\l@getref@num{7}{#1}}%
6712 %

```

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.

```

6713 \newcommand*{\l@dref@undefined}[1]{%
6714   \expandafter\ifx\csname the@label#1\endcsname\relax
6715     \led@warn@RefUndefined{#1}%
6716   \fi}
6717
6718 %

```

**\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.

```

6719 \newcommand*{\l@dgetref@num}[2]{%
6720   \expandafter
6721   \ifx\csname the@label#2\endcsname \relax
6722     000%
6723   \else
6724     \expandafter\expandafter\expandafter
6725     \l@dlabel@parse\csname the@label#2\endcsname|#1%
6726   \fi}
6727
6728 %

```

**\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`.)

```

6729 \newcommand*{\l@dlabel@parse}{%
6730   \def\l@dlabel@parse#1|#2|#3|#4|#5|#6|#7|#8{%
6731     \ifcase #8%
6732       \or #1%
6733       \or #2%
6734       \or #3%
6735       \or #4%
6736       \or #5%
6737       \or #6%
6738       \or #7%
6739     \fi}
6740 %

```

**\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.

```

6741 \newcommand*{\xxref}[2]{%
6742   {%
6743     \expandafter\ifx\csname the@label#1\endcsname\relax%
6744       \let\@tempa\zz@@@%
6745       \def\@tempc{}%
6746     \else%
6747       \def\@tempa{%
6748         \l@ldgetref@num{1}{#1}|%
6749         \l@ldgetref@num{2}{#1}|%
6750         \l@ldgetref@num{3}{#1}%
6751       }%
6752       \edef\@tempc{%
6753         \l@ldgetref@num{5}{#1}%
6754       }%
6755     \fi%
6756     \expandafter\ifx\csname the@label#2\endcsname\relax%
6757       \let\@tempb\zz@@@%
6758       \def\@tempd{}%
6759     \else%
6760       \def\@tempb{%
6761         \l@ldgetref@num{1}{#2}|%
6762         \l@ldgetref@num{2}{#2}|%
6763         \l@ldgetref@num{3}{#2}%
6764       }%
6765       \edef\@tempd{%
6766         \l@ldgetref@num{5}{#2}%
6767       }%
6768     \fi%
6769     \global\appto\@beforeinsertofthisedtext{%
6770       \def\@this@crossref@start{#1}%
6771     }%
6772     \global\appto\@beforeinsertofthisedtext{%
6773       \def\@this@crossref@end{#2}%
6774     }%
6775     \linenum{\@tempa|\@tempb}%
6776     \edef\@tempe{\@tempc|\@tempd}%
6777     \expandafter\lineannot\expandafter{\@tempe}%
6778   }%
6779 }%
6780
6781
6782 %

```

`\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`.

```

6783 \def\Xtwolines@appref{}%
6784 \def\Xtwolines@SEref{}%
6785
6786 \def\Xmorethantwolines@appref{}%
6787 \def\Xmorethantwolines@SEref{}%
6788
6789 \def\Xlinerangeseparator@appref{\endashchar}%
6790 \def\Xlinerangeseparator@SEref{\endashchar}%
6791
6792 \def\Xsublinesep@appref{\fullstop}%
6793 \def\Xsublinesep@SEref{\fullstop}%
6794
6795 \def\Xpagelinesep@appref{\fullstop}%
6796 \def\Xpagelinesep@SEref{\fullstop}%
6797
6798
6799 \newtoggle{Xtwolinesbutnotmore@appref}%
6800 \newtoggle{Xtwolinesbutnotmore@SEref}%
6801
6802 \newtoggle{Xtwolinesonlyinsamepage@appref}%
6803
6804 \newtoggle{Xtwolinesonlyinsamepage@SEref}%
6805
6806 \newtoggle{Xlineflag@appref}%
6807 \toggletrue{Xlineflag@appref}%Here exception
6808 \newtoggle{Xlineflag@SEref}%
6809 \toggletrue{Xlineflag@SEref}%Here exception
6810
6811 \def\Xlinenumannotationposition@appref{after}%
6812 \def\Xlinenumannotationposition@SEref{after}%
6813
6814 \def\Xwraplinenumannotation@appref{\textsuperscript}%
6815 \def\Xwraplinenumannotation@SEref{\textsuperscript}%
6816
6817 \def\Xendtwolines@apprefwithpage{}%
6818 \def\Xendtwolines@SErefwithpage{}%
6819
6820 \def\Xendmorethantwolines@apprefwithpage{}%
6821 \def\Xendmorethantwolines@SErefwithpage{}%
6822
6823 \def\Xendlinerangeseparator@apprefwithpage{\endashchar}
6824 \def\Xendlinerangeseparator@SErefwithpage{\endashchar}
6825 \def\Xendlinerangeseparator@SErefonlypage{\endashchar}

```

```

6826 \def\Xendbeforepagenumber@apprefwithpage{p.}%
6827 \def\Xendbeforepagenumber@SErefwithpage{p.}%
6828 \def\Xendbeforepagenumber@SEonlypage{p.}%
6829
6830
6831 \def\Xendafterpagenumber@apprefwithpage{} }%
6832 \def\Xendafterpagenumber@SErefwithpage{} }%
6833
6834
6835 \def\Xendlineprefixsingle@apprefwithpage{}%
6836 \def\Xendlineprefixsingle@SErefwithpage{}%
6837
6838 \def\Xendlineprefixmore@apprefwithpage{}%
6839 \def\Xendlineprefixmore@SErefwithpage{}%
6840
6841 \newtoggle{Xendtwolinesbutnotmore@apprefwithpage}%
6842 \newtoggle{Xendtwolinesbutnotmore@SErefwithpage}%
6843
6844 \def\Xendsublinesep@apprefwithpage{\fullstop}%
6845 \def\Xendsublinesep@SErefwithpage{\fullstop}%
6846
6847 \newtoggle{Xendtwolinesonlyinsamepage@apprefwithpage}%
6848 \newtoggle{Xendtwolinesonlyinsamepage@SErefwithpage}%
6849
6850 \newtoggle{Xendlineflag@apprefwithpage}
6851 \toggletrue{Xendlineflag@apprefwithpage}%Here, exception
6852 \newtoggle{Xendlineflag@SErefwithpage}
6853 \toggletrue{Xendlineflag@SErefwithpage}%Here, exception
6854
6855 \def\Xendlinenumannotationposition@apprefwithpage{after}%
6856 \def\Xendlinenumannotationposition@apprefwithpage@SErefwithpage{after}%
6857
6858 \def\Xendwraplinenumannotation@apprefwithpage{\textsuperscript}%
6859 \def\Xendwraplinenumannotation@SErefwithpage{\textsuperscript}%
6860 %

```

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.

```

6861
6862 \gdef\Xboxstartlinenum@appref{Opt}
6863 \gdef\Xboxstartlinenum@SEref{Opt}
6864
6865 \gdef\Xboxendlinenum@appref{Opt}
6866 \gdef\Xboxendlinenum@SEref{Opt}
6867
6868 \gdef\Xendboxstartlinenum@apprefwithpage{Opt}
6869 \gdef\Xendboxstartlinenum@SErefwithpage{Opt}
6870
6871 \gdef\Xendboxendlinenum@apprefwithpage{Opt}

```

```

6872 \gdef\Xendboxendlinenum@SErefwithpage{Opt}
6873
6874 \newtoggle{Xendpagenumberonlyfirst@apprefwithpage}
6875 \newtoggle{Xendpagenumberonlyfirst@SErefwithpage}
6876
6877 \newtoggle{Xendpagenumberonlyfirstifsingle@apprefwithpage}
6878 \newtoggle{Xendpagenumberonlyfirstifsingle@SErefwithpage}
6879
6880 \newtoggle{Xendpagenumberonlyfirstintwo@apprefwithpage}
6881 \newtoggle{Xendpagenumberonlyfirstintwo@SErefwithpage}
6882
6883 \gdef\Xendsympagenum@apprefwithpage{}
6884 \gdef\Xendsympagenum@SErefwithpage{}
6885
6886 \gdef\Xendinplaceofpagenumber@apprefwithpage{}
6887 \gdef\Xendinplaceofpagenumber@SErefwithpage{}
6888
6889 %

```

Now, declare the default values of \@apprefprefixsingle and \@apprefprefixmore, \@SErefprefix, \@SErefprefixmore and the commands which defines them.

```

6890 \newcommand\@apprefprefixsingle{}%
6891 \newcommand\@SErefprefixsingle{}%
6892
6893 \newcommand\@apprefprefixmore{}%
6894 \newcommand\@SErefprefixmore{}%
6895
6896 \newcommand{\setapprefprefixsingle}[1]{%
6897   \gdef\@apprefprefixsingle{#1}%
6898 }
6899 \newcommand{\setSErefprefixsingle}[1]{%
6900   \gdef\@SErefprefixsingle{#1}%
6901 }
6902
6903 \newcommand{\setapprefprefixmore}[1]{%
6904   \gdef\@apprefprefixmore{#1}%
6905 }
6906 \newcommand{\setSErefprefixmore}[1]{%
6907   \gdef\@SErefprefixmore{#1}%
6908 }
6909
6910 %

```

And not \setSErefonlypageprefixsingle and \setSErefonlypageprefixmore.

```

6911 \newcommand{\setSErefonlypageprefixsingle}[1]{%
6912   \gdef\SErefonlypage@prefixsingle{#1}%
6913 }%
6914 \newcommand{\setSErefonlypageprefixmore}[1]{%
6915   \gdef\SErefonlypage@prefixmore{#1}%
6916 }%

```



6917 %

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.

```

6918
6919 \newcommandx{\appref}[2][1,usedefault]{\reformatted@{#1}{#2}{appref}}
6920 \newcommandx{\Seref}[2][1,usedefault]{\reformatted@{#1}{#2}{Seref}}
6921
6922 \newcommandx{\apprefwithpage}[2][1,usedefault]{\reformattedwithpage@
6923 {#1}{#2}{appref}}
6924 \newcommandx{\Serefwithpage}[2][1,usedefault]{\reformattedwithpage@
6925 {#1}{#2}{Seref}}
6926
6927 \newcommand{\reformatted@}[3]{%
6928   \def\do##1{%
6929     \setkeys[mac]{truefootnoteoption}{##1}%
6930   }%
6931   \notblank{#1}{\docsvlist{#1}}{ }%
6932   \xdef\@currentseries{#3}%
6933   \ifcempty{@#3prefixmore}%
6934     {\@apprefprefixsingle}%
6935     {%
6936       \IfEq{\xlineref{#2:start}}{\xlineref{#2:end}}{%
6937         {\csuse{@#3prefixsingle}}%
6938         {\csuse{@#3prefixmore}}%
6939       }%
6940     \ifboolexpr{%
6941       test{\ifcsundef{the@label#2:start}}%
6942       or test{\ifcsundef{the@label#2:end}}%
6943     }%
6944     {\led@warn@pairRefUndefined{#2}\nfss@text{\reset@font\bfseries ??}}%
6945     {%
6946       \def\@this@crossref@start{#2:start}%
6947       \def\@this@crossref@end{#2:end}%
6948       \xdef\annot@start{\xannotationref{#2:start}}%
6949       \xdef\annot@end{\xannotationref{#2:end}}%
6950       \printlines\xpageref{#2:start}|\xlineref{#2:start}|\xsublineref{#2:
6951 start}|\xpageref{#2:end}|\xlineref{#2:end}|\xsublineref{#2:end}|\relax|\
6952 xflagref{#2:start}}|}%
6953       \undef\@this@crossref@end%
6954       \undef\@this@crossref@start%
6955     }%
6956   \def\do##1{%
6957     \setkeys[mac]{falsefootnoteoption}{##1}%
6958   }%

```

```

6957 \notblank{#1}{\docsvlist{#1}}{ }%
6958 }%
6959
6960 \newcommand{\reformattedwithpage@}[3]{%
6961   \def\do##1{%
6962     \setkeys[mac]{truefootnoteoption}{##1}%
6963   }%
6964   \notblank{#1}{\docsvlist{#1}}{ }%
6965   \xdef\@currentseries{#3withpage}%
6966   \ifboolexpr{%
6967     test{\ifcsundef{the@label#2:start}}%
6968     or test{\ifcsundef{the@label#2:end}}%
6969   }%
6970   {\led@warn@pairRefUndefined{#2}\nfss@text{\reset@font\bfseries ??}}%
6971   {%
6972     \def\@this@crossref@start{#2:start}%
6973     \def\@this@crossref@end{#2:end}%
6974     \printendlines\xpageref{#2:start}|\xlineref{#2:start}|\xsublineref{#2:
start}|\xpageref{#2:end}|\xlineref{#2:end}|\xsublineref{#2:end}|\relax|\
xflagref{#2:start}|}%
6975     \undef\@this@crossref@end%
6976     \undef\@this@crossref@start%
6977   }%
6978   \def\do##1{%
6979     \setkeys[mac]{falsefootnoteoption}{##1}%
6980   }%
6981   \notblank{#1}{\docsvlist{#1}}{ }%
6982 }%
6983
6984 \newcommand{\reformattedonlypage@}[3]{%
6985   \def\do##1{%
6986     \setkeys[mac]{truefootnoteoption}{##1}%
6987   }%
6988   \notblank{#1}{\docsvlist{#1}}{ }%
6989   \xdef\@currentseries{#3onlypage}%
6990   \ifboolexpr{%
6991     test{\ifcsundef{the@label#2:start}}%
6992     or test{\ifcsundef{the@label#2:end}}%
6993   }%
6994   {\led@warn@pairRefUndefined{#2}\nfss@text{\reset@font\bfseries ??}}%
6995   {\ifnumequal{\xpageref{#2:end}}{\xpageref{#2:start}}%
6996     {%
6997       \ifcsvoid{#3onlypage@prefixsingle}%
6998       {}%
6999       {\csletcs{Xendbeforepagenumber@#3onlypage}{#3onlypage@prefixsingle
}}%
7000       \printnpnum{%
7001         \wrap@edcrossref{#2:start}{\xpageref{#2:start}}%
7002       }%
7003     }%

```

```

7004 {%
7005   \ifcvoid{#3onlypage@prefixmore}%
7006   {}%
7007   {\csletcs{Xendbeforepagenumber@#3onlypage}{#3onlypage@prefixmore}}%
7008   \ifdefined\linangesep%
7009     \printnpnum{%
7010       \wrap@edcrossref{#2:start}{\xpageref{#2:start}}}%
7011     \linangesep%
7012     \wrap@edcrossref{#2:end}{\xpageref{#2:end}}}%
7013   }%
7014   \else%
7015     \printnpnum{%
7016       \wrap@edcrossref{#2:start}{\xpageref{#2:start}}}%
7017     \csuse{Xendlinangeseparator@%currentseries}%
7018     \wrap@edcrossref{#2:end}{\xpageref{#2:end}}}%
7019   }%
7020   \fi%
7021 }%
7022 }%
7023 \def\do##1{%
7024   \setkeys{mac}{falsefootnoteoption}{##1}%
7025   }%
7026   \notblank{#1}{\docsvlist{#1}}{}%
7027 }%
7028 %

```

**\edmakelabel** Sometimes the `\edlabel` command cannot be used to specify exactly the page and line desired; you can use the `\edmakelabel` macro make your own label. For example, if you insert `\edmakelabel{elephant}{10|25|0}` you will have created a new label, and a later call to `\edpageref{elephant}` would print ‘10’ and `\lineref{elephant}` would print ‘25’. The sub-line number here is zero. `\edmakelabel` takes a label, followed by a page and a line number(s) as arguments.  $\TeX$  defines a `\makelabel` macro which is used in lists. Peter Wilson has changed the name to `\edmakelabel`.

```

7029 \newcommand*{\edmakelabel}[2]{\expandafter\xdef\csname the@label#1\
7030   endcsname{#2}}
7031 %

```

(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. 142 and V.9 p. 106), 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`.

```
7032 \pretocmd{\XR@test}%
7033   {\XR@test@mac+++#1#2#3#4+++}%
7034   {}%
7035   {}%
7036 %
```

`\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`.

```
7037 \long\def\xR@test@mac+++#1+++{\XR@test@mac@test#1}
7038 %
```

`\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`.

```
7039 \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
7040   \ifx#1\l@dmake@labels%
7041     \l@dmake@labels#2%
7042   \else
7043     \ifx#1\l@dmake@labelsR%
7044       \l@dmake@labelsR #2%
7045     \fi%
7046   \fi%
7047 }%
7048 %
```

## XXVI Sidenotes

Regular `\marginpars` do not work inside numbered text — they do not produce any note but do put an extra unnumbered blank line into the text.

`\@xympar` Changing `\@xympar` a little at least ensures that `\marginpars` in numbered text do not disturb the flow.

```
7049 \pretocmd{\@xympar}%
7050   {\ifnumberedpar@
7051     \led@warn@NoMarginpars
7052     \@esphack
7053   \else}%
7054   {}%
7055   {}%
7056
7057 \apptocmd{\@xympar}%
7058   {\fi}%
```

```

7059 {}
7060 {}
7061
7062 %

```

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

```

7063 \newcount\sidenote@margin
7064 \newcommand*{\sidenotemargin}[1]{\%
7065   \l@dgetsidenote@margin{#1}%
7066   \ifnum\@l@dttempcntb>\m@ne
7067     \ifledRcol
7068       \global\sidenote@marginR=\@l@dttempcntb
7069     \else
7070       \global\sidenote@margin=\@l@dttempcntb
7071     \fi
7072   \fi}}
7073 \newcommand*{\l@dgetsidenote@margin}[1]{\%
7074   \def\@tempa{#1}\def\@tempb{left}%
7075   \ifx\@tempa\@tempb
7076     \@l@dttempcntb \z@
7077   \else
7078     \def\@tempb{right}%
7079     \ifx\@tempa\@tempb
7080       \@l@dttempcntb \@ne
7081     \else
7082       \def\@tempb{outer}%
7083       \ifx\@tempa\@tempb
7084         \@l@dttempcntb \tw@
7085       \else
7086         \def\@tempb{inner}%
7087         \ifx\@tempa\@tempb
7088           \@l@dttempcntb \thr@@
7089         \else
7090           \led@warn@BadSidenotemargin
7091           \@l@dttempcntb \m@ne
7092         \fi

```

```

7093     \fi
7094     \fi
7095     \fi}
7096 \sidenotemargin{right}
7097
7098 %

```

`\l@dlp@rbox` We need two boxes to store sidenote texts.

```

\l@drp@rbox
7099 \newbox\l@dlp@rbox
7100 \newbox\l@drp@rbox
7101
7102 %

```

`\ledlsnotewidth` These specify the width of the left/right boxes (initialised to `\marginparwidth`), their  
`\ledrsnotewidth` distance from the text (initialised to `\linenumsep`), and the fonts used.

```

\ledlsnotesep 7103 \newdimen\ledlsnotewidth \ledlsnotewidth=\marginparwidth
\ledrsnotesep 7104 \newdimen\ledrsnotewidth \ledrsnotewidth=\marginparwidth
\ledlsnotefontsetup 7105 \newdimen\ledlsnotesep \ledlsnotesep=\linenumsep
\ledrsnotefontsetup 7106 \newdimen\ledrsnotesep \ledrsnotesep=\linenumsep
7107 \newcommand*{\ledlsnotefontsetup}{\raggedleft\footnotesize}
7108 \newcommand*{\ledrsnotefontsetup}{\raggedright\footnotesize}
7109
7110 %

```

`\ledleftnote` `\ledleftnote`, `\ledrightnote`, `\ledinnernote`, `\ledouternote` are the user com-  
`\ledrightnote` mands for left, right, inner and outer sidenotes. The two last one are just alias for the  
`\ledinnernote` two first one, depending of the page number. `\ledsidenote{<text>}` is the command  
`\ledouternote` for a moveable sidenote.

```

\ledsidenote 7111 \newcommand*{\ledleftnote}[1]{\edtext{}{\l@dlsnote{#1}}}
7112 \newcommand*{\ledrightnote}[1]{\edtext{}{\l@drsnote{#1}}}
7113 \newcommand*{\ledsidenote}[1]{\edtext{}{\l@dcsnote{#1}}}%
7114 \newcommand*{\ledinnernote}[1]{\edtext{}{\l@disnote{#1}}}%
7115 \newcommand*{\ledouternote}[1]{\edtext{}{\l@dosnote{#1}}}%
7116 %

```

`\l@dlsnote` . The ‘footnotes’ for left, right, and moveable sidenotes. The whole scheme is reminiscent  
`\l@drsnote` of the critical footnotes code.

```

\l@dcsnote 7117 \newif\ifrighnoteup
\l@desnote 7118 \rightnoteuptrue
\l@disnote 7119
7120 \newcommand*{\l@dlsnote}[1]{%
7121 \begingroup%
7122 \newcommand{\content}{#1}%
7123 \ifnumberedpar@
7124 \ifledRcol%
7125 \xright@appenditem{\noexpand\l@dlsnote{\expandonce\content}}%

```

```

7126         \to\inserts@listR
7127         \global\advance\insert@countR \@ne%
7128     \else%
7129         \xright@appenditem{\noexpand\vl@dlsnote{\expandonce\content}}}%
7130         \to\inserts@list
7131         \global\advance\insert@count \@ne%
7132     \fi
7133 \fi%
7134 \ignorespaces%
7135 \endgroup%
7136 }%
7137
7138 \newcommand*{\l@drsnote}[1]{%
7139     \begingroup%
7140     \newcommand{\content}{#1}%
7141     \ifnumberedpar@
7142         \ifledRcol%
7143             \xright@appenditem{\noexpand\vl@drsnote{\expandonce\content}}}%
7144             \to\inserts@listR
7145             \global\advance\insert@countR \@ne%
7146         \else%
7147             \xright@appenditem{\noexpand\vl@drsnote{\expandonce\content}}}%
7148             \to\inserts@list
7149             \global\advance\insert@count \@ne%
7150         \fi
7151     \fi\ignorespaces%
7152 \endgroup%
7153 }%
7154
7155 \newcommand*{\l@dcsnote}[1]{%
7156     \begingroup%
7157     \newcommand{\content}{#1}%
7158     \ifnumberedpar@
7159         \ifledRcol%
7160             \xright@appenditem{\noexpand\vl@dcsnote{\expandonce\content}}}%
7161             \to\inserts@listR
7162             \global\advance\insert@countR \@ne%
7163         \else%
7164             \xright@appenditem{\noexpand\vl@dcsnote{\expandonce\content}}}%
7165             \to\inserts@list
7166             \global\advance\insert@count \@ne%
7167         \fi
7168     \fi\ignorespaces%
7169 \endgroup%
7170 }%
7171
7172 \newcommand*{\l@disnote}[1]{%
7173     \begingroup%
7174     \newcommand{\content}{#1}%
7175     \ifnumberedpar@

```

```

7176 \ifledRcol%
7177 \xright@appenditem{\noexpand\vl@disnote{\expandonce\content}}}%
7178 \to\inserts@listR%
7179 \global\advance\insert@countR \@ne%
7180 \else%
7181 \xright@appenditem{\noexpand\vl@disnote{\expandonce\content}}}%
7182 \to\inserts@list%
7183 \global\advance\insert@count \@ne%
7184 \fi%
7185 \fi\ignorespaces%
7186 \endgroup%
7187 }%
7188
7189 \newcommand*{\l@dosnote}[1]{%
7190 \begingroup%
7191 \newcommand{\content}{#1}%
7192 \ifnumberedpar%
7193 \ifledRcol%
7194 \xright@appenditem{\noexpand\vl@dosnote{\expandonce\content}}}%
7195 \to\inserts@listR%
7196 \global\advance\insert@countR \@ne%
7197 \else%
7198 \xright@appenditem{\noexpand\vl@dosnote{\expandonce\content}}}%
7199 \to\inserts@list%
7200 \global\advance\insert@count \@ne%
7201 \fi%
7202 \fi\ignorespaces%
7203 \endgroup%
7204 }%
7205
7206 %

```

**\vl@dlsnote** Put the left/right text into boxes, but just save the moveable text. **\l@dcsnotetext**, **\vl@drsnote** **\l@dcsnotetext@l** and **\l@dcsnotetext@r** are etoolbox's lists which will store the content of sidenotes. We store the content in lists, because we need to loop later on them, in case many sidenote co-exist for the same line. That is there some special test **\vl@disnote** to do, in order to:

- Store the content of **\ledsidenote** to **\l@dcsnotetext** in any cases.
- Store the content of **\rightsidenote** to:
  - **\l@dcsnotetext** if **\ledsidenote** is to be put on right.
  - **\l@dcsnotetext@r** if **\ledsidenote** is to be put on left.
- Store the content of **\leftsidenote** to:
  - **\l@dcsnotetext** if **\ledsidenote** is to be put on left.
  - **\l@dcsnotetext@l** if **\ledsidenote** is to be put on right.



\vl@disnote and \vl@dosnote just call \vl@dlsnote or \vl@drsnote, depending of the page.

```

7207 \newcommand*{\vl@dlsnote}[1]{%
7208   \ifledRcol{%
7209     \@l@tempcntb=\sidenote@marginR%
7210     \ifnum\@l@tempcntb>\@ne%
7211       \advance\@l@tempcntb by\page@numR%
7212     \fi%
7213   \else%
7214     \@l@tempcntb=\sidenote@margin%
7215     \ifnum\@l@tempcntb>\@ne%
7216       \advance\@l@tempcntb by\page@num%
7217     \fi%
7218   \fi%
7219   \ifodd\@l@tempcntb%
7220     \listgadd{\l@dcsnotetext@l}{#1}%
7221   \else%
7222     \listgadd{\l@dcsnotetext}{#1}%
7223   \fi
7224 }
7225 \newcommand*{\vl@drsnote}[1]{%
7226   \ifledRcol{%
7227     \@l@tempcntb=\sidenote@marginR%
7228     \ifnum\@l@tempcntb>\@ne%
7229       \advance\@l@tempcntb by\page@numR%
7230     \fi%
7231   \else%
7232     \@l@tempcntb=\sidenote@margin%
7233     \ifnum\@l@tempcntb>\@ne%
7234       \advance\@l@tempcntb by\page@num%
7235     \fi%
7236   \fi%
7237   \ifodd\@l@tempcntb%
7238     \listgadd{\l@dcsnotetext}{#1}%
7239   \else%
7240     \listgadd{\l@dcsnotetext@r}{#1}%
7241   \fi%
7242 }
7243 \newcommand*{\vl@dcsnote}[1]{\listgadd{\l@dcsnotetext}{#1}}
7244
7245 \newcommand{\vl@disnote}[1]{%
7246   \ifledRcol{%
7247     \@tempcnta=\page@numR%
7248   \else%
7249     \@tempcnta=\page@num%
7250   \fi%
7251   \ifodd\@tempcnta% ODD => right page => inner side = left side
7252     \vl@dlsnote{#1}%
7253   \else%
```

```

7254 \vl@drsnote{#1}%
7255 \fi%
7256 }%
7257
7258 \newcommand{\vl@dosnote}[1]{%
7259 \ifledRcol%
7260 \@tempcnta=\page@numR%
7261 \else%
7262 \@tempcnta=\page@num%
7263 \fi%
7264 \ifodd\@tempcnta% ODD => right page => outer side = right side
7265 \vl@drsnote{#1}%
7266 \else%
7267 \vl@dlsnote{#1}%
7268 \fi%
7269 }%
7270
7271 %

```

`\setl@dlp@rbox` `\setl@dlprbox{<lednums>}{<tag>}{<text>}` puts `<text>` into the `\l@dlp@rbox` box. And similarly for the right side box. It is these boxes that finally get displayed in the margins.

```

7272 \newcommand*{\setl@dlp@rbox}[1]{%
7273 \begingroup%
7274 \parindent\z@\hsize=\ledlsnotewidth%
7275 \ledlsnotefontsetup%We kept it outside of the vbox, because can affect
the ragging
7276 \global\setbox\l@dlp@rbox%
7277 \ifleftnoteup%
7278 =\vbox to\z@{\ledlsnotefontsetup\vss #1}}%We put \
ledlsnotefontsetup inside footnote because required for color command. Note
the {} to keep setting local.
7279 \else%
7280 =\vbox to 0.70\baselineskip{\ledlsnotefontsetup\strut#1\vss}}%
7281 \fi%
7282 \endgroup%
7283 }
7284
7285 \newcommand*{\setl@drp@rbox}[1]{%
7286 \begingroup%
7287 \parindent\z@\hsize=\ledrsnotewidth%
7288 \ledrsnotefontsetup%We kept it outside of the vbox, because can affect
the ragging
7289 \global\setbox\l@drp@rbox%
7290 \ifrightrightnoteup%
7291 =\vbox to\z@{\ledrsnotefontsetup\vss#1}}%We put \ledrsnotefontsetup
inside footnote because required for color command. Note the {} to keep
setting local.
7292 \else%

```

```

7293     =\vbox to0.7\baselineskip{{\ledrsnotefontsetup\strut#1\vss}}}%
7294     \fi%
7295     \endgroup%
7296 }%
7297 \newif\ifleftnoteup
7298 \leftnoteuptrue
7299 %

```

**\@sidenotesep** This macro is used to separate sidenotes of the same line.

```

7300 \newcommand{\setsidenotesep}[1]{\gdef\@sidenotesep{#1}}
7301 \newcommand{\@sidenotesep}{, }
7302 %

```

**\affixside@note** This macro puts any moveable sidenote text into the left or right sidenote box, depending on which margin it is meant to go in. It's a very much stripped down version of `\affixline@num`.

Before do it, we concatenate all moveable sidenotes of the line, using `\@sidenotesep` as separator. It is the result that we put on the sidenote.

```

7303 \newcommand*{\affixside@note}{%
7304     \prepare@edindex@fornote{\the\page@num|\the\line@num|\the\subline@num|\
7305     the\page@num|\the\line@num|\the\subline@num|}%
7306     \def\sidenotecontent@{}%
7307     \numgdef{\itemcount@}{0}%
7308     \def\do##1{%
7309         \ifnumequal{\itemcount@}{0}%
7310         {%
7311             \appto\sidenotecontent@{##1}}% Not print not separator before
7312             the 1st note
7313             \appto\sidenotecontent@{\@sidenotesep ##1}%
7314         }%
7315         \numgdef{\itemcount@}{\itemcount@+\@ne}%
7316     }%
7317     \dolistloop{\l@dcsonotetext}%
7318     \ifnumgreater{\itemcount@}{1}{\led@err@ManySidenotes}{}%
7319 %

```

And we do the same for left and right notes (not movable).

```

7318 \gdef\@templ@d{%
7319 \gdef\@templ@n{\l@dcsonotetext\l@dcsonotetext@l\l@dcsonotetext@r}%
7320 \ifx\@templ@d\@templ@n \else%
7321     \if@twocolumn%
7322         \if@firstcolumn%
7323             \setl@dlp@rbox{##1}{\sidenotecontent@}%
7324         \else%
7325             \setl@drp@rbox{\sidenotecontent@}%
7326         \fi%
7327     \else%

```

```

7328 \l@dttempcntb=\sidenote@margin%
7329 \ifnum\l@dttempcntb>\@ne%
7330 \advance\l@dttempcntb by\page@num%
7331 \fi%
7332 \ifodd\l@dttempcntb%
7333 \setl@drp@rbox{\sidenotecontent@}%
7334 \gdef\sidenotecontent@{}%
7335 \numgdef{\itemcount@}{0}%
7336 \dolistloop{\l@dcsnotetext@l}%
7337 \ifnumgreater{\itemcount@}{1}{\led@err@ManyLeftnotes}{}%
7338 \setl@dlp@rbox{\sidenotecontent@}%
7339 \else%
7340 \setl@dlp@rbox{\sidenotecontent@}%
7341 \gdef\sidenotecontent@{}%
7342 \numgdef{\itemcount@}{0}%
7343 \dolistloop{\l@dcsnotetext@r}%
7344 \ifnumgreater{\itemcount@}{1}{\led@err@ManyRightnotes}{}%
7345 \setl@drp@rbox{\sidenotecontent@}%
7346 \fi%
7347 \fi%
7348 \fi%
7349 \advance\@edindex@fornote@m@ne%
7350 }
7351 %

```

## XXVII Minipages and such

We can put footnotes into minipages. The preparatory code has been set up earlier, all that remains is to ensure that it is available inside a minipage box. This requires some alteration to the kernel code, specifically the `\@iiminipage` and `\endminipage` macros. We will arrange this so that additional series can be easily added.

`\l@dfeetbeginmini` These will be the hooks in `\@iiminipage` and `\endminipage`.  
`\l@dfeetendmini` They can be extended to handle other things if necessary.

```

7352 \ifnoledgroup@else%
7353 \newcommand*{\l@dfeetbeginmini}{\@ledgrouptrue\l@dedbeginmini\l@dfambeginmini}
7354 \newcommand*{\l@dfeetendmini}{%
7355 \IfStrEq{critical-familiar}{\@mpfnpos}%
7356 {\l@dedendmini\l@dfamendmini}%
7357 {%
7358 \IfStrEq{familiar-critical}{\@mpfnpos}%
7359 {\l@dfamendmini\l@dedendmini}%
7360 {\do@feet@custom@order{mp@}{\@mpfnpos}}}%
7361 }%
7362 }%
7363 %

```

`\l@dedbeginmini` These handle the initiation and closure of critical footnotes in a minipage environment.

`\l@dedendmini`

`\mp@append@Xnotes`

```

7364 \newcommand*{\l@dedbeginmini}{%
7365   \unless\ifnocritical%
7366     \def\do##1{%
7367       \csletcs{v##1footnote}{mpv##1footnote}%
7368     }%
7369     \dolistloop{\@series}%
7370   \fi%
7371 }
7372 \newcommand*{\l@dedendmini}{%
7373   \unless\ifnocritical%
7374     \ifl@dpairing%
7375       \ifledRcol%
7376         \flush@notesR%
7377       \else%
7378         \flush@notes%
7379       \fi%
7380     \fi
7381     \def\do##1{%
7382       \mp@append@Xnotes{##1}%
7383     }%
7384     \dolistloop{\@series}%
7385   \fi%
7386 }%
7387 \newcommand{\mp@append@Xnotes}[1]{%
7388   \ifvoid\csuse{mp#1footins}\else%
7389     \ifl@dpairing%
7390       \ifparledgroup%
7391         \ifledRcol%
7392           \dingdef{\parledgroup@beforenotesR}{\parledgroup@beforenotesR+\skip
\@nameuse{mp#1footins}}%
7393         \else%
7394           \dingdef{\parledgroup@beforenotesL}{\parledgroup@beforenotesL+\skip
\@nameuse{mp#1footins}}%
7395         \fi%
7396       \fi%
7397     \fi%
7398     \ifcsstring{series@display#1}{paragraph}{}{%
7399       \setbox\@nameuse{mp#1footins}=\vbox{%
7400         \csuse{Xnotefontsize@#1}%
7401         \ifcsdef{Xhsize}\csuse{series@display#1}@#1{%
7402           \hsize\csuse{Xhsize}\csuse{series@display#1}@#1}%
7403         }{%
7404           \noindent\csuse{Xtxtbeforenotes@#1}%
7405           \unvbox\@nameuse{mp#1footins}%
7406           \@parboxrestore%
7407         }%
7408       }%
7409       \csuse{mp#1footgroup}{#1}%

```

```

7410 \fi%
7411 }%
7412 %

```

`\l@dfambeginmini` These handle the initiation and closure of familiar footnotes in a minipage environment.

```

\l@dfamendmini
\mp@append@notesX
7413 \newcommand*{\l@dfambeginmini}{%
7414 \unless\ifnofamiliar{%
7415 \def\do##1{\csletcs{vfootnote##1}{mpvfootnote##1}}%
7416 \dolistloop{\@series}%
7417 \fi%
7418 }%
7419
7420 \newcommand*{\l@dfamendmini}{%
7421 \unless\ifnofamiliar{%
7422 \def\do##1{%
7423 \mp@append@notesX{##1}%
7424 }%
7425 \dolistloop{\@series}%
7426 \fi%
7427 }%
7428 \newcommand{\mp@append@notesX}[1]{%
7429 \ifvoid\csuse{mpfootins#1}\else%
7430 \ifcsstring{series@displayX#1}{paragraph}{%
7431 \setbox\@nameuse{mpfootins#1}=\vbox{%
7432 \csuse{notefontsizeX@#1}%
7433 \ifcsdef{hsize\csuse{series@display#1}X@#1}{%
7434 \hsize\csuse{hsize\csuse{series@display#1}X@#1}%
7435 }{}%
7436 \noindent\csuse{txtbeforenotesX@#1}%
7437 \unvbox\@nameuse{mpfootins#1}%
7438 \@parboxrestore%
7439 }%
7440 }%
7441 \csuse{mpfootgroup#1}{#1}%
7442 \fi%
7443 }%
7444 %

```

`\@iiiminipage` This is our extended form of the kernel `\@iiiminipage` defined in `ltboxes.dtx`.

```

7445 \patchcmd%
7446 {\@iiiminipage}%
7447 {\let\@footnotetext\@mpfootnotetext}%
7448 {\let\@footnotetext\@mpfootnotetext\l@dfeetbeginmini}%
7449 {}%
7450 {\led@error@fail@patch@\@iiiminipage}%
7451 %

```

`\endminipage` This is our extended form of the kernel `\endminipage` defined in `ltboxes.dtx`.

```

7452 \patchcmd%
7453   {\endminipage}%
7454   {\footnoterule}%
7455   {\footnoterule\l@advance@parledgroup@beforenormalnotes}%
7456   {}%
7457   {\led@error@fail@patch@endminipage}
7458
7459 \patchcmd%
7460   {\endminipage}%
7461   {\@minipagefalse}%
7462   {\l@dfeetendmini\@minipagefalse}%
7463   {}%
7464   {\led@error@fail@patch@endminipage}
7465
7466 %

```

`\l@dunboxmpfoot` `\@ldunboxmpfoot` insert normal footnotes for ledgroup.  
`edgroup@beforenormalnotes`

```

7467 \newcommand*{\l@dunboxmpfoot}{%
7468   \vskip\skip\@mpfootins
7469   \normalcolor
7470   \footnoterule
7471   \l@advance@parledgroup@beforenormalnotes
7472   \unvbox\@mpfootins%
7473 }
7474 %

```

When using parallel ledgroup, we need to store the vertical space added before footnote, in order to compensate them between left and right pages.

```

7475 \newcommand{\l@advance@parledgroup@beforenormalnotes}{%
7476   \ifparledgroup
7477     \ifl@dpairing
7478       \ifledRcol
7479         \dimdef{\parledgroup@beforenotesR}{\parledgroup@beforenotesR+
skip\@mpfootins}
7480       \else
7481         \dimdef{\parledgroup@beforenotesL}{\parledgroup@beforenotesL+
skip\@mpfootins}
7482       \fi
7483     \fi
7484   \fi
7485 }
7486 %

```

`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.

```

7487

```

```

7488 \newenvironment{ledgroup}{%
7489   \resetprevpage@num%
7490   \def\@mpfn{mpfootnote}\def\thempfn{\thempfootnote}\c@mpfootnote\z@%
7491   \let\@footnotetext\@mpfootnotetext
7492   \l@dfetbeginmini%
7493 }{%
7494   \par
7495   \unskip
7496   \ifvoid\@mpfootins\else
7497     \l@dunboxmpfoot
7498   \fi
7499   \l@dfetendmini%
7500   \@ledgroupfalse%
7501 }
7502
7503
7504 %

```

`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.

```

7505 \newenvironment{ledgroupsize}[2][1]{%
7506 %

```

Set the various text measures.

```

7507 \hsize #2\relax
7508 %

```

Initialize fills for centering.

```

7509 \let\ledllfill\hfil
7510 \let\ledrlfill\hfil
7511 \def\@tempa{#1}\def\@tempb{1}%
7512 %

```

Left adjusted numbered lines

```

7513 \ifx\@tempa\@tempb
7514 \let\ledllfill\relax
7515 \else
7516 \def\@tempb{r}%
7517 \ifx\@tempa\@tempb
7518 %

```

Right adjusted numbered lines

```

7519 \let\ledrlfill\relax
7520 \fi
7521 \fi
7522 %

```

Set up the footnoting.



```

7523 \def\@mpfn{mpfootnote}\def\thempfn{\thempfootnote}\c@mpfootnote\z@
7524 \let\@footnotetext\@mpfootnotetext
7525 \l@dfetbeginmini%
7526 }{f%
7527 \par
7528 \unskip
7529 \ifvoid\@mpfootins\else
7530 \l@dunboxmpfoot
7531 \fi
7532 \l@dfetendmini%
7533 }
7534
7535 %

```

Close the \ifnoledgroup@else.

```

7536 \fi%
7537 %

```

**\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.

```

7538 \newif\ifledgroupnotesL@
7539 \newif\ifledgroupnotesR@
7540 %

```

## 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.

```

7541 \AtBeginDocument{%
7542 \unless\ifl@imakeidx%
7543 \@ifpackageloaded{imakeidx}{\led@error@PackageAfterEledmac{imakeidx}}{}
7544 %
7545 \fi%
7546 \unless\ifl@indextools%
7547 \@ifpackageloaded{indextools}{\led@error@PackageAfterEledmac{indextools}}{}
7548 %
7549 \fi%
7550 \unless\ifl@footmisc%
7551 \@ifpackageloaded{footmisc}{\led@error@PackageAfterEledmac{footmisc}}{}
7552 %
7553 \fi%
7554 %

```

## XXVIII.2 Auxiliary macros for `\edindex`

`\pagelinesep` `\edindexlab` `\c@labidx` In order to get a correct line number we have to use the label/ref mechanism. These macros are for that.

```

7553 \newcommand{\pagelinesep}{-}
7554 \newcommand{\edindexlab}{$&}
7555 \newcounter{labidx}
7556 \setcounter{labidx}{0}
7557
7558 %

```

`\doedindexlabel` This macro sets an `\edlabel`.

```

7559 \newcommand{\doedindexlabel}{%
7560   \stepcounter{labidx}%
7561   \edlabel{\edindexlab\thelabidx}%
7562 }
7563
7564 %

```

`\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`.

```

7565 \newcounter{pageline}%
7566 \renewcommand{\thepageline}{%
7567   \thepage%
7568   \pagelinesep%
7569   \xlineref{\edindexlab\thelabidx}%
7570 }
7571 %

```

`\thestartpageline` `\theendpageline` These macros make up the page/line start/end number when the `\edindex` command is called in critical notes.

```

7572 \newcommand{\thestartpageline}{%
7573   \l@dparsedstartpage%
7574   \pagelinesep%
7575   \l@dparsedstartline%
7576 }
7577 \newcommand{\theendpageline}{%
7578   \l@dparsedendpage%
7579   \pagelinesep%
7580   \l@dparsedendline%
7581 }
7582 %

```

### XXVIII.3 Code specific to \edindex in critical footnotes

`\@edindex@fornote@` This counter is incremented at the beginning of each note (either a footnote or a side-note), and decremented at the end of each note. If its value is greater than 0, that means we are inside a note.

```
7583 \newcount\@edindex@fornote@
7584 %
```

`\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 |.

```
7585 \newcommand{\prepare@edindex@fornote}[1]{%
7586   \l@dp@rsefootspec#1|}%
7587   \advance\@edindex@fornote@\@ne%
7588 }
7589 %
```

`\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.

```
7590 \newcommand{\get@edindex@ledinnote@command}{%
7591   \ifxindy@%
7592     \gdef\@ledinnote@command{%
7593       ledinnote\thelabidx%
7594     }%
7595     \ifxindyhyperref@%
7596       \immediate\write\eledmac@xindy@out{%
7597         (define-attributes ("ledinnote\thelabidx"))^^J
7598         \space\space(markup-locref^^J
7599         \eledmacmarkuplocrefdepth^^J
7600         :open "\string\ledinnote[\edindexlab\thelabidx]{\@index@command
7601       }{"^^J
7602         :close "}"^^J
7603         :attr "ledinnote\thelabidx"^^J
7604       )
7605     }%
7606   \else%
7607     \immediate\write\eledmac@xindy@out{%
7608       (define-attributes ("ledinnote\thelabidx"))^^J
7609       \space\space(markup-locref^^J
7610       \eledmacmarkuplocrefdepth^^J
7611       :open "\string\ledinnote{\@index@command}{"}^^J
7612       :close "}"^^J
7613       :attr "ledinnote\thelabidx"^^J
7614     )
7615   }
```

```

7614     }%
7615     \fi%
7616 %

```

If we do not use xindy option, `\@ledinnote@command` will produce something like `ledinnote{formattingcommand}`.

```

7617 \else%
7618   \gdef\@ledinnote@command{%
7619     ledinnote[\edindexlab\thelabidx]{\@index@command}%
7620   }%
7621   \fi%
7622 }
7623 %

```

## 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 |.

```

7624 \def\get@index@command#1|#2+{%
7625   \gdef\@index@txt{#1}%
7626   \gdef\@index@command{#2}%
7627   \xdef\@index@parenthesis{%
7628     \IfBeginWith{\@index@command}{(}{%
7629       \StrGobbleLeft{\@index@command}{1}{\@index@command}%
7630       \global\let\@index@command\@index@command%
7631       \xdef\@index@parenthesis{(%
7632     }{)%
7633     \IfBeginWith{\@index@command}{(}{%
7634       \StrGobbleLeft{\@index@command}{1}{\@index@command}%
7635       \global\let\@index@command\@index@command%
7636       \xdef\@index@parenthesis{)%
7637     }{)%
7638   }
7639 %

```

## 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`

```

7640 \newcommandx{\ledinnote}[3][1,usedefault]{%
7641   \ifboolexpr{%
7642     test{\ifdefequal{\iftrue}{\ifHy@hyperindex}}%
7643     or%
7644     bool {xindyhyperref}%
7645   }%
7646   {%
7647     \csuse{#2}{\hyperlink{#1}{\ledinnotemark{#3}}}%

```

```

7648 }%
7649 {%
7650 \csuse{#2}{\ledinnotemark{#3}}%
7651 }%
7652 }%
7653 \newcommand{\ledinnotehyperpage}[2]{\csuse{#1}{\ledinnotemark{\hyperpage
7654 {#2}}}}%
7655 \newcommand{\ledinnotemark}[1]{#1\emph{n}}%
7656 %

```

## XXVIII.6 Main code

Eledmac and ledmac were using the specific indexing tools of the memoir in order to allow multiple index. However, eledmac used imakeidx or indextools tools when one these two package was loaded. This system forced to maintained a double code, which was not very useful. Since reledmac, we use only the imakeidx or indextools tools.

The memoir class provides more flexible indexing than the standard classes. We need different code if the memoir class is being used, except if imakeidx or indextools is used.

`\edindex` Write the index information to the idx file.

```

\@wredindex
\dummy@edindex
7656 \newcommandx{\@wredindex}[2][1=\expandonce\jobname,usedefault]{%#1 = the
index name, #2 = the text
7657 \begingroup%
7658 \let\emph\@firstofone%
7659 \let\textbf\@firstofone%
7660 \let\textit\@firstofone%
7661 \let\textmd\@firstofone%
7662 \let\textnormal\@firstofone%
7663 \let\textrm\@firstofone%
7664 \let\textsc\@firstofone%
7665 \let\textsf\@firstofone%
7666 \let\textsl\@firstofone%
7667 \let\texttt\@firstofone%
7668 \let\textup\@firstofone%
7669 \xdef\@tmp{#2}%To be used in IfSubStr instead of #2 directly. Avoid
some expansion bugs (for example with \edindex{textsc{something}})
7670 \endgroup%
7671 \ifl@imakeidx%
7672 \ifnum\@edindex@fornote@>\z@%
7673 \IfSubStr[1]{\@tmp}{|}{\get@index@command#2+}{\get@index@command#2|+}
%
7674 \get@edindex@ledinnote@command%
7675 \expandafter\imki@wrindexentry{#1}{\@index@txt|(\@ledinnote@command
}{\thestartpageline}%
7676 \expandafter\imki@wrindexentry{#1}{\@index@txt|)\@ledinnote@command
}{\theendpageline}%

```

```

7677 \else%
7678 \get@edindex@hyperref{#2}%
7679 \imki@wrindexentry{#1}{\@index@txt\@edindex@hyperref}{\thepageline}%
7680 \fi%
7681 \else%
7682 \ifnum\@edindex@fornote@>\z@%
7683 \IfSubStr[1]{\@tmp}{|}{\get@index@command#2+}{\get@index@command#2|+}
7684 %
7685 \get@edindex@ledinnote@command%
7686 \expandafter\protected@write\@indexfile{}%
7687 {\string\indexentry{\@index@txt|(\@ledinnote@command){\thestartpageline}
7688 }%
7689 \expandafter\protected@write\@indexfile{}%
7690 {\string\indexentry{\@index@txt|)\@ledinnote@command}{\theendpageline}
7691 }%
7692 \else%
7693 \protected@write\@indexfile{}%
7694 {\string\indexentry{#2}{\thepageline}
7695 }%
7696 \fi%
7697 \endgroup
7698 \esphack%
7699 }
7700 %

```

Need to add the definition of `\edindex` to `\makeindex`, and initialise `\edindex` to do nothing.

```

7701 \pretocmd{\makeindex}{%
7702 \def\edindex{%
7703 \ifboolexpr{bool{numbering} or bool{numberingR} or bool{
7704 l@dprintingpages} or bool{l@dprintingcolumns}}{%
7705 \bsphack%
7706 \doedindexlabel%
7707 \begingroup%
7708 \sanitize%
7709 \wredindex%
7710 }%
7711 {%
7712 \led@warn@edinde@outsidenumbering%
7713 \index%
7714 }%
7715 }%
7716 }%
7717 {\led@error@fail@patch@makeindex}%
7718 \newcommand{\edindex}[1]{\bsphack\esphack}
7719 \newcommandx{\dummyedindex}[2][1=\expandonce\jobname,usedefault]{}%
7720 %

```

## XXVIII.7 Hyperlink

`\hyperlinkformat` `\hyperlinkformat` command is to be used to have both a internal hyperlink and a format, when indexing.

```
7721 \newcommand{\hyperlinkformat}[3]{%
7722   \ifstrempy{#1}%
7723   {\hyperlink{#2}{#3}}%
7724   {\csuse{#1}{\hyperlink{#2}{#3}}}%
7725 }%
7726 %
```

`\hyperlinkR` `\hyperlinkR` command is to be used to create a internal hyperlink and `\ledRflag`, when indexing.

```
7727 \newcommand{\hyperlinkR}[2]{%
7728   \hyperlink{#1}{#2\@Rlineflag}%
7729 }%
7730 %
7731 %
```

`\hyperlinkformatR` `\hyperlinkformatR` command is to be used to create a internal hyperlink, a format and a `\@Rlineflag`, when indexing.

```
7732 \newcommand{\hyperlinkformatR}[3]{%
7733   \hyperlinkformat{#1}{#2}{#3\@Rlineflag}%
7734 }%
7735 %
7736 %
```

`\get@edindex@hyperref` `\get@edindex@hyperref` is to be used to define the `\@edindex@hyperref` macro, which, in index, links to the point where the index was called (with `hyperref`).

```
7737 \newcommand{\get@edindex@hyperref}[1]{%
7738 %
```

We have to disable temporary spaces to work through a `xstring` bug (or feature?)

```
7739 \edef\temp@{%
7740   \catcode\ =9 %space need for catcode
7741   \detokenize{#1}%For active character in unicode
7742   \catcode\ =10 % space need for catcode
7743 }%
7744 %
```

Now, we define `\@edindex@hyperref` if the `hyperindex` of `hyperref` is enabled.

```
7745 \ifdefequal{\iftrue}{\ifHy@hyperindex}{%
7746   \IfSubStr{\temp@}{|}%
7747   {\get@index@command#1+%
7748   \ifledRcol%
7749   \gdef\@edindex@hyperref{|\@index@parenthesis %space kept
```

```

7750     hyperlinkformatR{\@index@command}%
7751     {\edindexlab\thelabidx}}%
7752   \else%
7753     \gdef\@edindex@hyperref{|\@index@parenthesis %space kept
7754     hyperlinkformat{\@index@command}%
7755     {\edindexlab\thelabidx}}%
7756   \fi%
7757 }%
7758 {\get@index@command#1|+%
7759   \ifledRcol%
7760     \gdef\@edindex@hyperref{\hyperlinkR{\edindexlab\thelabidx}}%
7761   \else%
7762     \gdef\@edindex@hyperref{\hyperlink{\edindexlab\thelabidx}}%
7763   \fi%
7764 }%
7765 }%
7766 %

7767 % If we use both xindy and hyperref, first get the \protect\cs{
7768 index@command} command.
7769 % Then define \protect\cs{@edindex@hyperref} in the form \verb+eledmacXXX+
7770 % \begin{macrocode}
7771 {\ifxindyhyperref%
7772   \IfSubStr{\temp@}{|}%
7773   {\get@index@command#1+%
7774   {\get@index@command#1|+%
7775   \gdef\@edindex@hyperref{|eledmac\thelabidx}}%
7776 }%
7777 }%

```

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`.

```

7776   \IfStrEq{\@index@parenthesis}{(}%
7777   {%
7778     \csxdef{xindyparenthesis@\@index@txt}{\thelabidx}%
7779     \gdef\@edindex@hyperref{| (eledmac\thelabidx}%
7780   }%
7781   {}%
7782 %

```

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`.

```

7783   \IfStrEq{\@index@parenthesis}{)}%
7784   {%
7785     \xdef\@edindex@hyperref{|)eledmac\cuse{xindyparenthesis@\@index@txt}}%
7786     \global\csundef{xindyparenthesis@\@index@txt}%
7787   }%
7788 %

```



Write in the .xdy file the attributes of the location.

```

7789 {%
7790 \immediate\write\eledmac@xindy@out{%
7791 (define-attributes ("eledmac\thelabidx"))^^J
7792 \space\space(markup-locoref^^J
7793 \eledmacmarkuplocorefdepth^^J
7794 :open "\string\hyperlink%
7795 \ifledRcol R\fi%
7796 {\edindexlab\thelabidx}%
7797 {\ifdefempty{\@index@command}%
7798 {}%
7799 {\@backslashchar\@index@command}%
7800 {"^^J
7801 :close "}}^^J
7802 :attr "eledmac\thelabidx"^^J
7803 )
7804 }%
7805 }%
7806 %

```

And now, in any other case.

```

7807 \else%
7808 \gdef\@index@txt{#1}%
7809 \gdef\@edindex@hyperref{}%
7810 \fi%
7811 }%
7812 }
7813 %

```

## 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.

```

7814 \newcommand{\led@set@index@fornote}[1]{%
7815 \ifbool{indtl@innote}%
7816 {\let\index\nindex}%
7817 {}%
7818 \ifbool{indtl@notenumber}%
7819 {%
7820 \renewcommand{\index}[2][\indtl@jobname]{%
7821 \orig@@index[##1]{%
7822 ##2|innotenumber{\this@footnoteX@reading}%
7823 }%
7824 }%
7825 }%

```

```

7826 {}%
7827 }%
7828 %

```

`\led@reinit@index@fornote` The `\led@reinit@index@fornote` just reset the default value of `\index`.

```

7829 \newcommand{\led@reinit@index@fornote}{%
7830   \ifbool{indtl@innote}%
7831     {\let\index\orig@@index}%
7832   }%
7833   \ifbool{indtl@notenumber}%
7834     {\let\index\orig@@index}%
7835   }%
7836 }%
7837 %

```

## XXIX Glossaries

Here, we define the `\gls`-like commands prefixed by `ed`, only if the package `glossaries` is loaded.

```

7838 \AtBeginDocument{%
7839   \@ifpackageloaded{glossaries}{%
7840     %

```

First those which arguments are `[<options>]{<label>}[<insert>]`.

```

7841   \renewcommand{\do}[1]{%
7842     \expandafter\DeclareRobustCommand\csname ed#1\endcsname[3][1,3,
usedefault]{%
7843       \doedindexlabel%
7844       \csname#1\endcsname[counter=pageline,##1]{##2}[##3]%
7845     }%
7846     \expandafter\WithSuffix\expandafter\DeclareRobustCommand\csname ed
#1\endcsname*[3][1,3,usedefault]{%
7847       \doedindexlabel%
7848       \csname#1\endcsname*[counter=pageline,##1]{##2}[##3]%
7849     }%
7850   }%
7851   \docsvlist{%
7852     gls,%
7853     Gls,%
7854     GLS,%
7855     glspl,%
7856     Glspl,%
7857     GLSpl,%
7858     glstext,%
7859     Glstext,%
7860     GLStext,%
7861     Glsfirst,%

```

```

7862 GLSfirst,%
7863 glsplural%
7864 Glsplural,%
7865 GLSplural,%
7866 glsfirstplural,%
7867 Glsfirstplural,%
7868 GLSfirstplural,%
7869 glsname,%
7870 Glsname,%
7871 GLSname,%
7872 glssymbol,%
7873 Glsymbol,%
7874 GLSymbol,%
7875 glsdesc,%
7876 Glsdesc,%
7877 GLSdesc,%
7878 glsuseri,%
7879 Glsuseri,%
7880 GLSuseri,%
7881 glsuserii,%
7882 Glsuserii,%
7883 GLSuserii,%
7884 glsuseriii,%
7885 Glsuseriii,%
7886 GLSuseriii,%
7887 glsuseriv,%
7888 Glsuseriv,%
7889 GLSuseriv,%
7890 glsuserv,%
7891 Glsuserv,%
7892 GLSuserv,%
7893 glsuservi,%
7894 Glsuservi,%
7895 GLSuservi%
7896 }%
7897 %

```

Then those which arguments are [*options*]{*label*}{*link text*}.

```

7898 \renewcommand{\do}[1]{%
7899 \expandafter\DeclareRobustCommandx\csname ed#1\endcsname[3][1,
usedefault]{%
7900 \doedindexlabel%
7901 \csname#1\endcsname[counter=pageline,##1]{##2}{##3}%
7902 }%
7903 \expandafter\WithSuffix\expandafter\DeclareRobustCommandx\csname ed
#1\endcsname*[3][1,usedefault]{%
7904 \doedindexlabel%
7905 \csname#1\endcsname*[counter=pageline,##1]{##2}{##3}%
7906 }%
7907 }%

```

```

7908 \docsvlist{glsdisp,glslink}%
7909 %
Then those which arguments are [\langle options \rangle]{\langle label \rangle}.
7910 \renewcommand{\do}[1]{%
7911 \expandafter\DeclareRobustCommand\csname ed#1\endcsname[2][1,
usedefault]{%
7912 \doedindexlabel%
7913 \csname#1\endcsname[counter=pageline,##1]{##2}%
7914 }%
7915 \expandafter\WithSuffix\expandafter\DeclareRobustCommand\csname ed
#1\endcsname*[2][1,usedefault]{%
7916 \doedindexlabel%
7917 \csname#1\endcsname*[counter=pageline,##1]{##2}%
7918 }%
7919 }%
7920 \docsvlist{glsadd}%
7921 }{}%
7922 }%
7923 %

```

## 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`.

`\ifinstanza` For example, in French typography, the symbol is ' '. We obtain it by means of the following code:

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

The `\ifinstanza` boolean is used to ensure that we are in a stanza part.

```

7924 \def\@hangingsymbol{}
7925 \newcommand*\sethangingsymbol[1]{%
7926 \gdef\@hangingsymbol{#1}%
7927 }%
7928 \newif\ifinstanza
7929 %

```

`\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.

```

7930 \newif\ifinserthangingsymbol
7931 \newcommand{\inserthangingsymbol}{%
7932 \ifinserthangingsymbol%
7933   \ifinstanza%
7934     \@hangingsymbol%
7935   \fi%
7936 \fi%
7937 }
7938 %

```

### 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.

```

7939 \newcommand*{\ampersand}{\char`\&}
7940
7941 %

```

### 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.

```

7942 \chardef\body=\catcode`\@
7943 \catcode`\@=11
7944 \chardef\next=\catcode`\&
7945 \catcode`\&=\active
7946
7947 %

```

### 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.

```

7948 \newcount\stanza@count
7949 \newlength{\stanzaindentbase}
7950 \setlength{\stanzaindentbase}{20pt}
7951
7952 %

```

**\strip@szacnt** The indentations of stanza lines are non-negative integer multiples of the unit called  
**\setstanzavalues** `\stanzaindentbase`. To make it easier for the user to specify these numbers, some list macros are defined. These take numerical values in a list separated by commas and assign the values to special control sequences using `\mathchardef`. Though this does

limit the range from 0 to 32767, it should suffice for most applications, including *penalties*, which will be discussed below.

```

7953 \def\strip@szacnt#1,#2|{\def\@tempb{#1}\def\@tempa{#2|}}
7954 \newcommand*\setstanzavalues[2]{\def\@tempa{#2,,|}%
7955   \stanza@count\z@
7956   \def\next{\expandafter\strip@szacnt\@tempa
7957     \ifx\@tempb\empty\let\next\relax\else
7958     \expandafter\mathchardef\csname #1@number\stanza@count
7959     \@endcsname\@tempb\relax
7960     \advance\stanza@count\@ne\fi\next}%
7961   \next}
7962
7963 %

```

**\setstanzaindents** In the original edmac, `\setstanzavalues{sza}{\langle...⟩}` had to be called to set the indents, and similarly `\setstanzavalues{szp}{\langle...⟩}` 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).

```

7964 \newcommand*\setstanzaindents[1]{\setstanzavalues{sza}{#1}}
7965 \newcommand*\setstanzapenalties[1]{\setstanzavalues{szp}{#1}}
7966 %
7967 %

```

**\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.

```

7968 \newcounter{stanzaindentsrepetition}
7969 \newcount\stanza@modulo
7970
7971 \newcommand*\managestanza@modulo[0]{%
7972   \advance\stanza@modulo\@ne%
7973   \ifnum\stanza@modulo>\value{stanzaindentsrepetition}%
7974     \stanza@modulo\@ne%
7975   \fi%
7976 }
7977 %

```

**\stanzaindent** The macro `\stanzaindent`, when called at the beginning of a verse, changes the indentation normally defined for this verse by `\setstanzaindent`. The starred version skips the current verse for the repetition of stanza indent.

```

7978 \newcommand*\stanzaindent[1]{%
7979   \hspace{\dimexpr#1\stanzaindentbase-\parindent\relax}%
7980   \ignorespaces%
7981 }%

```

```

7982 \WithSuffix\newcommand\stanzaindent*[1]{%
7983   \stanzaindent{#1}%
7984   \global\advance\stanza@modulo-\@ne%
7985   \ifnum\stanza@modulo=0%
7986     \global\stanza@modulo=\value{stanzaindentsrepetition}%
7987   \fi%
7988   \ignorespaces%
7989 }%
7990 %

```

### XXX.5 Numbering stanza

Here, macro for numbering stanza. First, the stanza counter.

```

\thestanzaa91 \newcounter{stanza}
7992 \renewcommand{\thestanza}{%
7993   \textbf{\arabic{stanza}}%
7994 }
7995 %

```

`\ifnumberstanza` Then, macro to activate automatically numbering of stanza.

```

7996 \newif\ifnumberstanza%
7997 %

```

`\@insertstanzanumber` Now, macro called at the first line of of verse of a stanza.

```

7998 \newcommand{\@insertstanzanumber}[0]{%
7999   \ifnumberstanza%
8000     \ifl@dpairing%
8001       \ifledRcol%
8002         \stanzanumwrapper{\thestanzaR}%
8003       \else%
8004         \stanzanumwrapper{\thestanzaL}%
8005       \fi%
8006     \else%
8007       \stanzanumwrapper{\thestanza}%
8008     \fi%
8009     \setline{1}%
8010   \fi%
8011 }%
8012 %

```

`\@advancestanzanumber` Also a command to advance the counter of stanza.

```

8013 \newcommand{\@advancestanzanumber}[0]{%
8014   \ifnumberstanza%
8015     \ifl@dpairing%
8016       \ifledRcol%

```

```

8017     \addtocounter{stanzaR}{1}%
8018     \else%
8019     \addtocounter{stanzaL}{1}%
8020     \fi%
8021     \else%
8022     \addtocounter{stanza}{1}%
8023     \fi%
8024     \fi%
8025 }%
8026 %

```

`\stanzanumwrapper` And finally, the wrapper for stanza number

```

8027 \newcommand{\stanzanumwrapper}[1]{%
8028     \flagstanza{#1}%
8029 }%
8030 %

```

### XXX.6 Stanza number in note

Here, the command called when printing stanza number in notes.

```

8031 \newcommand{\printstanza}[0]{%
8032     \ifboolexpr{bool{l@dpairing} or bool{l@dprintingpages} or bool{
8033         l@dprintingcolumns}}{%
8034         \ifledRcol{%
8035             \thestanzaR%
8036         \else%
8037             \thestanzaL%
8038         \fi%
8039     }{%
8040         \thestanza%
8041     }%
8042 }%

```

### 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.

```

8043 \newcommandx{\stanza@line}[2][1,2,usedefault]{%
8044     \ifnum\value{stanzaindentrepetition}=0

```



```

8045     \ifcsdef{sza@\number\stanza@count @}{%
8046         {%
8047             \parindent=\csname sza@\number\stanza@count @\endcsname\
stanzaindentbase%
8048         }{%
8049             \led@err@StanzaIndentNotDefined%
8050         }%
8051     \else
8052         \ifcsdef{sza@\number\stanza@modulo @}{%
8053             \parindent=\csname sza@\number\stanza@modulo @\endcsname\
stanzaindentbase%
8054             \managestanza@modulo%
8055         }%
8056         {%
8057             \led@err@StanzaIndentNotDefined%
8058         }%
8059     \fi
8060     \pstart[#1][#2]\stanza@hang\ignorespaces%
8061 }%
8062 \xdef\stanza@hang{\noexpand\leavevmode\noexpand\startlock
8063     \hangindent\expandafter
8064     \noexpand\csname sza@0@\endcsname\stanzaindentbase
8065     \hangafter\@ne}
8066 \def\sza@penalty{\count@\csname szp@\number\stanza@count @\endcsname
8067     \ifnum\count@>\@M\advance\count@-\@M\penalty-\else
8068     \penalty\fi\count@}
8069 %

```

\@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  
                  \AtEveryStanza    desired, invoke \let\startlock\relax and do the same for \endlock. Here and  
 \AtStartEveryStanza    above we have used \xdef to make the stored macros take up a bit less space, but it also  
 \BeforeEveryStopStanza    makes them more obscure to the reader. Lines of the stanza are delimited by ampersands  
                  \newverse    &. The last line of the stanza must end with \&.

```

8070 \xdef\@startstanza[#1][#2]{%
8071     \noexpand\instanzatrue\expandafter
8072     \begingroup%
8073     \catcode`\noexpand\&\active%
8074     \global\stanza@count\@ne\stanza@modulo\@ne
8075     \noexpand\ifnum\expandafter\noexpand
8076     \csname sza@0@\endcsname=\z@\let\noexpand\stanza@hang\relax
8077     \let\noexpand\endlock\relax\noexpand\else\interlinepenalty
8078     \@M\rightskip\z@ plus 1fil\relax\noexpand\fi\noexpand\ifnum
8079     \expandafter\noexpand\csname szp@0@\endcsname=\z@
8080     \let\noexpand\sza@penalty\relax\noexpand\fi%
8081     \def\noexpand&{%

```

```

8082 \noexpand\newverse[] []}%
8083 \def\noexpand\&\noexpand\@stopstanza}%
8084 \noexpand\@advancestanza\advance\stanza\advance\stanza}%
8085 \noexpand\stanza@line[#1] [#2]}%
8086 \noexpand\@insertstanza\advance\stanza\advance\stanza}%
8087 \let\par\relax\ignorespaces%No paragraph in verses
8088 }
8089
8090 \newcommandx{\stanza}[2][1,2,usedefault]{%
8091 \ifboolexpr{%
8092 not test{\ifdefvoid{\at@every@stanza}}}%
8093 and test{\ifstrempy{#1}}}%
8094 and test{\ifstrempy{#2}}}%
8095 {\@startstanza[] [\at@every@stanza] \at@start@every@stanza}%
8096 {\@startstanza[#1] [#2] \at@start@every@stanza}%
8097 }%
8098
8099 \newcommandx{\@stopstanza}[2][1,2,usedefault]{%
8100 \unskip%
8101 \endlock%
8102 \ifboolexpr{%
8103 not test{\ifdefvoid{\at@every@stop@stanza}}}%
8104 and test{\ifstrempy{#1}}}%
8105 and test{\ifstrempy{#2}}}%
8106 {\before@every@stop@stanza\pend[] [\at@every@stop@stanza]}%
8107 {\before@every@stop@stanza\pend[#1] [#2]}%
8108 \endgroup%
8109 \instanzafalse%
8110 }
8111
8112 \newcommand{\AtEveryStopStanza}[1]{%
8113 \ifstrempy{#1}%
8114 {\gdef\at@every@stop@stanza{}}%
8115 {\gdef\at@every@stop@stanza{\noindent#1}}%
8116 }%
8117 \WithSuffix\newcommand\AtEveryStopStanza*[1]{%
8118 \ifstrempy{#1}%
8119 {\gdef\at@every@stop@stanza{}}%
8120 {\gdef\at@every@stop@stanza{#1}}%
8121 }%
8122 \def\at@every@stop@stanza{}%
8123
8124 \newcommand{\AtEveryStanza}[1]{%
8125 \ifstrempy{#1}%
8126 {\gdef\at@every@stanza{}}%
8127 {\gdef\at@every@stanza{\noindent#1}}%
8128 }%
8129 \WithSuffix\newcommand\AtEveryStanza*[1]{%
8130 \ifstrempy{#1}%
8131 {\gdef\at@every@stanza{}}%

```

```

8132 {\gdef\at@every@stanza{#1}}%
8133 }%
8134
8135
8136
8137 \newcommand{\AtStartEveryStanza}[1]{%
8138   \ifstrempy{#1}%
8139   {\gdef\at@start@every@stanza{}}%
8140   {\gdef\at@start@every@stanza{#1}}%
8141 }%
8142 \def\at@start@every@stanza{}%
8143
8144 \newcommand{\BeforeEveryStopStanza}[1]{%
8145   \ifstrempy{#1}%
8146   {\gdef\before@every@stop@stanza{}}%
8147   {\gdef\before@every@stop@stanza{#1}}%
8148 }%
8149 \def\before@every@stop@stanza{}%
8150
8151 \newcommand*\newverse}[4][1,2,3,4,usedefault]{%
8152   \unskip%
8153   \endlock\pend[#1][#3]\sza@penalty\global%
8154   \advance\stanza@count\@ne\stanza@line[#2][#4]%
8155 }
8156
8157 %

```

**\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`.

```

8158 \newcommand*\flagstanza}[2][\stanzaindentbase]{%
8159   \hskip -#1\llap{#2}\hskip #1\ignorespaces}
8160
8161 %

```

## 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.

```

8162 \catcode`\&=\next
8163 \catcode`\@=\body
8164 \setstanzavalues{szp}{0}
8165
8166 %

```

## 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
\msdata@cR
8167 \def\msdata@c{}%
8168 \def\msdata@cR{}%
8169 \newcommand{\msdata}[1]{%
8170   \leavevmode%
8171   \unless\ifstopmsdata@inserted%
8172     \stopmsdata%
8173     \led@warning@msdatawithoutstop%
8174   \fi%
8175   \global\stopmsdata@inserted@false%
8176   \unless\ifledRcol%
8177     \numgdef{\msdata@c}{\msdata@c+1}%
8178     \ifdef{\hypertarget}{%
8179       \edlabel{\msdata@c:start:msdata}%
8180     }{}%
8181     \protected@write\linenum@out{}{%
8182       \string\@msd{#1}%
8183     }%
8184   \else%
8185     \numgdef{\msdata@cR}{\msdata@cR+1}%
8186     \ifdef{\hypertarget}{%
8187       \edlabel{\msdata@cR:start:msdata}%
8188     }{}%
8189     \protected@write\linenum@outR{}{%
8190       \string\@msd{#1}%
8191     }%
8192   \fi%
8193 }%
8194 %

```

**\stopmsdata** The user level **\stopmsdata** command only writes information about the end of manuscripts data in numbered auxiliary file.

```

8195 \newcommand{\stopmsdata}[0]{%
8196   \leavevmode%
8197   \unless\ifledRcol%
8198     \protected@write\linenum@out{}{%
8199       \string\@stopmsd%
8200     }%
8201     \ifdef{\hypertarget}{%
8202       \edlabel{\msdata@c:end:msdata}%
8203     }{}%
8204   \else%
8205     \protected@write\linenum@outR{}{%

```

```

8206 \string\@stopmsd%
8207 }%
8208 \ifdef{\hypertarget}{%
8209 \edlabel{\msdata@cR:end:msdata}%
8210 }{}%
8211 \fi%
8212 \global\stopmsdata@inserted@true%
8213 }%
8214 %

```

**\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`

```

8215 \newif\ifstopmsdata@inserted@%
8216 %

```

## 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?

```

8217 \newcommand{\setmsdataseries}[1]{%
8218 \gdef\@msdata@series{#1}%
8219 }%
8220 \def\@msdata@series{A}%
8221 %

```

**\setmsdataposition** The label for the manuscripts data.

```

8222 \def\ms@data@position{msdata-regular}%
8223 \newcommand{\setmsdataposition}[1]{%
8224 \gdef\ms@data@position{#1}%
8225 }%
8226 %

```

**\setmsdatalabel** The label for the manuscripts data.

```

8227 \def\ms@data@label{Ms.}%
8228 \newcommand{\setmsdatalabel}[1]{%
8229 \gdef\ms@data@label{#1}%
8230 }%
8231 %

```

### XXXI.3 Counters and lists

**\@msd@c** \@msd@c is a counter incremented at each \@msd read in auxiliary file.

```
8232 \numdef{\@msd@c}{0}
8233 \numdef{\@msd@cR}{0}
8234 %
```

**\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.

```
8235 \numdef{\add@msd@c}{0}%
8236 \numdef{\add@msd@cR}{0}%
8237 %
```

**\@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.

```
8238 \def\@msdata@list{}%
8239 %
```

### 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.

```
8240 \newcommand{\@msd}[1]{%
8241   \unless\ifledRcol%
8242     \global\numdef{\@msd@c}{\@msd@c+\@ne}%
8243     \csgdef{\@msdata@\@msd@c @data}{#1}%
8244     \csxdef{\@msdata@\@msd@c @linenumber}{\the\line@num}%
8245     \csxdef{\@msdata@\@msd@c @abslinenumber}{\the\absline@num}%
8246     \xright@appenditem{\the\absline@num}\to\actionlines@list%
8247     \xright@appenditem{-1010}\to\actions@list%
8248   \else%
8249     \global\numdef{\@msd@cR}{\@msd@cR+\@ne}%
8250     \csgdef{\@msdata@\@msd@cR @dataR}{#1}%
8251     \csxdef{\@msdata@\@msd@cR @linenumberR}{\the\line@numR}%
8252     \csxdef{\@msdata@\@msd@cR @abslinenumberR}{\the\absline@numR}%
8253     \xright@appenditem{\the\absline@numR}\to\actionlines@listR%
8254     \xright@appenditem{-1010}\to\actions@listR%
8255   \fi%
8256 }%
8257 %
```

**\@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.

```

8258 \newcommand{\@stopmsd}[0]{%
8259   \unless\ifledRcol%
8260     \ifcsundef{@msdata@\@msd@c @stoplinenumber}{%
8261       \csxdef{@msdata@\@msd@c @stopabslinenumber}{\the\absline@num}%
8262       \csxdef{@msdata@\@msd@c @stoplinenumber}{\the\line@num}%
8263     }{}%
8264   \else%
8265     \ifcsundef{@msdata@\@msd@cR @stoplinenumberR}{%
8266       \csxdef{@msdata@\@msd@cR @stopabslinenumberR}{\the\absline@numR}%
8267       \csxdef{@msdata@\@msd@cR @stoplinenumberR}{\the\line@numR}%
8268     }%
8269   }{}%
8270   \fi%
8271 }%
8272 %

```

### 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.

```

8273 \newcommand{\add@msdata}{%
8274   \bgroup%
8275   \normalfont%
8276   \unless\ifledRcol%
8277     \numgdef{\add@msd@c}{\add@msd@c+\@one}%
8278     \ifcsdef{@msdata@\add@msd@c @data}{%
8279       \letcs{\@data}{@msdata@\add@msd@c @data}%
8280       \edef\l@d@nums{%
8281         000| % Start page = we don't print it
8282         \csuse{@msdata@\add@msd@c @linenumber}| % Start line number
8283         000| % Start subline number, for now, not used
8284         000| % End page number, we don't print it
8285         \ifnumless{\csuse{@msdata@\add@msd@c @stopabslinenumber}}{\csuse{
@lastabsline@forpage@\the\page@num}}}%
8286         {\csuse{@msdata@\add@msd@c @stoplinenumber}}}% End line number if
in the same page
8287         {\csuse{@lastline@forpage@\the\page@num}}}% Otherwiser, last
number of the page
8288       |%
8289       000| % End sub line number, for now, not used
8290       \edfont@info%Font
8291     }%
8292     \@msd@options@fullpagefalse%
8293     \if@firstlineofpage% Try if the data are for the full page. If yes
, will add options to the list.
8294     \unless\if@msdata@insertedfrompreviouspage%

```

```

8295 \ifnumless{\csuse{@lastabsline@forpage@the\page@num}}{\csuse
      {\@msdata@add@msd@c @stopabslinenumber}+\@ne}%
8296 {%
8297 \numdef{\@tmp}{\add@msd@c+\@ne}%
8298 \ifcsdef{\@msdata@\@tmp @abslinenumber}%
8299 {\ifnumequal{\csuse{\@msdata@\@tmp @abslinenumber}}{\csuse{
      \@lastabsline@forpage@the\page@num}}}%
8300 {}%
8301 {\@msd@options@fullpagetrue}%
8302 }%
8303 {\@msd@options@fullpagetrue}%
8304 }%
8305 {}%
8306 \fi%
8307 \fi%
8308 \listxadd{\@msdata@list}{%
8309 \@msd@options@iffullpage%
8310 \ifluatex%
8311 \csxdef{footnote@luatextextdir}{the\textdir}%
8312 \csxdef{footnote@luatexpardir}{the\pardir}%
8313 \fi%
8314 \csdef{@this@crossref@start}{\add@msd@c:start:msdata}%
8315 \csdef{@this@crossref@end}{\add@msd@c:end:msdata}%
8316 \noexpand\csuse{v\@msdata@series footnote}{\@msdata@series}{\@
      expandonce\l@d@nums}{\ms@data@label}{\expandonce\data}}%
8317 \reset@msd@options@iffullpage%
8318 }%
8319 }%
8320 {}%
8321 \else%
8322 \numgdef{\add@msd@cR}{\add@msd@cR+\@ne}%
8323 \ifcsdef{\@msdata@\add@msd@cR @dataR}{%
8324 \letcs{\@data}{\@msdata@\add@msd@cR @dataR}%
8325 \edef\l@d@nums{%
8326 000| % Start page = we don't print it
8327 \csuse{\@msdata@\add@msd@cR @linenumberR}| % Start line number
8328 000| % Start subline number, for now, not used
8329 000| % End page number, we don't print it
8330 \ifnumless{\csuse{\@msdata@\add@msd@cR @stopabslinenumberR}}{\@
      csuse{@lastline@forpageR@the\page@numR}}%
8331 {\csuse{\@msdata@\add@msd@cR @stoplinenumberR}}%End line number
      if in the same page
8332 {\csuse{@lastline@forpageR@the\page@numR}}%Otherwiser, last
      number of the page
8333 |%
8334 000| % End sub line number, for now, not used
8335 \edfont@info%Font
8336 }%
8337 \@msd@options@fullpagefalse%
8338 \if@firstlineofpageR%

```



```

8339         \unless\if@msdata@insertedfrompreviouspage%
8340             \ifnumless{\csuse{@lastabsline@forpageR@the\page@numR}}{\
csuse{@msdata@add@msd@c @stopabslinenumberR}+\@ne}%
8341             {%
8342                 \numdef{\@tmp}{\add@msd@cR+\@ne}%
8343                 \ifcsdef{@msdata@\@tmp @abslinenumberR}%
8344                     {\ifnumequal{\csuse{@msdata@\@tmp @abslinenumberR}}{\csuse{
@lastabsline@forpageR@the\page@numR}}}%
8345                     {}%
8346                     {\@msd@options@fullpagetrue}%
8347                 }%
8348                 {\@msd@options@fullpagetrue}%
8349             }%
8350             {}%
8351         \fi%
8352     \fi%
8353     \listxadd{\@msdata@list}{%
8354         \@msd@options@iffullpage%
8355         \ifluatex%
8356             \csxdef{footnote@luatextextdir}{\the\textdir}%
8357             \csxdef{footnote@luatexpardir}{\the\pardir}%
8358         \fi%
8359         \csdef{@this@crossref@start}{\add@msd@cR:start:msdata}%
8360         \csdef{@this@crossref@end}{\add@msd@cR:end:msdata}%
8361         \noexpand\csuse{v\@msdata@series footnote}{\@msdata@series}{\
expandonce\l@d@nums}{\ms@data@label}{\expandonce\@data}}%
8362         \reset@msd@options@iffullpage%
8363     }%
8364     }%
8365     {}%
8366     \fi%
8367     \egroup%
8368 }%
8369 %

```

**\@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.

```

8370 \newif\if@msdata@insertedfrompreviouspage%
8371 %

```

**\@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.

```

8372 \newcommand{\add@msdata@firstlineofpage}{%
8373     \bgroup%
8374     \normalfont%
8375     \unless\ifledRcol@%
8376         \ifcsdef{@msdata@\add@msd@c @data}{%

```

```

8377 \ifnumless{\the\absline@num-\@ne}{\csuse{@msdata@\add@msd@c
@stopabslinenumber}}}%
8378 {%
8379 \global\@msdata@insertedfrompreviouspagetrue%
8380 \letcs{\@data}{@msdata@\add@msd@c \@data}%
8381 \edef\l@d@nums{%
8382 000|% Start page = we don't print it
8383 \numexpr\the\line@num+\@ne\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
8384 000|% Start subline number, for now, not used
8385 000|% End page number, we don't print it
8386 \ifnumless{\csuse{@msdata@\add@msd@c @stopabslinenumber}}{\csuse{
@lastabsline@forpage@\the\page@num}}}%
8387 {\csuse{@msdata@\add@msd@c @stoplinenumber}}}%End line number if
in the same page
8388 {\csuse{@lastline@forpage@\the\page@num}}}%Otherwise, last
number of the page
8389 |%
8390 000|% End sub line number, for now, not used
8391 \edfont@info%Font
8392 }%
8393 \@msd@options@fullpagefalse%
8394 \ifnumless{\csuse{@lastabsline@forpage@\the\page@num}}{\csuse{
@msdata@\add@msd@c @stopabslinenumber}+\@ne}%We will test if the ms data is
for the full page
8395 {%
8396 \numdef{\@tmp}{\add@msd@c+\@ne}%
8397 \ifcsdef{@msdata@\@tmp @abslinenumber}%
8398 {\ifnumequal{\csuse{@msdata@\@tmp @abslinenumber}}{\csuse{
@lastabsline@forpage@\the\page@num}}}%
8399 {}%
8400 {\@msd@options@fullpagetrue}%
8401 }%
8402 {\@msd@options@fullpagetrue}%
8403 }%
8404 {}%
8405 \listxadd{\@msdata@list}{%
8406 \@msd@options@iffullpage%
8407 \ifluatex%
8408 \csxdef{footnote@luatextextdir}{\the\textdir}%
8409 \csxdef{footnote@luatexpardir}{\the\pardir}%
8410 \fi%
8411 \csdef{@this@crossref@start}{\add@msd@c:start:msdata}%
8412 \csdef{@this@crossref@end}{\add@msd@c:end:msdata}%
8413 \noexpand\csuse{v\@msdata@series footnote}{\@msdata@series}{\{
expandonce\l@d@nums}{\msdata@label}{\expandonce\@data}}}%
8414 \reset@msd@options@iffullpage%
8415 }%
8416 }%

```

```

8417     {\global\@msdata@insertedfrompreviouspagefalse}%
8418   }{}%
8419   \else%
8420     \ifcsdef{@msdata@\add@msd@cR @dataR}{%
8421       \ifnumless{\the\absline@numR-\@ne}{\csuse{@msdata@\add@msd@cR
@stopabslinenumberR}}%
8422       {%
8423         \global\@msdata@insertedfrompreviouspagetrue%
8424         \letcs{\@data}{@msdata@\add@msd@cR @dataR}%
8425         \edef\l@d@nums{%
8426           000}% Start page = we don't print it
8427         \numexpr\the\line@numR+\@ne\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
8428         000}% Start subline number, for now, not used
8429         000}% End page number, we don't print it
8430         \ifnumless{\csuse{@msdata@\add@msd@cR @stopabslinenumberR}}{\
csuse{@lastline@forpageR@\the\page@numR}}%
8431         {\csuse{@msdata@\add@msd@cR @stoplinenumberR}}}%End line number
if in the same page
8432         {\csuse{@lastline@forpageR@\the\page@numR}}}%Otherwise, last
number of the page
8433         |%
8434         000}% End sub line number, for now, not used
8435         \edfont@info%Font
8436       }%
8437       \@msd@options@fullpagefalse%
8438       \ifnumless{\csuse{@lastabsline@forpageR@\the\page@numR}}{\csuse{
@msdata@\add@msd@cR @stopabslinenumberR}+\@ne}%
8439       {%
8440         \numdef{\@tmp}{\add@msd@cR+\@ne}%
8441         \ifcsdef{@msdata@\@tmp @abslinenumberR}%
8442         {\ifnumequal{\csuse{@msdata@\@tmp @abslinenumberR}}{\csuse{
@lastabsline@forpageR@\the\page@numR}}}%
8443         {}%
8444         {\@msd@options@fullpagetrue}%
8445       }%
8446       {\@msd@options@fullpagetrue}%
8447     }%
8448   {}%
8449   \listxadd{\@msdata@list}{%
8450     \@msd@options@iffullpage%
8451     \ifluatex%
8452       \csxdef{footnote@luatextextdir}{\the\textdir}%
8453       \csxdef{footnote@luatexpardir}{\the\pardir}%
8454     \fi%
8455     \csdef{@this@crossref@start}{\add@msd@cR:start:msdata}%
8456     \csdef{@this@crossref@end}{\add@msd@cR:end:msdata}%
8457     \noexpand\csuse{v@msdata@series footnote}{\@msdata@series}{\
expandonce\l@d@nums}{\ms@data@label}{\expandonce\@data}}%

```

```

8458         \reset@msd@options@iffullpage%
8459     }%
8460 }%
8461 {\global\@msdata@insertedfrompreviouspagefalse}%
8462 }{}%
8463 \fi%
8464 \egroup%
8465 }%
8466 %

```

### 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@msdata67 \newcommand{\insert@msdata}{%
8468     \def\do##1{##1}%
8469     \dolistloop{\@msdata@list}%
8470     \global\let\@msdata@list\relax%
8471 }%
8472 %

```

### 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.

```

8473 \newif\if@msd@options@fullpage%
8474 \newcommand{\@msd@options@iffullpage}[0]{%
8475     \if@msd@options@fullpage%
8476         \noexpand\toggletrue{nonum@}%
8477         \ifdefvoid{\ms@data@label}%
8478             {\noexpand\toggletrue{nosep@}}%
8479         {}%
8480     \fi%
8481 }%
8482 \newcommand{\reset@msd@options@iffullpage}[0]{%
8483     \noexpand\togglefalse{nonum@}%
8484     \noexpand\togglefalse{nosep@}%
8485 }%
8486 %

```

## 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 ‘`eq and amstex`’, 1995/08/31, started by Keith Reckdahl and ended definitively by David M. Jones.

Several of the `[math]` macros scan their body twice. This means we must collect all text in the body of an environment form before calling the macro.

`\@emptytoks` This is actually defined in the `amsgen` package.

```
8487 \newtoks\@emptytoks
8488
8489 %
```

The rest is from `amsmath`.

`\l@denbody` A token register to contain the body.

```
8490 \newtoks\l@denbody
8491
8492 %
```

`\addtol@denbody` `\addtol@denbody{arg}` adds `arg` to the token register `\l@denbody`.

```
8493 \newcommand{\addtol@denbody}[1]{%
8494   \global\l@denbody\expandafter{\the\l@denbody#1}}
8495
8496 %
```

`\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`.

```
8497 \newcommand{\l@dcollect@body}[1]{%
8498   \l@denbody{\expandafter#1\expandafter{\the\l@denbody}}%
8499   \edef\processl@denbody{\the\l@denbody\noexpand\end{\@currentenv}}%
8500   \l@denbody\@emptytoks \def\l@dbegin@stack{b}%
8501   \begingroup
8502     \expandafter\let\csname\@currentenv\endcsname\l@dcollect@body
8503     \edef\processl@denbody{\expandafter\noexpand\csname\@currentenv\endcsname}%
8504     \processl@denbody%
8505   }%
8506
8507 %
```

`\l@dpush@begins` When adding a piece of the current environment's contents to `\l@denvbody`, we scan it to check for additional `\begin` tokens, and add a 'b' to the stack for any that we find.

```
8508 \def\l@dpush@begins#1\begin#2{%
8509   \ifx\end#2\else b\expandafter\l@dpush@begins\fi}
8510
8511 %
```

`\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.

```
8512 \def\l@dcollect@@body#1\end#2{%
8513   \edef\l@dbegin@stack{\l@dpush@begins#1\begin\end
8514     \expandafter\@gobble\l@dbegin@stack}%
8515   \ifx\@empty\l@dbegin@stack
8516     \endgroup
8517     \@checkend{#2}%
8518     \addtol@denvbody{#1}%
8519   \else
8520     \addtol@denvbody{#1\end{#2}}%
8521   \fi
8522   \processl@denvbody % A little tricky! Note the grouping
8523 }
8524
8525 %
```

There was a question on CTT about how to use `\collect@body` for a macro taking an argument. The following is part of that thread.

From: Heiko Oberdiek <oberdiek@uni-freiburg.de>  
 Newsgroups: comp.text.tex  
 Subject: Re: Using `\collect@body` with commands that take >1 argument  
 Date: Fri, 08 Aug 2003 09:03:20 +0200

eed132@psu.edu (Evan) wrote:  
 > I'm trying to make a new Latex environment that acts like the  
 > `\colorbox` command that is part of the color package. I looked through  
 > the FAQ and ran across this bit about using the `\collect@body` command  
 > that is part of AMSLaTeX:  
 > <http://www.tex.ac.uk/cgi-bin/texfaq2html?label=cmdasenv>  
 >  
 > It almost works. If I do something like the following:  
 > `\newcommand{\redbox}[1]{\colorbox{red}{#1}}`  
 >  
 > `\makeatletter`  
 > `\newenvironment{redbox}{\collect@body \redbox}{}`

You will get an error message: Command `\redbox` already defined.  
Thus you must rename either the command `\redbox` or the environment name.

```
> \begin{coloredbox}{blue}
>   Yadda yadda yadda... this is on a blue background...
> \end{coloredbox}
> and can't figure out how to make the \collect@body take this.

> \collect@body \colorbox{red}
> \collect@body {\colorbox{red}}
```

The argument of `\collect@body` has to be one token exactly.

```
\documentclass{article}
\usepackage{color}
\usepackage{amsmath}

\newcommand{\redbox}[1]{\colorbox{red}{#1}}
\makeatletter
\newenvironment{coloredbox}[1]{%
  \def\next@{\colorbox{#1}}%
  \collect@body\next@
}{%

% ignore spaces at begin and end of environment
\newenvironment{coloredboxII}[1]{%
  \def\next@{\mycoloredbox{#1}}%
  \collect@body\next@
}{%
\newcommand{\mycoloredbox}[2]{%
  \colorbox{#1}{\ignorespaces#2\unskip}%
}

% support of optional color model argument
\newcommand\coloredboxIII\endcsname{}
\def\coloredboxIII#1#2{%
  \@coloredboxIII{#1}%
}
\def\@coloredboxIII#1#2{%
  \def\next@{\mycoloredboxIII{#1}{#2}}%
  \collect@body\next@
}
\newcommand{\mycoloredboxIII}[3]{%
  \colorbox{#1}{#2}{\ignorespaces#3\unskip}%
}

\makeatother
```

```

\begin{document}
  Black text before
  \begin{coloredbox}{blue}
    Hello World
  \end{coloredbox}
  Black text after

  Black text before
  \begin{coloredboxII}{blue}
    Hello World
  \end{coloredboxII}
  Black text after

  Black text before
  \begin{coloredboxIII}[rgb]{0,0,1}
    Hello World
  \end{coloredboxIII}
  Black text after

\end{document}

Yours sincerely
Heiko <oberdiek@uni-freiburg.de>

```

## XXXII.2 Tabular environments

This is based on the work by Herbert Breger in developing `tabmac.tex`.

The original `tabmac.tex` file was void of comments or any explanatory text other than the above notice. The algorithm is Breger's. Peter Wilson have made some cosmetic changes to the original code and reimplemented some things so they are more LaTeX-like. All the commentary are from Peter Wilson, as are any mistake or errors.

However, Maïeul Rouquette has modified code in order to add new features of `eledmac` and `reledmac`.

### XXXII.2.1 Disabling and restoring commands

`\l@dtabnoexpands` More no expansion for critical and familiar footnotes in tabular environment.

```

8526 \newcommand*{\l@dtabnoexpands}{%
8527   \let\rtab=0%
8528   \let\ctab=0%
8529   \let\ltab=0%
8530   \let\rtabtext=0%
8531   \let\ltabtext=0%
8532   \let\ctabtext=0%
8533   \let\edbeforetab=0%
8534   \let\edaftertab=0%
8535   \let\edatleft=0%

```



```

8536 \let\edatright=0%
8537 \let\edvertline=0%
8538 \let\edvertdots=0%
8539 \let\edrowfill=0%
8540 }
8541
8542 %

```

**\disable@familiarnotes** Macros to disable and restore familiar notes, to prevent them from printing multiple times in edtabularx and edarrayx environments.

**\restore@familiarnotes**

```

8543 \newcommand{\disable@familiarnotes}{%
8544 \unless\ifnofamiliar@%
8545 \def\do##1{%
8546 \csletcs{footnote@@##1}{footnote##1}%
8547 \expandafter\renewcommand \csname footnote##1\endcsname[1]{%
8548 \protected@csxdef{@thefnmark##1}{\csuse{thefootnote##1}}%
8549 \csuse{@footnotemark##1}%
8550 }%
8551 }%
8552 \dolistloop{\@series}%
8553 \fi%
8554 }%
8555 \newcommand{\restore@familiarnotes}{%
8556 \unless\ifnofamiliar@%
8557 \def\do##1{%
8558 \csletcs{footnote##1}{footnote@@##1}%
8559 }%
8560 \dolistloop{\@series}%
8561 \fi%
8562 }%
8563
8564 %

```

**\disable@sidenotes** The same for sidenotes.

**\restore@sidenotes**

```

8565 \newcommand{\disable@sidenotes}{%
8566 \let\@@ledrightnote\ledrightnote%
8567 \let\@@ledleftnote\ledleftnote%
8568 \let\@@ledsidenote\ledsidenote%
8569 \let\ledrightnote\@gobble%
8570 \let\ledleftnote\@gobble%
8571 \let\ledsidenote\@gobble%
8572 }%
8573 \newcommand{\restore@sidenotes}{%
8574 \let\ledrightnote\@@ledrightnote%
8575 \let\ledleftnote\@@ledleftnote%
8576 \let\ledsidenote\@@ledsidenote%
8577 }%
8578 %

```

`\disable@endnotes` The same for endnotes.

```

\restore@endnotes
8579 \newcommand{\disable@endnotes}{%
8580   \unless\ifnoend@%
8581     \def\do##1{%
8582       \csletcs{##1@endnote}{##1endnote}%
8583       \expandafter\renewcommand \csname ##1endnote\endcsname[1]{}%
8584     }%
8585     \dolistloop{\@series}%
8586   \fi%
8587 }%
8588 \newcommand{\restore@endnotes}{%
8589   \unless\ifnofamiliar@%
8590     \def\do##1{%
8591       \csletcs{##1endnote}{##1@endnote}%
8592     }%
8593     \dolistloop{\@series}%
8594   \fi%
8595 }%
8596 %

```

`\disable@notes` Disable/restore side, familiar and end notes.

```

\restore@notes
8597 \newcommand{\disable@notes}{%
8598   \disable@sidenotes%
8599   \disable@familiarnotes%
8600   \disable@endnotes%
8601 }%
8602 \newcommand{\restore@notes}{%
8603   \restore@sidenotes%
8604   \restore@familiarnotes%
8605   \restore@endnotes%
8606 }%
8607 %

```

`\EDTEXT` We need to be able to modify the `\edtext` macros and also restore their original definitions.

```

\xedtext
8608 \let\EDTEXT=\edtext
8609 \newcommand{\xedtext}[2]{\EDTEXT{#1}{#2}}
8610 %

```

`\EDLABEL` We need to be able to modify and restore the `\edlabel` macro.

```

\xedlabel
8611 \let\EDLABEL=\edlabel
8612 \newcommand*\xedlabel[1]{\EDLABEL{#1}}
8613 %

```

`\xedindex` Macros supporting modification and restoration of `\edindex`.

`\nulledindex`

```

8614 \AtBeginDocument{\let\xedndex\edíndex}%
8615 \newcommand{\nulledíndex}[2][\jobname]{\@bsphack\@esphack}
8616
8617 %

```

**\@line@num** Macro supporting restoration of \linenum.

```

8618 \let\@line@num=\linenum
8619 %

```

**\l@dgobbleoptarg** \l@dgobbleoptarg[⟨arg⟩]{⟨arg⟩} replaces these two arguments (first is optional) by \relax.

```

8620 \newcommand*\l@dgobbleoptarg[2][ ]{\relax}%
8621
8622 %

```

**\l@secondmandarg** \l@secondoptarg[⟨arg⟩]{⟨arg⟩} gobble the first (optional) argument, and expand to the second (mandatory) argument.

```

8623 \NewExpandableDocumentCommand{\l@secondmandarg}{om}{#2}%
8624 %

```

**\Relax** \let\Relax=\relax

**\NEXT** \let\NEXT=\next

```

8627
8628 %

```

**\l@dmodforedtext** Modify and restore various macros for when \edtext is used.

**\l@drestoreforedtext**

```

8629 \newcommand{\l@dmodforedtext}{%
8630   \let\edtext\relax
8631   \def\do##1{\global\csletcs{##1footnote}{\l@dgobbleoptarg}}%
8632   \dolistloop{\@series}%
8633   \let\edíndex\nulledíndex
8634   \let\linenum@gobble}
8635 \newcommand{\l@drestoreforedtext}{%
8636   \def\do##1{\global\csletcs{##1footnote}{##1@footnote}}
8637   \dolistloop{\@series}%
8638   \let\edíndex\xedndex}
8639 %

```

**\l@dnullfills** Nullify and restore some column fillers, etc.

**\l@drestorefills**

```

8640 \newcommand{\l@dnullfills}{%
8641   \def\edlabel##1{%
8642     \def\edrowfill##1##2##3{%
8643     }
8644   \newcommand{\l@drestorefills}{%

```

```

8645 \def\edrowfill##1##2##3{\@EDROWFILL@{##1}{##2}{##3}}%
8646 }
8647
8648 %

```

**\letsforverteilen** Gathers some lets and other code that is common to the *\*verteilen\** macros.

```

8649 \newcommand{\letsforverteilen}{%
8650 \let\edtext\xedtext
8651 \let\edindex\xedindex
8652 \def\do##1{\global\csletcs{##1footnote}{##1@@footnote}}
8653 \dolistloop{\@series}%
8654 \let\linenum\@line@num
8655 \hilfsskip=\l@dcolwidth%
8656 \advance\hilfsskip by -\wd\hilfsbox
8657 \def\edlabel##1{\xedlabel{##1}}
8658
8659 %

```

**\disablel@dtabfeet** Declarations for using or using \edtext inside tabulars. The default at this point is for  
**\enablel@dtabfeet** \edtext.

```

8660 \newcommand\disablel@dtabfeet{\l@dmodforedtext}%
8661 \newcommand\enablel@dtabfeet{\l@drestoreforedtext}%
8662 %

```

### XXXII.2.2 Counters, boxes and lengths

**\l@dampcount** \l@dampcount is a counter for the & column dividers and \l@dcolcount is a counter  
**\l@dcolcount** for the columns.

```

8663 \newcount\l@dampcount
8664 \l@dampcount=1\relax
8665 \newcount\l@dcolcount
8666 \l@dcolcount=0\relax
8667
8668 %

```

**\hilfsbox** Some (temporary) helper items.

```

\hilfsskip
\Hilfsbox
\hilfscount
8669 \newbox\hilfsbox
8670 \newskip\hilfsskip
8671 \newbox\Hilfsbox
8672 \newcount\hilfscount
8673
8674 %

```

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).

```

8675 \newdimen\dcoli
8676 \newdimen\dcolii
8677 \newdimen\dcoliii
8678 \newdimen\dcoliv
8679 \newdimen\dcolv
8680 \newdimen\dcolvi
8681 \newdimen\dcolvii
8682 \newdimen\dcolviii
8683 \newdimen\dcolix
8684 \newdimen\dcolx
8685 \newdimen\dcolxi
8686 \newdimen\dcolxii
8687 \newdimen\dcolxiii
8688 \newdimen\dcolxiv
8689 \newdimen\dcolxv
8690 \newdimen\dcolxvi
8691 \newdimen\dcolxvii
8692 \newdimen\dcolxviii
8693 \newdimen\dcolxix
8694 \newdimen\dcolxx
8695 \newdimen\dcolxxi
8696 \newdimen\dcolxxii
8697 \newdimen\dcolxxiii
8698 \newdimen\dcolxxiv
8699 \newdimen\dcolxxv
8700 \newdimen\dcolxxvi
8701 \newdimen\dcolxxvii
8702 \newdimen\dcolxxviii
8703 \newdimen\dcolxxix
8704 \newdimen\dcolxxx
8705 \newdimen\dcolerr % added for error handling
8706
8707 %

```

`\l@dcolwidth` This is a cunning way of storing the columnwidths indexed by the column number `\l@dcolcount`, like an array. (was `\Dimenzuordnung`)

```

8708 \newcommand{\l@dcolwidth}{\ifcase \the\l@dcolcount \dcoli %???
8709 \or \dcoli \or \dcolii \or \dcoliii
8710 \or \dcoliv \or \dcolv \or \dcolvi
8711 \or \dcolvii \or \dcolviii \or \dcolix \or \dcolx
8712 \or \dcolxi \or \dcolxii \or \dcolxiii
8713 \or \dcolxiv \or \dcolxv \or \dcolxvi
8714 \or \dcolxvii \or \dcolxviii \or \dcolxix \or \dcolxx
8715 \or \dcolxxi \or \dcolxxii \or \dcolxxiii
8716 \or \dcolxxiv \or \dcolxxv \or \dcolxxvi
8717 \or \dcolxxvii \or \dcolxxviii \or \dcolxxix \or \dcolxxx
8718 \else \dcolerr \fi}
8719
8720 %

```

`\step1@dcolcount` This increments the column counter, and issues an error message if it is too large.

```

8721 \newcommand*{\step1@dcolcount}{\advance\l@dcolcount\@ne
8722 \ifnum\l@dcolcount>30\relax
8723 \led@err@TooManyColumns
8724 \fi}
8725
8726 %

```

`\l@dsetmaxcolwidth` Sets the column width to the maximum value seen so far.

```

8727 \newcommand{\l@dsetmaxcolwidth}{%
8728 \ifdim\l@dcolwidth < \wd\hilfsbox
8729 \l@dcolwidth = \wd\hilfsbox
8730 \else \relax \fi}
8731
8732 %

```

`\measuremcell` Measure (recursively) the width required for a math cell.

```

8733 \def\measuremcell #1{%
8734 \ifx #1\ \ifnum\l@dcolcount=0\let\NEXT\relax%
8735 \else\l@dcheckcols%
8736 \l@dcolcount=0%
8737 \let\NEXT\measuremcell%
8738 \fi%
8739 \else\setbox\hilfsbox=\hbox{$\displaystyle{#1}$}%
8740 \step1@dcolcount%
8741 \l@dsetmaxcolwidth%
8742 \let\NEXT\measuremcell%
8743 \fi\NEXT}
8744
8745 %

```

`\measuretcell` Measure (recursively) the width required for a text cell.

```

8746 \def\measuretcell #1{%
8747 \ifx #1\ \ifnum\l@dcolcount=0\let\NEXT\relax%
8748 \else\l@dcheckcols%
8749 \l@dcolcount=0%
8750 \let\NEXT\measuretcell%
8751 \fi%
8752 \else\setbox\hilfsbox=\hbox{#1}%
8753 \step1@dcolcount%
8754 \l@dsetmaxcolwidth%
8755 \let\NEXT\measuretcell%
8756 \fi\NEXT}
8757
8758 %

```

`\measuremrow` Measure (recursively) the width required for a math row.

```

8759 \def\measuremrow #1\{ {%
8760   \ifx #1&\let\NEXT\relax%
8761   \else\measuremcell #1&\&\&%
8762     \let\NEXT\measuremrow%
8763   \fi\NEXT}
8764 %

```

`\measuretrow` Measure (recursively) the width required for a text row.

```

8765 \def\measuretrow #1\{ {%
8766   \ifx #1&\let\NEXT\relax%
8767   \else\measuretcell #1&\&\&%
8768     \let\NEXT\measuretrow%
8769   \fi\NEXT}
8770 %
8771 %

```

`\edtabcolsep` The length `\edtabcolsep` controls the distance between columns.

```

8772 \newskip\edtabcolsep
8773 \global\edtabcolsep=10pt
8774 %
8775 %

```

`\variab`<sub>76</sub> `\newcommand{\variab}{\relax}`

```

8777 %
8778 %

```

`\l@dccheckcols` Check that the number of columns is consistent.

```

8779 \newcommand*{\l@dccheckcols}{%
8780   \ifnum\l@dcolcount=1\relax
8781   \else
8782     \ifnum\l@dampcount=1\relax
8783     \else
8784       \ifnum\l@dcolcount=\l@dampcount\relax
8785       \else
8786         \l@d@err@UnequalColumns
8787       \fi
8788     \fi
8789     \l@dampcount=\l@dcolcount
8790   \fi}
8791 %
8792 %

```

`\edfilldimen` A length.

```

8793 \newdimen\edfilldimen
8794 \edfilldimen=0pt
8795
8796 %

```

**\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).

**\theaddcolcount**

```

8797 \newcounter{addcolcount}
8798 \renewcommand{\theaddcolcount}{\romannumeral \c@addcolcount}
8799 %

```

### XXXII.2.3 Tabular typesetting

**\setmcellright** Typeset (recursively) cells of display math right justified.

```

8800 \def\setmcellright #1&{\def\edlabel##1{}%
8801   \let\edindex\nulledindex
8802   \ifx #1\\ \ifnum\l@dcolcount=0%\removelastskip
8803     \let\Next\relax%
8804     \else\l@dcolcount=0%
8805     \let\Next=\setmcellright%
8806     \fi%
8807   \else%
8808     \disablel@dtabfeet%
8809     \step\l@dcolcount%
8810     \disable@notes%
8811     \setbox\hilfsbox=\hbox{$\displaystyle{#1}$}%
8812     \restore@notes%
8813     \letsforverteilen%
8814     \hskip\hilfsskip$\displaystyle{#1}$%
8815     \hskip\edtabcolsep%
8816     \let\Next=\setmcellright%
8817   \fi\Next}
8818
8819 %

```

**\settcclright** Typeset (recursively) cells of text right justified.

```

8820 \def\settcclright #1&{\def\edlabel##1{}%
8821   \let\edindex\nulledindex
8822   \ifx #1\\ \ifnum\l@dcolcount=0%\removelastskip
8823     \let\Next\relax%
8824     \else\l@dcolcount=0%
8825     \let\Next=\settcclright%
8826     \fi%
8827   \else%
8828     \disablel@dtabfeet%

```



```

8829         \stepl@dc colcount%
8830         \disable@notes%
8831         \setbox\hilfsbox=\hbox{#1}%
8832         \restore@notes%
8833         \letsforverteilen%
8834         \hskip\hilfsskip#1%
8835         \hskip\edtabcolsep%
8836         \let\Next=\settcellright%
8837     \fi\Next}
8838 %

```

**\setmcellleft** Typeset (recursively) cells of display math left justified.

```

8839 \def\setmcellleft #1&{\def\edlabel##1{}}%
8840 \let\edindex\nulledindex
8841 \ifx #1\\ \ifnum\l@dc colcount=0 \let\Next\relax%
8842         \else\l@dc colcount=0%
8843         \let\Next=\setmcellleft%
8844         \fi%
8845 \else \disablel@dtabfeet%
8846         \stepl@dc colcount%
8847         \disable@notes%
8848         \setbox\hilfsbox=\hbox{$\displaystyle{#1}$}%
8849         \restore@notes%
8850         \letsforverteilen%
8851         $\displaystyle{#1}$\hskip\hilfsskip\hskip\edtabcolsep%
8852         \let\Next=\setmcellleft%
8853     \fi\Next}
8854 %
8855 %

```

**\settcellleft** Typeset (recursively) cells of text left justified.

```

8856 \def\settcellleft #1&{\def\edlabel##1{}}%
8857 \let\edindex\nulledindex
8858 \ifx #1\\ \ifnum\l@dc colcount=0 \let\Next\relax%
8859         \else\l@dc colcount=0%
8860         \let\Next=\settcellleft%
8861         \fi%
8862 \else \disablel@dtabfeet%
8863         \stepl@dc colcount%
8864         \disable@notes%
8865         \setbox\hilfsbox=\hbox{#1}%
8866         \restore@notes%
8867         \letsforverteilen%
8868         #1\hskip\hilfsskip\hskip\edtabcolsep%
8869         \let\Next=\settcellleft%
8870     \fi\Next}
8871 %

```

**\setmcellcenter** Typeset (recursively) cells of display math centered.

```

8872 \def\setmcellcenter #1&{\def\edlabel##1{}%
8873   \let\edindex\nulledindex
8874   \ifx #1\\ \ifnum\l@dc@lcount=0\let\Next\relax%
8875     \else\l@dc@lcount=0%
8876       \let\Next=\setmcellcenter%
8877     \fi%
8878   \else \disablel@dt@bfeet%
8879     \stepl@dc@lcount%
8880     \disable@notes%
8881     \setbox\hilfsbox=\hbox{$\displaystyle{#1}$}%
8882     \restore@notes%
8883     \letsforverteilen%
8884     \hskip 0.5\hilfsskip$\displaystyle{#1}$\hskip0.5\hilfsskip%
8885     \hskip\edtabcolsep%
8886     \let\Next=\setmcellcenter%
8887   \fi\Next}
8888
8889 %

```

**\settc@lcenter** Typeset (recursively) cells of text centered.

```

8890 \def\settc@lcenter #1&{\def\edlabel##1{}%
8891   \let\edindex\nulledindex
8892   \ifx #1\\ \ifnum\l@dc@lcount=0 \let\Next\relax%
8893     \else\l@dc@lcount=0%
8894       \let\Next=\settc@lcenter%
8895     \fi%
8896   \else \disablel@dt@bfeet%
8897     \stepl@dc@lcount%
8898     \disable@notes%
8899     \setbox\hilfsbox=\hbox{#1}%
8900     \restore@notes%
8901     \letsforverteilen%
8902     \hskip 0.5\hilfsskip #1\hskip 0.5\hilfsskip%
8903     \hskip\edtabcolsep%
8904     \let\Next=\settc@lcenter%
8905   \fi\Next}
8906
8907 %

```

**\NEXT** \let\NEXT=\relax

```

8909
8910 %

```

**\setmrowright** Typeset (recursively) rows of right justified math.

```

8911 \def\setmrowright #1\\{

```

```

8912 \ifx #1& \let\NEXT\relax
8913 \else \centerline{\setmcellright #1&\\&\\&}
8914 \let\NEXT=\setmrowright
8915 \fi\NEXT}
8916 %

```

**\settroright** Typeset (recursively) rows of right justified text.

```

8917 \def\settroright #1\\{%
8918 \ifx #1& \let\NEXT\relax
8919 \else \centerline{\settcclright #1&\\&\\&}
8920 \let\NEXT=\settroright
8921 \fi\NEXT}
8922 %
8923 %

```

**\setmrowleft** Typeset (recursively) rows of left justified math.

```

8924 \def\setmrowleft #1\\{%
8925 \ifx #1& \let\NEXT\relax
8926 \else \centerline{\setmcellleft #1&\\&\\&}
8927 \let\NEXT=\setmrowleft
8928 \fi\NEXT}
8929 %

```

**\settrorleft** Typeset (recursively) rows of left justified text.

```

8930 \def\settrorleft #1\\{%
8931 \ifx #1& \let\NEXT\relax
8932 \else \centerline{\settcclleft #1&\\&\\&}
8933 \let\NEXT=\settrorleft
8934 \fi\NEXT}
8935 %
8936 %

```

**\setmrowcenter** Typeset (recursively) rows of centered math.

```

8937 \def\setmrowcenter #1\\{%
8938 \ifx #1& \let\NEXT\relax%
8939 \else \centerline{\setmcellcenter #1&\\&\\&}
8940 \let\NEXT=\setmrowcenter
8941 \fi\NEXT}
8942 %

```

**\settrorcenter** Typeset (recursively) rows of centered text.

```

8943 \def\settrorcenter #1\\{%
8944 \ifx #1& \let\NEXT\relax
8945 \else \centerline{\settcclcenter #1&\\&\\&}
8946 \let\NEXT=\settrorcenter

```

```

8947 \fi\NEXT}
8948
8949 %

```

```

\newcommand{\nullsetzen}{%
8951 \step1@dcolcount%
8952 \l@dcolwidth=0pt%
8953 \ifnum\l@dcolcount=30\let\NEXT\relax%
8954 \l@dcolcount=0\relax
8955 \else\let\NEXT\nullsetzen%
8956 \fi\NEXT}
8957
8958 %

```

**\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.

```

8959 \newcommand{\edatleft}[3][\@empty]{%
8960 \ifx#1\@empty
8961 \vbox to 10pt{\vss\hbox{$\left#2\vrule width0pt height #3
8962 \depth 0pt \right. $\hss}\vfil}
8963 \else
8964 \vbox to 4pt{\vss\hbox{$#1\left#2\vrule width0pt height #3
8965 \depth 0pt \right. $\}\vfil}
8966 \fi}
8967 %

```

**\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.

```

8968 \newcommand{\edatright}[3][\@empty]{%
8969 \ifx#1\@empty
8970 \vbox to 10pt{\vss\hbox{$\left.\vrule width0pt height #3
8971 \depth 0pt \right#2 $\hss}\vfil}
8972 \else
8973 \vbox to 4pt{\vss\hbox{$\left.\vrule width0pt height #3
8974 \depth 0pt \right#2 #1 $\}\vfil}
8975 \fi}
8976
8977 %

```

**\edvertline** `\edvertline{ $\langle len \rangle$ }` vertical line  $\langle len \rangle$  high.

```

8978 \newcommand{\edvertline}[1]{\vbox to 8pt{\vss\hbox{\vrule height #1}\vfil}}
8979
8980 %

```

**\edvertdots** `\edvertdots{ $\langle len \rangle$ }` vertical dotted line  $\langle len \rangle$  high.

```

8981 \newcommand{\edvertdots}[1]{\vbox to 1pt{\vss\vbox to #1%
8982   {\cleaders\hbox{$\m@th\hbox{.}\vbox to 0.5em{ }$}\vfil}}}
8983
8984 %

```

**\l@dtabaddcols** `\l@dtabaddcols{<startcol>}{<endcol>}` adds the widths of the columns `<startcol>` through `<endcol>` to `\edfilldimen`. It is a  $\TeX$  style reimplementation of the original `\@add@`.

```

8985 \newcommand{\l@dtabaddcols}[2]{%
8986   \l@dcheckstartend{#1}{#2}%
8987   \ifl@dstartendok
8988     \setcounter{addcolcount}{#1}%
8989     \@whilenum \value{addcolcount}<#2\relax \do
8990     {\advance\edfilldimen by \the\csname dcol\theaddcolcount\endcsname
8991      \advance\edfilldimen by \edtabcolsep
8992      \stepcounter{addcolcount}}%
8993     \advance\edfilldimen by \the\csname dcol\theaddcolcount\endcsname
8994     \fi
8995   }
8996
8997 %

```

**\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.

```

8998 \newif\ifl@dstartendok
8999 \newcommand{\l@dcheckstartend}[2]{%
9000   \l@dstartendoktrue
9001   \ifnum #1<\@ne
9002     \l@dstartendokfalse
9003     \led@err@LowStartColumn
9004   \fi
9005   \ifnum #2>30\relax
9006     \l@dstartendokfalse
9007     \led@err@HighEndColumn
9008   \fi
9009   \ifnum #1>#2\relax
9010     \l@dstartendokfalse
9011     \led@err@ReverseColumns
9012   \fi
9013 }
9014
9015 %

```

**\edrowfill** `\edrowfill{<startcol>}{<endcol>}` fill fills columns `<startcol>` to `<endcol>` inclusive with `<fill>` (e.g. `\hrulefill`, `\upbracefill`). This is a  $\TeX$  style reimplementation and generalization of the original `\waklam`, `\Waklam`, `\waklamec`, `\wastricht` and `\wapunktetel` macros.

```

9016 \newcommand*\edrowfill}[3]{%
9017   \l@dtabaddcols{#1}{#2}%
9018   \hb@xt@ \the\l@dcollwidth{\hb@xt@ \the\edfilldimen{#3}\hss}}
9019 \let\@edrowfill=\edrowfill
9020 \def\@EDROWFILL@#1#2#3{\@edrowfill@{#1}{#2}{#3}}
9021
9022 %

```

**\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`.

```

9023 \newcommand{\leftltab}[1]{%
9024   \hb@xt@ \z@{\vbox{\edtabindent%
9025     \moveleft\Hilfsskip\hbox{\ #1}}\hss}}
9026
9027 %

```

**\leftrtab**      `\leftrtab{<text>}{<math>}` for `\edbeforetab` in `\rtab`.

```

9028 \newcommand{\leftrtab}[2]{%
9029   #2\hb@xt@ \z@{\vbox{\edtabindent%
9030     \advance\Hilfsskip by\dcoli%
9031     \moveleft\Hilfsskip\hbox{\ #1}}\hss}}
9032
9033 %

```

**\leftctab**      `\leftctab{<text>}{<math>}` for `\edbeforetab` in `\ctab`.

```

9034 \newcommand{\leftctab}[2]{%
9035   \hb@xt@ \z@{\vbox{\edtabindent\l@dcolcount=\l@dampcount%
9036     \advance\Hilfsskip by 0.5\dcoli%
9037     \setbox\hilfsbox=\hbox{\def\edlabel##1{}%
9038       \disablel@dtabfeet$\displaystyle{#2}$}%
9039     \advance\Hilfsskip by -0.5\wd\hilfsbox%
9040     \moveleft\Hilfsskip\hbox{\ #1}}\hss}%
9041   #2}
9042
9043 %

```

**\rightctab**      `\rightctab{<math>}{<text>}` for `\edaftertab` in `\ctab`.

```

9044 \newcommand{\rightctab}[2]{%
9045   \setbox\hilfsbox=\hbox{\def\edlabel##1{}%
9046     \disablel@dtabfeet#2}\l@dampcount=\l@dcolcount%
9047   #1\hb@xt@ \z@{\vbox{\edtabindent\l@dcolcount=\l@dampcount%

```

```

9048 \advance\Hilfsskip by 0.5\l@dcwidth%
9049 \advance\Hilfsskip by -\wd\hilfsbox%
9050 \setbox\hilfsbox=\hbox{\def\edlabel##1{}}%
9051 \disablel@dtabfeet$\displaystyle{#1}$}%
9052 \advance\Hilfsskip by -0.5\wd\hilfsbox%
9053 \advance\Hilfsskip by \edtabcolsep%
9054 \moveright\Hilfsskip\hbox{ #2}}\hss}%
9055 }
9056
9057 %

```

**\rightltab** `\rightltab{<math><math>}{<text>}` for `\edaftertab` in `\ltab`.

```

9058 \newcommand{\rightltab}[2]{%
9059 \setbox\hilfsbox=\hbox{\def\edlabel##1{}}%
9060 \disablel@dtabfeet#2}\l@dampcount=\l@dcwidthcount%
9061 #1\hb@xt@{z@{\vbox{\edtabindent\l@dcwidthcount=\l@dampcount%
9062 \advance\Hilfsskip by\l@dcwidth%
9063 \advance\Hilfsskip by-\wd\hilfsbox%
9064 \setbox\hilfsbox=\hbox{\def\edlabel##1{}}%
9065 \disablel@dtabfeet$\displaystyle{#1}$}%
9066 \advance\Hilfsskip by-\wd\hilfsbox%
9067 \advance\Hilfsskip by\edtabcolsep%
9068 \moveright\Hilfsskip\hbox{ #2}}\hss}%
9069 }
9070
9071 %

```

**\rightrtab** `\rightrtab{<math><math>}{<text>}` for `\edaftertab` in `\rtab`.

```

9072 \newcommand{\rightrtab}[2]{%
9073 \setbox\hilfsbox=\hbox{\def\edlabel##1{}}%
9074 \disablel@dtabfeet#2}%
9075 #1\hb@xt@{z@{\vbox{\edtabindent%
9076 \advance\Hilfsskip by-\wd\hilfsbox%
9077 \advance\Hilfsskip by\edtabcolsep%
9078 \moveright\Hilfsskip\hbox{ #2}}\hss}%
9079 }
9080
9081 %

```

**\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.

```

9082 \newcommand{\rtab}[1]{%
9083 \l@dnnullfills
9084 \def\edbeforetab##1##2{\lefttab{##1}{##2}}%
9085 \def\edaftertab##1##2{\righttab{##1}{##2}}%
9086 \measurembody{#1}%

```

```

9087 \l@drestorefills
9088 \variab
9089 \setmrowright #1\\&\\%
9090 \enablel@dtabfeet}
9091
9092 %

```

**\measurembody** `\measurembody{⟨body⟩}` measures the array `⟨body⟩`.

```

9093 \newcommand{\measurembody}[1]{%
9094 \disablel@dtabfeet%
9095 \l@dcolcount=0%
9096 \nullsetzen%
9097 \l@dcolcount=0
9098 \measuremrow #1\\&\\%
9099 \global\l@dampcount=1}
9100
9101 %

```

**\rtabtext** `\rtabtext{⟨body⟩}` typesets `⟨body⟩` as a tabular with the entries right justified.

```

9102 \newcommand{\rtabtext}[1]{%
9103 \l@dnullfills
9104 \measuretbody{#1}%
9105 \l@drestorefills
9106 \variab
9107 \settroright #1\\&\\%
9108 \enablel@dtabfeet}
9109
9110 %

```

**\measuretbody** `\measuretbody{⟨body⟩}` measures the tabular `⟨body⟩`.

```

9111 \newcommand{\measuretbody}[1]{%
9112 \disable@notes%
9113 \disablel@dtabfeet%
9114 \l@dcolcount=0%
9115 \nullsetzen%
9116 \l@dcolcount=0
9117 \measuretbody #1\\&\\%
9118 \restore@notes%
9119 \global\l@dampcount=1}
9120
9121 %

```

**\ltab** Array with entries left justified.

```

\edbeforetab
\edaftertab
9122 \newcommand{\ltab}[1]{%
9123 \l@dnullfills
9124 \def\edbeforetab##1##2{\leftltab{##1}{##2}}%

```



```

9125 \def\edaftertab##1##2{\rightltab{##1}{##2}}%
9126 \measuretbody{#1}%
9127 \l@drestorefills
9128 \variab
9129 \setmrowleft #1\\&\\%
9130 \enablel@dtabfeet}
9131
9132 %

```

**\ltabtext** Tabular with entries left justified.

```

9133 \newcommand{\ltabtext}[1]{%
9134 \l@dnnullfills
9135 \measuretbody{#1}%
9136 \l@drestorefills
9137 \variab
9138 \settrrowleft #1\\&\\%
9139 \enablel@dtabfeet}
9140
9141 %

```

**\ctab** Array with centered entries.

```

\edbeforetab
\edaftertab
9142 \newcommand{\ctab}[1]{%
9143 \l@dnnullfills
9144 \def\edbeforetab##1##2{\leftctab{##1}{##2}}%
9145 \def\edaftertab##1##2{\rightctab{##1}{##2}}%
9146 \measuretbody{#1}%
9147 \l@drestorefills
9148 \variab
9149 \setmrowcenter #1\\&\\%
9150 \enablel@dtabfeet}
9151
9152 %

```

**\ctabtext** Tabular with entries centered.

```

9153 \newcommand{\ctabtext}[1]{%
9154 \l@dnnullfills
9155 \measuretbody{#1}%
9156 \l@drestorefills
9157 \variab
9158 \settrrowcenter #1\\&\\%
9159 \enablel@dtabfeet}
9160
9161 %

```

```

\spreadtext 62 \newcommand{\spreadtext}[1]{%\l@dcolcount=\l@dampcount%
9163 \hb@xt@ \the\l@dcolwidth{\hbox{#1}\hss}}
9164 %

```

```

\spreadmath65 \newcommand{\spreadmath}[1]{%
9166   \hb@xt@ \the\l@dcolwidth{\hbox{$\displaystyle{#1}$}\hss}}
9167
9168 %

```

**\HILFSskip** More helpers.

```

\Hilfsskip9169 \newskip\HILFSskip
9170 \newskip\Hilfsskip
9171
9172 %

```

```

\EDTABINDENT73 \newcommand{\EDTABINDENT}{%
9174   \ifnum\l@dcolcount=30\let\NEXT\relax\l@dcolcount=0%
9175   \else\step1@dcolcount%
9176     \advance\Hilfsskip by\l@dcolwidth%
9177     \ifdim\l@dcolwidth=0pt\advance\hilfscout\@ne
9178     \else\advance\Hilfsskip by \the\hilfscout\edtabcolsep%
9179     \hilfscout=1\fi%
9180     \let\NEXT=\EDTABINDENT%
9181   \fi\NEXT}%
9182 %

```

**\edtabindent** (was \tabindent)

```

9183 \newcommand{\edtabindent}{%
9184   \l@dcolcount=0\relax
9185   \Hilfsskip=0pt%
9186   \hilfscout=1\relax
9187   \EDTABINDENT%
9188   \hilfsskip=\hsize%
9189   \advance\hilfsskip -\Hilfsskip%
9190   \Hilfsskip=0.5\hilfsskip%
9191   }%
9192
9193 %

```

**\EDTAB** (was \TAB)

```

9194 \def\EDTAB #1|#2|{%
9195   \setbox\tabhilfbox=\hbox{$\displaystyle{#1}$}%
9196   \setbox\tabHilfbox=\hbox{$\displaystyle{#2}$}%
9197   \advance\tabelskip -\wd\tabhilfbox%
9198   \advance\tabelskip -\wd\tabHilfbox%
9199   \unhbox\tabhilfbox\hskip\tabelskip%
9200   \unhbox\tabHilfbox}%
9201
9202 %

```

`\tabhilfbox` Further helpers.

## XXXII.2.4 Environments

edtabularl edtabularc edtabularr The ‘environment’ forms for `\ltabtext`, `\ctabtext` and `\rtabtext`.

### XXXIII Quotation's commands

`\quotation` `\initnumbering@quote` defines quotation environment.

```

\endquotation
\quote
\endquote
\newcommand{\initnumbering@quote}{
\ifnoquotation@else
\renewcommand{\quotation}{\par\leavevmode%
\parindent=1.5em%
\skipnumbering%

```

```

9230         \ifautopar%
9231             \vskip-\parskip%
9232         \else%
9233             \vskip\topsep%
9234         \fi%
9235         \global\leftskip=\leftmargin%
9236         \global\rightskip=\leftmargin%
9237     }
9238     \renewcommand{\endquotation}{\par%
9239         \global\leftskip=0pt%
9240         \global\rightskip=0pt%
9241         \leavevmode%
9242         \skipnumbering%
9243         \ifautopar%
9244             \vskip-\parskip%
9245         \else%
9246             \vskip\topsep%
9247         \fi%
9248     }
9249     \renewcommand{\quote}{\par\leavevmode%
9250         \parindent=0pt%
9251         \skipnumbering%
9252         \ifautopar%
9253             \vskip-\parskip%
9254         \else%
9255             \vskip\topsep%
9256         \fi%
9257         \global\leftskip=\leftmargin%
9258         \global\rightskip=\leftmargin%
9259     }
9260     \renewcommand{\endquote}{\par%
9261         \global\leftskip=0pt%
9262         \global\rightskip=0pt%
9263         \leavevmode%
9264         \skipnumbering%
9265         \ifautopar%
9266             \vskip-\parskip%
9267         \else%
9268             \vskip\topsep%
9269         \fi%
9270     }
9271     \fi
9272 }
9273 %

```

## XXXIV Section's title commands

### XXXIV.1 Commands to disable some feature

`\ledsectnotoc` The `\ledsectnotoc` only disables the `\addcontentsline` macro.

```
9274 \newcommand{\ledsectnotoc}{\let\addcontentsline@gobblethree}
9275 %
```

`\ledsectnomark` The `\ledsectnomark` only disables the `\chaptermark`, `\sectionmark` and `\subsectionmark` macros.

```
9276 \newcommand{\ledsectnomark}{%
9277   \let\chaptermark@gobble%
9278   \let\sectionmark@gobble%
9279   \let\subsectionmark@gobble%
9280 }
9281 %
```

### 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  $\LaTeX$  is run a other time, this file is read. That:
  - Adds the pstart number to a list of pstarts where a sectioning command is used.
  - Defines a command, the name of which contains the pstart number, and which calls the normal  $\LaTeX$  sectioning command.
4. This last command is called when the pstart is effectively printed.

### 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).

```
9282 \catcode`\#=12
9283 \notbool{@noeled@sec}{%
9284 %
```

`\beforeeledchapter` For technical reasons, not yet solved, page-breaking before chapters can't be made automatically by `eledmac`. Users have to use `\beforeeledchapter`.

```
9285 \ifl@dmemoir
9286 \newcommand\beforeeledchapter{%
9287 \clearforchapter%
9288 }
9289 \else
9290 \newcommand\beforeeledchapter{%
9291 \if@openright%
9292 \cleardoublepage%
9293 \else%
9294 \clearpage%
9295 \fi%
9296 }
9297 \fi
9298 %
```

### XXXIV.4 Auxiliary commands

`\print@leftmargin@eledsection` `\print@leftmargin@eledsection` and `\print@rightmargin@eledsection` are added by `reledmac` inside the code of sectioning command, in order to affix lines numbers. They include tests for RTL languages.

```
9299 \def\print@rightmargin@eledsection{%
9300 \if@eled@sectioning%
9301 \begingroup%
9302 \if@RTL%
9303 \let\llap\rlap%
9304 \let\leftlinenum\rightlinenum%
9305 \let\leftlinenumR\rightlinenumR%
9306 \let\l@drd@ta\l@dld@ta%
9307 \let\l@drsn@te\l@dlsn@te%
9308 \fi%
9309 \hfill\l@drd@ta \csuse{LR}{\l@drsn@te}%
9310 \endgroup%
9311 \fi%
```

```

9312 }%
9313
9314 \def\print@leftmargin@eledsection{%
9315   \if@eled@sectioning%
9316     \leavevmode%
9317     \begingroup%
9318     \if@RTL%
9319       \let\rlap\llap%
9320       \let\rightlinenum\leftlinenum%
9321       \let\rightlinenumR\leftlinenumR%
9322       \let\l@dld@ta\l@drd@ta%
9323       \let\l@dlsn@te\l@drsn@te%
9324     \fi%
9325     \l@dld@ta\csuse{LR}{\l@dlsn@te}%
9326     \endgroup%
9327   \fi%
9328 }%
9329
9330 %

```

### XXXIV.5 Patching standard commands

`\M@sect` We have to patch `TeX`, `book` and `memoir` sectioning commands in order to:

- `\@mem@old@ssect` • Disable `\edtext` inside.
- `\@makechapterhead` • Disable page breaking (for `\chapter`).
- `\@makechapterhead` • Add line numbers and sidenotes.
- `\@makeschapterhead`
- `\@sect`
- `\@ssect`

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.

```

9331 \AtBeginDocument{%
9332
9333
9334 \pretocmd{\M@sect}
9335   {\let\old@edtext=\edtext%
9336    \let\edtext=\dummy@edtext@showlemma%
9337   }
9338   {}
9339   {}
9340
9341 \apptocmd{\M@sect}
9342   {\let\edtext=\old@edtext}
9343   {}
9344   {}
9345
9346 \patchcmd{\M@sect}
9347   { #9}

```

```

9348 { #9%
9349 \print@rightmargin@eledsection%
9350 }
9351 {}
9352 {}
9353
9354 \patchcmd{\M@sect}
9355 {\hskip #3\relax}
9356 {\hskip #3\relax%
9357 \print@leftmargin@eledsection%
9358 }
9359 {}
9360 {}
9361
9362 \patchcmd{\@mem@old@ssect}
9363 {#5}
9364 {#5%
9365 \print@leftmargin@eledsection%
9366 }
9367 {}
9368 {}
9369
9370 \patchcmd{\@mem@old@ssect}
9371 {\hskip #1}
9372 {\hskip #1%
9373 \print@rightmargin@eledsection%
9374 }
9375 {}
9376 {}
9377
9378
9379
9380 \patchcmd{\scr@startchapter}{\if@openright\cleardoublepage\else\clearpage\fi}{%
9381 \if@eled@sectioning\else%
9382 \ifl@dprintingpages\else%
9383 \if@openright\cleardoublepage\else\clearpage\fi}%No clearpage inside a
\Pages: will keep critical notes from printing on the title page. Here for
scrbook.
9384 \fi%
9385 \fi%
9386 }
9387 {}
9388 {}
9389
9390 \patchcmd{\@makechapterhead}
9391 {#1}
9392 {\print@leftmargin@eledsection%
9393 #1%
9394 \print@rightmargin@eledsection%

```



```

9395 }
9396 {}
9397 {}
9398
9399 \patchcmd{\@makechapterhead}% For BIDI
9400 {\if@RTL\raggedleft\else\raggedright\fi}%
9401 {\if@eled@sectioning\else%
9402   \if@RTL\raggedleft\else\raggedright\fi%
9403   \fi%
9404 }%
9405 {}%
9406 {}%
9407
9408 \patchcmd{\@makeschapterhead}
9409   {\#1}
9410   {\print@leftmargin@eledsection%
9411     \#1%
9412     \print@rightmargin@eledsection%
9413   }
9414   {}
9415   {}
9416
9417 \pretocmd{\@sect}
9418   {\let\old@edtext=\edtext
9419     \let\edtext=\dummy@edtext@showlemma%
9420   }
9421   {}
9422   {}
9423
9424 \apptocmd{\@sect}
9425   {\let\edtext=\old@edtext}
9426   {}
9427   {}
9428
9429 \pretocmd{\@ssect}
9430   {\let\old@edtext=\edtext%
9431     \let\edtext=\dummy@edtext@showlemma%
9432   }
9433   {}
9434   {}
9435
9436 \apptocmd{\@ssect}
9437   {\let\edtext=\old@edtext}
9438   {}
9439   {}
9440
9441 %

```

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.

```

9442 \@ifpackageloaded{nameref}{
9443
9444   \patchcmd{\NR@sect}
9445     {\#8}
9446     {\#8%
9447       \print@rightmargin@eledsection%
9448     }
9449     {}
9450     {}
9451
9452   \patchcmd{\NR@sect}
9453     {\hskip #3\relax}
9454     {\hskip #3\relax%
9455       \print@leftmargin@eledsection%
9456     }
9457     {}
9458     {}
9459
9460   \patchcmd{\NR@ssect}
9461     {\#5}
9462     {\#5%
9463       \print@rightmargin@eledsection%
9464     }
9465     {}
9466     {}
9467
9468   \patchcmd{\NR@ssect}
9469     {\hskip #1}
9470     {\hskip #1%
9471       \print@leftmargin@eledsection%
9472     }
9473     {}
9474     {}
9475   }%
9476   {
9477     \patchcmd{\@sect}
9478       {\#8}
9479       {\#8%
9480         \print@rightmargin@eledsection%
9481       }
9482       {}
9483       {}
9484
9485     \patchcmd{\@sect}
9486       {\hskip #3\relax}
9487       {\hskip #3\relax%
9488         \print@leftmargin@eledsection%
9489       }
9490       {}
9491       {}

```

```

9492 \patchcmd{\@ssect}
9493   {#5}
9494   {#5%
9495   \print@rightmargin@eledsection%
9496   }
9497   {}
9498   {}
9499   {}
9500
9501 \patchcmd{\@ssect}
9502   {\hskip #1}
9503   {\hskip #1%
9504   \print@leftmargin@eledsection%
9505   }
9506   {}
9507   {}
9508   }%
9509 }%
9510 %

```

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 `#`.

```

9511 {}}%
9512 \protect\catcode`\#=6 %Space NEEDS by \catcode
9513 %

```

**\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.

```

9514 \AtBeginDocument{%
9515 \patchcmd{\chapter}{\clearforchapter}{%
9516   \if@eled@sectioning\else%
9517   \ifl@dprintingpages\else%
9518   \clearforchapter%
9519   \fi%
9520 \fi%
9521 }%
9522 {}%
9523 {}%
9524
9525 \patchcmd{\chapter}{\if@openright\cleardoublepage\else\clearpage\fi}{%
9526   \if@eled@sectioning\else%
9527   \ifl@dprintingpages%
9528   \endgraf%
9529   \else%
9530   \if@openright\cleardoublepage\else\clearpage\fi%No clearpage inside a
\Pages: will keep critical notes from printing on the title page. Here for
classical classes
9531   \fi%

```

```

9532 \fi%
9533 }%
9534 {}%
9535 {}%
9536 }%
9537 %

```

**\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.

```

9538 \newif\if@eled@sectioning%
9539 %

```

We reopen a new `\notbool{@noeled@sec}` statement, as we will define the `\elesection` commands.

```

9540 \notbool{@noeled@sec}{%
9541 %

```

## 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.

```

9542 \newwrite\eled@sectioning@out
9543 %

```

**\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*
9544 \newcommand{\eledchapter}[2] [] {%
9545   \disable@familiarnotes%
9546   #2%
9547   \restore@familiarnotes%
9548   \ifledRcol%
9549     \immediate\write\eled@sectioningR@out{%
9550       \string\eled@chapter{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR}{R}
9551     }%
9552   \else%
9553     \immediate\write\eled@sectioning@out{%
9554       \string\eled@chapter{#1}{\unexpanded{#2}}{\the\pstarts@read@L}{L}%
9555     }%
9556   \fi%
9557 }
9558
9559 \newcommand{\eledsection}[2] [] {%
9560   \disable@familiarnotes%
9561   #2%

```

```

9562 \restore@familiarnotes%
9563 \ifledRcol%
9564 \immediate\write\eled@sectioningR@out{%
9565 \string\eled@section{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR}{R}
9566 }%
9567 \else%
9568 \immediate\write\eled@sectioning@out{%
9569 \string\eled@section{#1}{\unexpanded{#2}}{\the\pstarts@read@L}{R}%
9570 }%
9571 \fi%
9572 }
9573
9574 \newcommand{\eledsubsection}[2][{}]{%
9575 \disable@familiarnotes%
9576 #2%
9577 \restore@familiarnotes%
9578 \ifledRcol%
9579 \immediate\write\eled@sectioningR@out{%
9580 \string\eled@subsection{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR}{R}
9581 }%
9582 \else%
9583 \immediate\write\eled@sectioning@out{%
9584 \string\eled@subsection{#1}{\unexpanded{#2}}{\the\pstarts@read@L}{R}%
9585 }%
9586 \fi%
9587 }
9588 \newcommand{\eledsubsubsection}[2][{}]{%
9589 \disable@familiarnotes%
9590 #2%
9591 \restore@familiarnotes%
9592 \ifledRcol%
9593 \immediate\write\eled@sectioningR@out{%
9594 \string\eled@subsubsection{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR}
9595 }{R}%
9596 \else%
9597 \immediate\write\eled@sectioning@out{%
9598 \string\eled@subsubsection{#1}{\unexpanded{#2}}{\the\pstarts@read@L}
9599 }{R}%
9600 \fi%
9601 }
9602
9603
9604 \WithSuffix\newcommand\eledchapter*[2][{}]{%
9605 \disable@familiarnotes%
9606 #2%
9607 \restore@familiarnotes%

```

```

9608 \ifledRcol%
9609 \immediate\write\eled@sectioningR@out{%
9610 \string\eled@chapter{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR}{*}{R}
9611 }%
9612 \else%
9613 \immediate\write\eled@sectioning@out{%
9614 \string\eled@chapter{#1}{\unexpanded{#2}}{\the\pstarts@read@L}{*}{}%
9615 }%
9616 \fi%
9617 }
9618
9619 \WithSuffix\newcommand\eledsection*[2][]{%
9620 \disable@familiarnotes%
9621 #2%
9622 \restore@familiarnotes%
9623 \ifledRcol%
9624 \immediate\write\eled@sectioningR@out{%
9625 \string\eled@section{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR}{*}{R}
9626 }%
9627 \else%
9628 \immediate\write\eled@sectioning@out{%
9629 \string\eled@section{#1}{\unexpanded{#2}}{\the\pstarts@read@L}{*}{}%
9630 }%
9631 \fi%
9632 }
9633
9634 \WithSuffix\newcommand\eledsubsection*[2][]{%
9635 \disable@familiarnotes%
9636 #2%
9637 \restore@familiarnotes%
9638 \ifledRcol%
9639 \immediate\write\eled@sectioningR@out{%
9640 \string\eled@subsection{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR}{*}{R}
9641 }%
9642 \else%
9643 \immediate\write\eled@sectioning@out{%
9644 \string\eled@subsection{#1}{\unexpanded{#2}}{\the\pstarts@read@L}
9645 }{*}{}%
9646 }%
9647 \fi%
9648 }
9649
9649 \WithSuffix\newcommand\eledsubsubsection*[2][]{%
9650 \disable@familiarnotes%
9651 #2%
9652 \restore@familiarnotes%
9653 \ifledRcol%
9654 \immediate\write\eled@sectioningR@out{%
9655 \string\eled@subsubsection{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR}

```

```

}{{*}}{R}
    }%
9656 \else%
9657 \immediate\write\eled@sectioning@out{%
9658 \string\eled@subsubsection{#1}{\unexpanded{#2}}{\the\pstarts@read@L
9659 }{{*}}{}%
9660 }%
9661 \fi%
9662 }
9663 %

```

### XXXIV.7 Macros written in the auxiliary file

`\eled@chapter`  
`\eled@section`  
`\eled@subsection`  
`\eled@subsubsection`

The sectioning macros, called in the auxiliary file. They have five arguments:

1. Optional arguments of  $\LaTeX$  sectioning command.
2. Mandatory arguments of  $\LaTeX$  sectioning command.
3. Pstart number.
4. Side: R if right, nothing if left.
5. Starred or not.

```

9664 \def\eled@chapter#1#2#3#4#5{%
9665 \ifstrempy{#4}%
9666 {%
9667 \ifstrempy{#1}%
9668 {%
9669 \csgdef{eled@sectioning@#3#5}{\let\edtext=\dummy@edtext@showlemma\
chapter{#2}}%
9670 \csgdef{eled@sectmark@#3#5}{\let\edtext=\dummy@edtext{}\chaptermark
{#2}}%
9671 }%Need for \pairs, because of using parbox.
9672 {%
9673 \csgdef{eled@sectioning@#3#5}{\let\edtext=\dummy@edtext@showlemma\
chapter[#1]{#2}}%
9674 \csgdef{eled@sectmark@#3#5}{\let\edtext=\dummy@edtext{}\chaptermark
{#2}}%Need for \pairs, because of using parbox.
9675 }%
9676 }%
9677 {%
9678 \ifstrempy{#1}%
9679 {\csgdef{eled@sectioning@#3#5}{\let\edtext=\dummy@edtext@showlemma\
chapter*{#2}}}%
9680 {\csgdef{eled@sectioning@#3#5}{\let\edtext=\dummy@edtext@showlemma\
chapter*[#1]{#2}}}%Bug in LaTeX!
9681 }%
9682 \listcsgadd{eled@sections#5@@}{#3}%

```

```

9683 }
9684 \def\eled@section#1#2#3#4#5{%
9685   \ifstrempy{#4}%
9686     {\ifstrempy{#1}%
9687       {%
9688         \csgdef{eled@sectioning@#3#5}{\section{#2}}%
9689         \csgdef{eled@sectmark@#3#5}{\let\edtext=dummy@edtext{}\sectionmark
9690 {#2}}}%Need for \pairs, because of using parbox.
9691       }%
9692       {\csgdef{eled@sectioning@#3#5}{\section[#1]{#2}}%
9693       \csgdef{eled@sectmark@#3#5}{\let\edtext=dummy@edtext{}\sectionmark
9694 {#1}}}%Need for \pairs, because of using parbox.
9695     }%
9696     {\ifstrempy{#1}%
9697       {\csgdef{eled@sectioning@#3#5}{\section*{#2}}}%
9698       {\csgdef{eled@sectioning@#3#5}{\section*{#1}{#2}}}%Bug in LaTeX!
9699     }
9700   \listcsgadd{eled@sections#5@@}{#3}%
9701 }
9702 \def\eled@subsection#1#2#3#4#5{%
9703   \ifstrempy{#4}%
9704     {\ifstrempy{#1}%
9705       {%
9706         \csgdef{eled@sectioning@#3#5}{\subsection{#2}}%
9707         \csgdef{eled@sectmark@#3#5}{\let\edtext=dummy@edtext{}\csuse{
9708 subsectionmark}{#2}}}%Need for \pairs, because of using parbox. \csuse in
9709 case of \subsectionmark is not defined (book)
9710       }%
9711       {\csgdef{eled@sectioning@#3#5}{\subsection[#1]{#2}}%
9712       \csgdef{eled@sectmark@#3#5}{\let\edtext=dummy@edtext{}\csuse{
9713 subsectionmark}{#1}}}%Need for \pairs, because of using parbox. \csuse in
9714 case of \subsectionmark is not defined (book)
9715     }%
9716     {\ifstrempy{#1}%
9717       {\csgdef{eled@sectioning@#3#5}{\subsection*{#2}}}%
9718       {\csgdef{eled@sectioning@#3#5}{\subsection*{#1}{#2}}}%Bug in LaTeX!
9719     }
9720   \listcsgadd{eled@sections#5@@}{#3}%
9721 }
9722 \def\eled@subsubsection#1#2#3#4#5{%
9723   \ifstrempy{#4}%
9724     {\ifstrempy{#1}%
9725       {\csgdef{eled@sectioning@#3#5}{\subsubsection{#2}}}%
9726       {\csgdef{eled@sectioning@#3#5}{\subsubsection[#1]{#2}}}%
9727     }%
9728     {\ifstrempy{#1}%

```



```

9727     {\csgdef{eled@sectioning@#3#5}{\subsubsection*{#2}}}%
9728     {\csgdef{eled@sectioning@#3#5}{\subsubsection*{#1}{#2}}}%Bug in
LaTeX!
9729     }
9730     \listcsgadd{eled@sections#5@@}{#3}%
9731     }
9732
9733 %

```

End of the conditional test about noeledsec option.

```

9734 }{}
9735 %

```

## 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.

```

9736 \def\normal@page@break{}
9737 %

```

**\l@prev@pb** The \l@prev@pb macro is a etoolbox list, which contains the lines in which page breaks occur (before or after). The \l@prev@nopb macro is a etoolbox list, which contains the lines with NO page break before or after.

```

9738 \def\l@prev@pb{}
9739 \def\l@prev@nopb{}
9740 %

```

**\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.

```

9741 \newcommand{\ledpb}{\write\linenum@out{\string\led@pb}}
9742 \newcommand{\ledpbnum}[1]{\write\linenum@out{\string\led@pbnum{#1}}}
9743 \newcommand{\lednopb}{\write\linenum@out{\string\led@nopb}}
9744 \newcommand{\lednopbnum}[1]{\write\linenum@out{\string\led@nopbnum{#1}}}
9745 %

```

`\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.

```

9746 \newcommand{\led@pb}{\listxadd{\l@prev@pb}{\the\absline@num}}
9747 \newcommand{\led@pbnum}[1]{\listxadd{\l@prev@pb}{#1}}
9748 \newcommand{\led@nopb}{\listxadd{\l@prev@nopb}{\the\absline@num}}
9749 \newcommand{\led@nopbnum}[1]{\listxadd{\l@prev@nopb}{#1}}
9750 %

```

`\ledpbsetting` The `\ledpbsetting` macro only changes the value of `\led@pb@macro`, for which the default value is before.

```

9751 \def\led@pb@setting{before}
9752 \newcommand{\ledpbsetting}[1]{\gdef\led@pb@setting{#1}}
9753 %

```

`\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`.

```

9754 \newcommand{\led@check@pb}{\xifinlist{\the\absline@num}{\l@prev@pb}{\pagebreak[4]}}
9755 \newcommand{\led@check@nopb}{%
9756   \IfStrEq{\led@pb@setting}{before}{%
9757     \xifinlist{\the\absline@num}{\l@prev@nopb}{%
9758       {\numdef{\abs@prevline}{\the\absline@num-1}%
9759       \xifinlist{\abs@prevline}{\normal@page@break}%
9760       {\nopagebreak[4]\enlargethispage{\baselineskip}}%
9761       {}}%
9762     {}}%
9763   }%
9764 }%
9765 \IfStrEq{\led@pb@setting}{after}{%
9766   \xifinlist{\the\absline@num}{\l@prev@nopb}{%
9767     \xifinlist{\the\absline@num}{\normal@page@break}%
9768     {\nopagebreak[4]\enlargethispage{\baselineskip}}%
9769     {}}%
9770 }%
9771   {}}%
9772   {}%
9773 }%
9774 }
9775 %

```

## 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.

```

9776 \newcommand{\check@pb@in@verse}{%
9777   \ifinstanza\iflednopbinverse\ifinserthangingsymbol% Using stanzas and
enabling page breaks in verse control, while on a hanging verse.
9778   \ifnum\page@num=\last@page@num\else%If we have change page
9779   \IfStrEq{\led@pb@setting}{before}{%
9780     \numgdef{\abs@line@verse}{\the\absline@num-1}%
9781     \ledpbnum{\abs@line@verse}%
9782   }{}%
9783   \IfStrEq{\led@pb@setting}{after}{%
9784     \numgdef{\abs@line@verse}{\the\absline@num-1}%
9785     \lednopbnum{\abs@line@verse}%
9786   }{}%
9787   \fi%
9788 \fi\fi\fi%
9789 }
9790 %

```

## 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>

```

9791 \def\Hy@raisedlink@left#1{%
9792   \ifvmode
9793     #1%
9794   \else

```

<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>.

```

9795 \Hy@SaveSpaceFactor
9796 \llap{\smash{%
9797 \begingroup
9798 \let\HyperRaiseLinkLength\@tempdima
9799 \setlength\HyperRaiseLinkLength\HyperRaiseLinkDefault
9800 \HyperRaiseLinkHook
9801 \expandafter\endgroup
9802 \expandafter\raise\the\HyperRaiseLinkLength\hbox{%
9803 \Hy@RestoreSpaceFactor
9804 #1%
9805 \Hy@SaveSpaceFactor
9806 }%
9807 }}%
9808 \Hy@RestoreSpaceFactor
9809 \penalty\@M\hskip\z@\relax
9810 \fi
9811 }
9812 %

```

## XXXVIII Compatibility with eledmac

Here, we define some commands for the eledmac-compat option.

```

9813 \ifeledmaccompat@%
9814
9815 \newcommand{\footnormalX}[1]{\arrangementX[#1]{normal}}%
9816 \newcommand{\footparagraphX}[1]{\arrangementX[#1]{paragraph}}%
9817 \newcommand{\foottwocolX}[1]{\arrangementX[#1]{twocol}}%
9818 \newcommand{\footthreecolX}[1]{\XarrangementX[#1]{threecol}}%
9819
9820 \unless\ifnocritical@
9821 \newcommand{\footnormal}[1]{\Xarrangement[#1]{normal}}%
9822 \newcommand{\footparagraph}[1]{\Xarrangement[#1]{paragraph}}%
9823 \newcommand{\foottwocol}[1]{\Xarrangement[#1]{twocol}}%
9824 \newcommand{\footthreecol}[1]{\Xarrangement[#1]{threecol}}%
9825 \let\hsizetwocol\Xhsizetwocol
9826 \let\hsizethreecol\Xsizethreecol
9827 \let\bhookXnote\Xbhooknote
9828 \let\boxsymlinenum\Xboxsymlinenum
9829 \let\symlinenum\Xsymlinenum
9830 \let\beforenumberinfootnote\Xbeforenumber
9831 \let\afternumberinfootnote\Xafternumber
9832 \let\beforeXsymlinenum\Xbeforesymlinenum
9833 \let\afterXsymlinenum\Xaftersymlinenum
9834 \let\inplaceofnumber\Xinplaceofnumber
9835 \let\Xlemmaseparator\lemmaseparator
9836 \let\afterlemmaseparator\Xafterlemmaseparator
9837 \let\beforelemmaseparator\Xbeforelemmaseparator
9838 \let\inplaceoflemmaseparator\Xinplaceoflemmaseparator

```

```

9839 \let\txbeforeXnotes\Xtxbeforenotes
9840 \let\afterXrule\Xafterrule
9841 \let\numberonlyfirstinline\Xnumberonlyfirstinline
9842 \let\numberonlyfirstintwoline\Xnumberonlyfirstintwoline
9843 \let\nonumberinfootnote\Xnonumberinfootnote
9844 \let\pstartinfootnote\Xpstart
9845 \let\pstartinfootnoteeverytime\Xpstarteverytime
9846 \let\onlyXpstart\Xonlypstart
9847 \let\Xnonumberinfootnote\Xnonumber
9848 \let\Xnonbreakableafternumber\Xnonbreakableafternumber
9849 \let\maxhXnotes\Xmaxhnotes
9850 \let\beforeXnotes\Xbeforenotes
9851 \let\boxlinenum\Xboxlinenum
9852 \let\boxlinenumalign\Xboxlinenumalign
9853 \let\boxstartlinenum\Xboxstartlinenum
9854 \let\boxendlinenum\Xboxendlinenum
9855 \let\twoline\Xtwoline
9856 \let\morethantwoline\Xmorethantwoline
9857 \let\twolinebutnotmore\Xtwolinebutnotmore
9858 \let\twolineonlyinsamepage\Xtwolineonlyinsamepage
9859 \fi
9860
9861 \unless\ifnofamiliar@
9862 \let\notesXwidthliketwocolumns\noteswidthliketwocolumnsX
9863 \fi
9864 \newcommandx{\parafootsep}[2][1,usedefault]{%
9865 \Xparafootsep[#1]{#2}%
9866 \parafootsepX[#1]{#2}
9867 }%
9868
9869 \newcommandx{\afternote}[2][1,usedefault]{%
9870 \Xafternote[#1]{#2}%
9871 \afternoteX[#1]{#2}%
9872 }%
9873
9874 \unless\ifnoend@
9875 \let\XendXtwoline\Xendtwoline
9876 \let\XendXmorethantwoline\Xendmorethantwoline
9877 \let\XhookXendnote\Xendhooknote
9878 \let\XboxXendlinenum\Xendboxlinenum%
9879 \let\XboxXendlinenumalign\Xendboxlinenumalign%
9880 \let\XboxXendstartlinenum\Xendboxstartlinenum%
9881 \let\XboxXendendlinenum\Xendboxendlinenum%
9882 \let\XendXlemmaseparator\Xendlemmaseparator
9883 \let\XendXbeforelemmaseparator\Xendbeforelemmaseparator
9884 \let\XendXafterlemmaseparator\Xendafterlemmaseparator
9885 \let\XendXinplaceoflemmaseparator\Xendinplaceoflemmaseparator
9886 \fi
9887
9888 \AtBeginDocument{%

```

```
9889 \ifdef\lineref{}\let\lineref\edlineref}%  
9890 }%  
9891  
9892  
9893 \fi%  
9894 %
```

</code>

## Appendix A Things to do when changing versions

### A.1 Migrating from edmac to ledmac

If you have never used edmac, ignore this section. If you have used edmac and are starting on a completely new document, ignore this section. Only read this section if you are converting an original edmac document to use ledmac.

The package still provides the original `\text` command, but it is (a) deprecated, and (b) its name has been changed<sup>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. 133 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. 37), 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. 38) or `\Xfootnote` options (6.2.2 p. 25). Otherwise please add a new ticket on Github to request a new function for doing this.<sup>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. 53). 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. 18). If so, add a `\relax` between `\pstart`/`\pend` and the first bracket.

The version 1.12.0 also introduce a better way to handle sectional divisions inside numbered text. Please read 16.2 p. 72.

## A.5 Migration to eledmac 17.1

This version changes the default setting of `\Xpstart`. Henceforth, `pstart` numbers will be printed in footnotes within the section of text where you have called `\numberpstarttrue`.

We do not see any reason to print them in the other sections. However, if you want to print the `pstart` numbers in all of the footnotes, whatever the section, without having to use `\numberpstarttrue`, you can use `\Xpstarteverytime`.

## 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. 75 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 memoirare 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 every endnotes set (`.Aend`, `.Bend`, `.Cend` etc.). If you have overridden `\doendnotes` (which you would 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` in 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 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` (7.2.4 p. 41).

### A.14 Migration to reledmac 2.5.0

It is strongly recommended to stop redefining `\printnpnum` and to use the hooks documented in 7.2.3 p. 41.

`\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 fix 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`.



## References

- [Bre96] Herbert Breger. `tabmac`. October 1996. (Available from CTAN in `macros/plain/contrib/tabmac`)
- [Bur01] John Burt. ‘Typesetting critical editions of poetry’. *TUGboat*, **22**, 4, pp 353–361, December 2001. (Code available from CTAN in `macros/latex/contrib/poemscol`)
- [Eck03] Matthias Eckermann. *The Parallel-Package*. April 2003. (Available from CTAN in `macros/latex/contrib/parallel`)
- [Fai03] Robin Fairbairns. *footmisc—a portmanteau package for customising footnotes in L<sup>A</sup>T<sub>E</sub>X*. February 2003. (Available from CTAN in `macros/latex/contrib/footmisc`)
- [LW90] John Lavagnino and Dominik Wujastyk. ‘An overview of `edmac`: a PLAIN TeX format for critical editions’. *TUGboat*, **11**, 4, pp. 623–643, November 1990. (Code available from CTAN in `macros/plain/contrib/edmac`)
- [Lüc03] Uwe Lück. ‘`ednotes`—critical edition typesetting with LaTeX’. *TUGboat*, **24**, 2, pp. 224–236, 2003. (Code available from CTAN in `macros/latex/contrib/ednotes`)
- [Sul92] Wayne G. Sullivan. *The file `edstanzan.doc`*. June 1992. (Available from CTAN in `macros/plain/contrib/edmac`)
- [Wil02] Peter Wilson. *The memoir class for configurable typesetting*. November 2002. (Available from CTAN in `macros/latex/contrib/memoir`)
- [Wil04] Peter Wilson and Maïeul Rouquette. *Parallel typesetting for critical editions: the `eledpar` package*. December 2004. (Available from CTAN in `macros/latex/contrib/ledmmac`)

## Index

### Symbols

<code>\&amp;</code>	52
<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>\@doclearpage</code>	1
<code>\@doreinfeetX</code>	1

<code>\@edindex@fornote@</code>	1
<code>\@edindex@hyperref</code>	1
<code>\@edrowfill@</code>	1
<code>\@edtext@level</code>	1
<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>\@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>\@n1</code>	1
<code>\@n1@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
<code>\@sidenotesep</code>	1

\@ssect	1
\@startstanza	1
\@stopmsd	1
\@stopstanza	1
\@sw	1
\@tag	1
\@wredindex	1
\@xloop	1
\@xympar	1
CLASSarticle	75
CLASSbook	75, 367
CLASSmemoir	216, 217, 279–282, 317, 367, 387, 448, 452
CLASSscrbook	452
COMMAND\*footnote	76
COMMAND\...@footnotemark...	220
COMMAND\...d@ta	162
COMMAND\<hook	
@<series	267
COMMAND\<hookname	
<pseudoserries	269, 270
COMMAND\<type	
footfmt	205
COMMAND\@@line	196
COMMAND\@MM	183, 449
COMMAND\@Rlineflag	319, 449
COMMAND\@Srefprefix	296
COMMAND\@Srefprefixmore	296
COMMAND\@add@	357
COMMAND\@adv	116
COMMAND\@annot	168
COMMAND\@annot@end@print	208
COMMAND\@annot@start@print	208
COMMAND\@apprefprefixmore	296
COMMAND\@apprefprefixsingle	296
COMMAND\@beforeinsertofthisedtext	139, 140
COMMAND\@bsphack	284
COMMAND\@doclearpage	280, 281, 442, 452
COMMAND\@doreinfeetX	452
COMMAND\@dprintingcolumns	449
COMMAND\@edindex@hyperref	319, 320
COMMAND\@edtext@level	136
COMMAND\@esphack	284
COMMAND\@firstofone	33
COMMAND\@fnpos	238, 276
COMMAND\@footnotemark	217, 442, 452
COMMAND\@footnotetext	217, 218, 442
COMMAND\@gobble	33, 58, 134, 135, 168, 262
COMMAND\@gobblefive	451
COMMAND\@gobblefour	448
COMMAND\@gobbleseven	264

COMMAND\@gobblethree	441
COMMAND\@h	200
COMMAND\@hangingsymbol	324
COMMAND\@iiiminipage	308, 310, 441, 452
COMMAND\@iiiminpage	308
COMMAND\@l	446
COMMAND\@l@tempcnta	165, 169, 177
COMMAND\@l@tempcntb	169
COMMAND\@l@reg	446
COMMAND\@lab	112, 284, 287, 292, 441
COMMAND\@ldunboxmpfoot	311
COMMAND\@led@extranofeet	388
COMMAND\@ledinnote@command	315, 316
COMMAND\@lemma	139, 142
COMMAND\@lock	106, 324
COMMAND\@lopL	442
COMMAND\@lopR	442
COMMAND\@makecol	276, 277, 279, 452
COMMAND\@mempnum	282
COMMAND\@mpfnpos	238
COMMAND\@msd	334
COMMAND\@msd@c	334
COMMAND\@msd@options@iffullpage	340
COMMAND\@msdata@list	334, 335
COMMAND\@nl	112–117, 127, 168, 287, 441, 442
COMMAND\@nl@reg	113, 385, 442, 446
COMMAND\@opXfeet	442
COMMAND\@opfeetX	452
COMMAND\@opextrafeeti	452
COMMAND\@outputpage	282, 283
COMMAND\@page	112, 114, 287
COMMAND\@pend	442
COMMAND\@pendR	442
COMMAND\@ref	112, 122–124, 128, 129, 134
COMMAND\@ref@later	124, 130
COMMAND\@ref@reg	123, 168, 442
COMMAND\@ref@reg@parsearg	124
COMMAND\@reinserts	276–279, 452
COMMAND\@secondoftwo	77
COMMAND\@sect	369
COMMAND\@series	266
COMMAND\@set	117
COMMAND\@sidenotesep	307
COMMAND\@stopmsd	334
COMMAND\@sw	124, 144, 147, 148
COMMAND\@tag	136, 137, 141
COMMAND\@tempcnta	90
COMMAND\@tempcntb	90
COMMAND\@toksa	97
COMMAND\@toksb	97

COMMAND\@xloop	178
COMMAND\@xympar	300, 452
COMMAND\Aendnote	16, 26
COMMAND\Afootfmt	182
COMMAND\Afootgroup	182
COMMAND\Afootnote	8, 16, 24–26, 29, 138, 191, 216, 239, 256, 451
COMMAND\Afootstart	182
COMMAND\AtBeginDocument	279
COMMAND\AtEndEveryPend	19, 458
COMMAND\AtEveryPend	18, 19, 55, 155, 448, 450, 452, 458
COMMAND\AtEveryPend*	19
COMMAND\AtEveryPstart	18, 19, 55, 448, 450, 452, 455, 458
COMMAND\AtEveryPstart*	19
COMMAND\AtEveryStanza	55, 456, 458
COMMAND\AtEveryStopStanza	55, 456, 458
COMMAND\AtStartEveryPstart	19, 458
COMMAND\AtStartEveryStanza	55, 458
COMMAND\BeforeEveryStopStanza	458
COMMAND\Bendnote	16, 25
COMMAND\Bfootnote	8, 16, 216, 239, 256
COMMAND\Centering	48
COMMAND\Cfootnote	216
COMMAND\Columns	91, 151, 188
COMMAND\Dfootnote	216
COMMAND\Efootnote	216
COMMAND\Gls	68
COMMAND\Hy@raisedlink	379
COMMAND\Hy@raisedlink@left	379
COMMAND\LTR	47
COMMAND\NR	369
COMMAND\Pages	91, 151, 277, 278, 282
COMMAND\ProcessOptionsX	81
COMMAND\RL	46
COMMAND\RaggedLeft	48
COMMAND\RaggedRight	48
COMMAND\RenewExpandableDocumentCommand	34, 146
COMMAND\SEonlypage	293, 454
COMMAND\SEref	62–64, 293, 297, 455, 457
COMMAND\SErefonlypage	62–64, 391, 454
COMMAND\SErefwithpage	62, 64, 293, 297, 391, 454, 456
COMMAND\Stanza	447
COMMAND\Waklam	357
COMMAND\X@doreinfeet	278, 452
COMMAND\XR@prefix	300
COMMAND\XR@test	300
COMMAND\XR@test@mac	300
COMMAND\XR@test@mac@test	300
COMMAND\XXXXXXfmt	384
COMMAND\XXXXXXfmt	384
COMMAND\Xafterenumber	43

COMMAND\Xafterlemmaseparator	45, 388
COMMAND\Xafternote	48, 49, 387, 388
COMMAND\Xafternumber	43, 388
COMMAND\Xafterrule	50, 239, 388, 447, 450
COMMAND\Xaftersymlinenum	43, 388
COMMAND\Xarrangement	39, 49, 76, 183, 184, 268, 388
COMMAND\Xarrangement@footparagraph	189
COMMAND\Xarrangement@normal	184
COMMAND\Xarrangement@paragraph	189
COMMAND\Xbeforeinserting	47
COMMAND\Xbeforelemmaseparator	45, 388
COMMAND\Xbeforenotes	50, 239, 388, 447, 450
COMMAND\Xbeforenumber	41, 43, 388
COMMAND\Xbeforesymlinenum	43, 388
COMMAND\Xbhookendnote	389
COMMAND\Xbhookgroup	49, 454, 455
COMMAND\Xbhooknote	47, 389, 390, 452, 453
COMMAND\Xboxendlinenum	44, 389, 451
COMMAND\Xboxlinenum	43, 44, 389
COMMAND\Xboxlinenumalign	44, 389, 451
COMMAND\Xboxstartlinenum	44, 389, 451
COMMAND\Xboxsymlinenum	44, 389
COMMAND\Xcolalign	48, 450
COMMAND\Xdo@feet	452, 457
COMMAND\Xend	264
COMMAND\XendXafterlemmaseparator	389
COMMAND\XendXbeforelemmaseparator	389
COMMAND\XendXinplaceoflemmaseparator	389
COMMAND\XendXlemmaseparator	389
COMMAND\XendXmorethantwolines	389
COMMAND\XendXtwolines	389
COMMAND\Xendafterenumber	43, 453
COMMAND\Xendafterlemmaseparator	45, 389
COMMAND\Xendafternote	51, 455
COMMAND\Xendafternumber	44
COMMAND\Xendafterpagenumbe	456
COMMAND\Xendafterpagenumber	41, 64
COMMAND\XendafterpagenumberSErefonlypage	391
COMMAND\Xendaftersymlinenum	43, 44, 453
COMMAND\Xendahookinplaceofnumber	44, 453
COMMAND\Xendahooklinenumber	44, 453
COMMAND\Xendbeforelemmaseparator	45, 389
COMMAND\Xendbeforelinenumber	44
COMMAND\Xendbeforenumber	43, 453
COMMAND\Xendbeforepagenumber	41, 63, 64
COMMAND\XendbeforepagenumberSErefonlypage	63
COMMAND\Xendbeforesymlinenum	43, 44, 453
COMMAND\Xendbhookinplaceofnumber	44, 453
COMMAND\Xendbhooklinenumber	44, 453
COMMAND\Xendbhooknote	47

COMMAND\Xendboxendlinum	44, 451
COMMAND\Xendboxlinenum	44, 389, 449
COMMAND\Xendboxlinenumalign	44, 389, 451
COMMAND\Xendboxstartlinenum	44, 451
COMMAND\Xendboxsymlinenum	44, 453
COMMAND\Xendhangindent	47, 453, 455
COMMAND\Xendinplaceoflemmaseparator	26, 45, 389
COMMAND\Xendinplaceofnumber	43, 452
COMMAND\Xendinplaceofpagenumber	40, 457
COMMAND\Xendinsertsep@	246
COMMAND\Xendlemmadisablefontselection	46
COMMAND\Xendlemmafont	46, 454
COMMAND\Xendlemmaseparator	27, 45, 389
COMMAND\Xendletboxendlinum	389
COMMAND\Xendlineflag	64
COMMAND\Xendlinenumannotationposition	58, 64
COMMAND\Xendlineprefixmore	41, 64
COMMAND\Xendlineprefixsingle	41, 64
COMMAND\Xendlinerangeseparator	41, 64, 203, 391, 453
COMMAND\Xendmorethantwolines	26, 42, 64, 389, 450, 451
COMMAND\Xendnonumber	42, 452
COMMAND\Xendnote	242, 262–264, 450
COMMAND\Xendnotefontsize	46, 388
COMMAND\Xendnotenumfont	44, 45, 388
COMMAND\Xendnotes	245
COMMAND\Xendnumberonlyfirstinline	40, 453
COMMAND\Xendnumberonlyfirstintwolines	40, 453
COMMAND\Xendpagenumberonlyfirst	40, 457
COMMAND\Xendpagenumberonlyfirstifsingle	40, 457
COMMAND\Xendpagenumberonlyfirstintwo	40, 457
COMMAND\Xendparagraph	51, 447
COMMAND\Xendsep	51
COMMAND\Xendsublinesep	42, 64, 203
COMMAND\Xendsymlinenum	40, 453
COMMAND\Xendsympagenum	40, 457
COMMAND\Xendtwolines	26, 41, 42, 64, 389, 450, 451
COMMAND\Xendtwolinesbutnotmore	42, 64, 450, 451
COMMAND\Xendtwolinesonlyinsamepage	42, 64, 450, 451
COMMAND\Xendtxtbeforenotes	49, 459
COMMAND\Xendwrapcontent	47, 456
COMMAND\Xendwraplemma	46, 456
COMMAND\Xendwraplinenumannotation	58, 64
COMMAND\Xfootfmt	390
COMMAND\Xfootgroup	188
COMMAND\Xfootins	187
COMMAND\Xfootnote	60, 67, 136, 384, 444, 448–450, 454, 456
COMMAND\Xfootstarts	188
COMMAND\Xgroupbyline	49, 175, 215, 459, 460
COMMAND\Xgroupbylines	457
COMMAND\Xgroupbylineseparetwolines	49

COMMAND\Xhangindent	47, 453
COMMAND\Xhsize	391, 454, 455
COMMAND\Xhsizethreecol	48, 51, 389
COMMAND\Xhsizetwocol	48, 51, 269, 389
COMMAND\Xinplaceoflemmaseparator	26, 45, 389
COMMAND\Xinplaceofnumber	43, 389, 449, 451
COMMAND\Xinplaceofnnumber	26
COMMAND\Xinsertparafootsep	194, 195
COMMAND\Xledsetnormalparstuff	386, 387, 450
COMMAND\Xlemmadisablefontselection	46
COMMAND\Xlemmafont	46, 454
COMMAND\Xlemmaseparator	45, 203, 270, 273, 275, 389
COMMAND\Xlineflag	63, 391, 392, 456
COMMAND\Xlinenumannotationposition	57, 58, 63, 167
COMMAND\Xlinenumannotationposition@side	167
COMMAND\Xlinenumannotationpositionside	58
COMMAND\Xlinerrangeseparator	41, 63, 203, 391, 453
COMMAND\Xmaxhnotes	50, 75, 76, 388, 389, 447, 449
COMMAND\Xmorethantwolines	26, 41, 42, 63, 389, 449
COMMAND\Xnoindent	453
COMMAND\Xnolemmaseparator	45, 275, 389
COMMAND\Xnonbreakableaftersnumber	43, 389, 445
COMMAND\Xnonumber	42, 389
COMMAND\Xnonumberinfootnote	389
COMMAND\Xnotefontsize	46, 388
COMMAND\Xnotefontsize@(<s>)	194, 198, 199
COMMAND\Xnotenumfont	45, 388
COMMAND\Xnoteswidththiketwocolumns	51, 448
COMMAND\Xnumberonlyfirstinline	39–41, 49, 109, 205, 269, 270, 273, 389, 444, 449, 458
COMMAND\Xnumberonlyfirstintwolines	40, 49, 389, 444, 458
COMMAND\Xonlypstart	42, 389, 444, 449
COMMAND\Xpagelinesep	43, 457
COMMAND\Xparafootsep	48, 49, 109, 388, 389, 456, 458, 459
COMMAND\Xparafootsep@series	194
COMMAND\Xparindent	47, 391, 450, 453, 455
COMMAND\Xprenotes	50, 239, 456
COMMAND\Xprenotes@	187, 239, 444
COMMAND\Xpstart	42, 386, 389, 444, 449
COMMAND\Xpstarteverytime	42, 386, 389, 449
COMMAND\Xragged	49
COMMAND\Xrigidbalance	196, 391, 454
COMMAND\Xstanza	42, 56
COMMAND\Xstanzaseparator	42
COMMAND\Xstorelineinfo	205
COMMAND\Xsublinesep	22, 42, 43, 63, 203, 391
COMMAND\Xsublinesepsid	22, 43
COMMAND\Xsymlinenum	40, 49, 388, 389, 451, 458
COMMAND\Xtextbeforenotes	178
COMMAND\Xtoendnotes	27, 263
COMMAND\Xtwolines	26, 41, 42, 63, 211, 212, 269, 389, 449



COMMAND\Xtwolinesappref	269
COMMAND\Xtwolinesbutnotmore	41, 42, 63, 389, 450
COMMAND\Xtwolinesbutnotmoreappref	270
COMMAND\Xtwolinesonlyinsamepage	41, 42, 63, 389, 450
COMMAND\Xtxtbeforenotes	49, 389, 456, 457, 459
COMMAND\Xtxtbeforenotesonlyonce	49, 459
COMMAND\Xunvxh	192, 390
COMMAND\Xwidth	51, 391, 455
COMMAND\Xwrapcontent	47, 456
COMMAND\Xwraplemma	46, 47, 456
COMMAND\Xwraplinenumannotation	58, 63, 167
COMMAND\Xwraplinenumannotation@ref	167
COMMAND\Xwraplinenumannotation@side	167
COMMAND\Xwraplinenumannotationref	58
COMMAND\Xwraplinenumannotationside	58
COMMAND\&	388
COMMAND\absline@num	105, 106, 163
COMMAND\accent	135
COMMAND\actionlines@list	107, 165
COMMAND\actions@list	107
COMMAND\add@Xgroupbyline	175
COMMAND\add@inserts	106, 174, 175
COMMAND\add@inserts@next	174, 175
COMMAND\add@msd@	334
COMMAND\add@msdata	334, 335
COMMAND\add@msdata@firstlineofpage	337
COMMAND\add@msddata	334
COMMAND\add@penalties	163, 176
COMMAND\addcontentsline	365
COMMAND\addfootins	384, 387
COMMAND\addfootinsX	384, 387
COMMAND\addtoendnotes	263
COMMAND\advancelabel@refs	286
COMMAND\advanceline	23, 108, 116, 131, 452
COMMAND\affixline@num	169, 172, 173, 307, 442
COMMAND\affixpstart@num	173
COMMAND\afterXrule	388
COMMAND\afterXsymlinenum	388
COMMAND\aftergroup	134, 138
COMMAND\afterlemmaseparator	388
COMMAND\afternote	388
COMMAND\afternoteX	48, 387, 388
COMMAND\afternumberinfootnote	388
COMMAND\afterruleX	50, 447, 450
COMMAND\annot@list	108, 168
COMMAND\annotationref	58, 60
COMMAND\applabel	62, 288, 289, 450, 456
COMMAND\appref	60, 62–64, 293, 297, 454, 455
COMMAND\apprefprefixmore	63, 388
COMMAND\apprefprefixsingle	62, 388

COMMAND\apprefwithpage	62, 64, 293, 297, 451, 454
COMMAND\arrangementX	39, 76, 220, 268, 388
COMMAND\arrangementX@normal	226
COMMAND\article	15
COMMAND\at@every@pend	155
COMMAND\autopar	18, 151, 156, 157, 236, 443, 445, 446, 450, 460
COMMAND\ballast	76
COMMAND\ballast@count	163, 177
COMMAND\baselineskip	39, 190, 194
COMMAND\beforeXnotes	388
COMMAND\beforeXsymlinenum	388
COMMAND\beforeelectedchapter	10, 73, 366
COMMAND\beforeinsertingX	47
COMMAND\beforelemmaseparator	388
COMMAND\beforenotesX	50, 390, 446, 447, 450
COMMAND\beforenumberinfootnote	388
COMMAND\begin	342
COMMAND\beginnumbering	17–20, 91, 92, 95, 96, 105, 109, 110, 126, 156, 242, 333, 444, 447, 451, 452, 457, 458
COMMAND\beginnumberingR	91
COMMAND\bf	444
COMMAND\bfseries	45, 46, 444
COMMAND\bhookXnote	389
COMMAND\bhookgroupX	49, 454
COMMAND\bhooknoteX	47, 390, 452, 453
COMMAND\body	325
COMMAND\bodyfootmarkA	36
COMMAND\book	15
COMMAND\boxXendlinenum	389
COMMAND\boxXendlinenumalign	389
COMMAND\boxXendstartlinenum	389
COMMAND\boxendlinenum	389
COMMAND\boxlinefootnote	207
COMMAND\boxlinenum	389
COMMAND\boxlinenumalign	389
COMMAND\boxstartlinenum	389
COMMAND\boxsymlinenum	389
COMMAND\break	39, 192
COMMAND\brokenpenalty	176
COMMAND\c@page	127
COMMAND\c@page@page	127
COMMAND\centering	48
COMMAND\ch@ck@l@ck	442
COMMAND\ch@cksub@l@ck	172, 442
COMMAND\chapter	72, 367, 371, 447, 450, 452, 456
COMMAND\chaptermark	365
COMMAND\check@pb@in@verse	379
COMMAND\colalignX	48, 450
COMMAND\collect@body	342
COMMAND\color	456

COMMAND\colorbox	77
COMMAND\columns	51
COMMAND\columnwidth	190, 448
COMMAND\command names	269
COMMAND\continuousnumberingwithcolumns@sync@linenumber@singletext	166
COMMAND\copyright	135
COMMAND\correct@Xfootins@box	449
COMMAND\correct@footinsX@box	449
COMMAND\count	197, 198
COMMAND\critex	443
COMMAND\critext	142, 383, 384, 387
COMMAND\csname	81, 145, 146
COMMAND\csquotes	260
COMMAND\ctab	358, 363
COMMAND\ctabtext	363
COMMAND\current@annot	168
COMMAND\dc col	352
COMMAND\def	78, 94
COMMAND\detokenize	146
COMMAND\dimen	198
COMMAND\dimexpr	51
COMMAND\discretionary	191
COMMAND\displaywidowpenalty	176
COMMAND\do@Xfeet	277, 442, 452, 457
COMMAND\do@actions	163–165, 442
COMMAND\do@actions@fixedcode	442
COMMAND\do@actions@next	164, 165
COMMAND\do@ballast	163, 177
COMMAND\do@feet@custom@order	276
COMMAND\do@insidelinehook	445
COMMAND\do@line	106, 133, 153, 157, 161, 162, 174, 176, 324, 442, 443, 445, 447
COMMAND\do@linehook	442
COMMAND\do@lockoff	107
COMMAND\do@lockon	107
COMMAND\do@reinxtrafeet	441
COMMAND\doendnotes	26, 246, 390, 451, 458
COMMAND\doendnotesbysection	26, 246, 264, 451, 458
COMMAND\doennotes	458
COMMAND\doinsidelinehook	24, 447
COMMAND\doinsidethislinehook	24, 161, 460
COMMAND\dolinehook	24, 447
COMMAND\doreinxtrafeeti	452
COMMAND\doreinxtrafeetii	452
COMMAND\doxtrafeet	441
COMMAND\doxtrafeeti	452
COMMAND\doxtrafeetii	452
COMMAND\dummy@ref	134
COMMAND\edaftertab	71, 358, 359
COMMAND\edannotationref	291
COMMAND\edatleft	71, 356

COMMAND\edatright	71, 356
COMMAND\edbeforetab	71, 358
COMMAND\edfilldimen	357
COMMAND\edfont@info	141
COMMAND\edgls	68, 314
COMMAND\edglsadd	458
COMMAND\edgls...	455
COMMAND\edindex	34, 66–68, 314, 315, 318, 320, 346, 445, 447, 449, 452, 453, 457–459
COMMAND\edindexlab	68
COMMAND\edlabel	60–62, 64, 135, 283, 284, 286, 287, 290, 291, 299, 314, 346, 379, 392, 441, 444–446, 449, 454, 459
COMMAND\edlabelE	62, 289
COMMAND\edlabelS	62, 289
COMMAND\edlabelSE	62
COMMAND\edlineref	60, 284, 389, 449, 451, 454, 458
COMMAND\edmakelabel	62, 299
COMMAND\edpageref	60, 283, 284, 290, 299
COMMAND\edrowfill	357
COMMAND\edsublineref	60
COMMAND\edtabcolsep	351
COMMAND\edtext	7, 25, 28, 30, 31, 33, 52, 57, 60–62, 66, 69, 76, 94, 106, 108, 122, 124, 128, 129, 133–142, 144, 145, 147, 149, 167, 168, 288, 289, 293, 346, 348, 367, 379, 383, 384, 387, 441, 443, 445, 447–451, 457–460
COMMAND\edtext@level	451
COMMAND\edtextlater	125
COMMAND\edvertdots	72, 356
COMMAND\edvertline	71, 72, 356
COMMAND\elechapter	73
COMMAND\eled@sectioning@out	372
COMMAND\eledchapter	72, 387, 447, 452
COMMAND\eledchapter*	72
COMMAND\eledmac@error	441
COMMAND\eledsection	7, 16, 72, 134, 160, 366, 387, 449, 458, 460
COMMAND\eledsection*	72
COMMAND\eledsubsection	72, 387
COMMAND\eledsubsection*	72
COMMAND\eledsubsubsection	72, 387
COMMAND\eledsubsubsection*	72
COMMAND\eledxxx	10, 73, 372, 447
COMMAND\eledxxxx	365
COMMAND\elesection	372
COMMAND\else	313, 366
COMMAND\emph	33
COMMAND\empty	90, 170, 284
COMMAND\end	341, 342
COMMAND\end@lemmas	134
COMMAND\endashchar	203
COMMAND\endgraf	153, 194, 236
COMMAND\endlock	22, 108, 132, 329
COMMAND\endminipage	308, 310, 441, 452

COMMAND\endmsdata	35
COMMAND\endnotes	450, 454
COMMAND\endnumbering	17, 18, 20, 91, 92, 94, 95, 126, 442, 451, 457, 458
COMMAND\endnumberingR	91
COMMAND\endprint	242, 245, 264, 386
COMMAND\endstanzaextra	388
COMMAND\endsub	22, 107, 130
COMMAND\endverse	390
COMMAND\everypar	156
COMMAND\extensionchars	74, 92
COMMAND\externaldocument	64, 299
COMMAND\fax@l@cks	442
COMMAND\falseverse	387, 445, 447
COMMAND\fi	366
COMMAND\firstlinenum	21, 169, 443
COMMAND\firstsublinenum	21, 443
COMMAND\flag@end	128, 129, 140, 447
COMMAND\flag@end@RTL	129
COMMAND\flag@end@later	130
COMMAND\flag@start	128, 129, 140, 447, 448
COMMAND\flag@start@RTL	129
COMMAND\flag@start@later	130
COMMAND\flagstanza	59
COMMAND\floatingpenalty	183, 449
COMMAND\flush@notes	177, 178
COMMAND\fnpos	38, 238, 446, 457
COMMAND\footfmt	182, 185
COMMAND\footfmt...	221
COMMAND\footfootmarkA	36
COMMAND\footfudgefactor	192
COMMAND\footfudgefiddle	76, 190, 441
COMMAND\footgroup	182
COMMAND\footins	187
COMMAND\footnormal	268, 388, 441
COMMAND\footnormalX	388
COMMAND\footnote	36, 75, 216–218, 385, 442
COMMAND\footnote@lang	203
COMMAND\footnoteA	16, 36
COMMAND\footnoteB	16
COMMAND\footnoteC	25
COMMAND\footnoteE	36
COMMAND\footnoteX	9, 36, 260, 262, 457
COMMAND\footnoteX@reading	261
COMMAND\footnoteXmark	262, 458
COMMAND\footnoteXmk	275
COMMAND\footnoteXtext	262, 458
COMMAND\footnote⟨X⟩	135
COMMAND\footnote⟨X⟩mark	37
COMMAND\footnote⟨X⟩mk	37
COMMAND\footnote⟨X⟩nomk	37

COMMAND\footnote⟨X⟩text	37
COMMAND\footnotelang@lua	181
COMMAND\footnotelang@poly	181
COMMAND\footnotemark	37, 262
COMMAND\footnoteoption@	180, 453
COMMAND\footnoterule	197
COMMAND\footnotesize	46
COMMAND\footnotetext	37, 262
COMMAND\footparagraph	190, 268, 388, 447
COMMAND\footparagraphX	231, 388, 447
COMMAND\footsplitskips	442, 449
COMMAND\footstart	182, 187, 197
COMMAND\footstrut	194
COMMAND\foothreecol	388
COMMAND\foothreecolX	388, 450
COMMAND\foottwocol	388
COMMAND\foottwocolX	388, 450
COMMAND\foreignlanguage	47
COMMAND\fullstop	391
COMMAND\get@edindex@hyperref	319
COMMAND\get@edindex@ledinnote@command	315
COMMAND\get@fnmark	218
COMMAND\get@index@command	446
COMMAND\get@linelistfile	442
COMMAND\get@thisfootnote	225
COMMAND\getline@num	163, 164
COMMAND\gl@p	98
COMMAND\global	112
COMMAND\globaldefs	112
COMMAND\gls	68, 322
COMMAND\hangindentX	47, 450, 453
COMMAND\hangingsymbol	388, 443
COMMAND\hbox	191
COMMAND\hfill	445
COMMAND\hidenumbering	23, 121, 392, 450, 460
COMMAND\hidenumberingonleftpage	23, 121, 456
COMMAND\hidenumberingonrightpage	23, 121, 456
COMMAND\hline	69
COMMAND\hrulefill	357
COMMAND\hsize	39, 187, 190, 192, 198, 201, 237, 442, 448
COMMAND\hsizeX	391, 454, 455
COMMAND\hsizethreecol	389
COMMAND\hsizethreecolX	48, 51
COMMAND\hsizetwocol	389
COMMAND\hsizetwocolX	48, 51
COMMAND\hyperlinkR	319
COMMAND\hyperlinkformat	319
COMMAND\hyperlinkformatR	319
COMMAND\if@RTL	82, 138
COMMAND\if@edtext@	448, 451

COMMAND\if@eled@sectioning	372
COMMAND\if@firstlineofpage	83
COMMAND\if@firstlineofpageR	83
COMMAND\if@msd@options@fullpage	340
COMMAND\if@msdata@insertedfrompreviouspage	337
COMMAND\if@nobreak	155
COMMAND\if@noneed@Footnote	128
COMMAND\ifXnote@	91
COMMAND\ifafterendnumberingR	91
COMMAND\ifautopar	156
COMMAND\ifby@autopar	156
COMMAND\ifbypage@	98
COMMAND\ifbypage@R	98
COMMAND\ifbypstart@	98
COMMAND\ifbypstart@R	98
COMMAND\iffirst@linenum@out@	125, 126
COMMAND\ifindtl@innote	91
COMMAND\ifindtl@notenumber	91
COMMAND\ifinserthangingsymbol	324
COMMAND\ifinstanza	324
COMMAND\ifstwofollowinglines	212
COMMAND\ifl@d@Xmorethantwolines	208, 449
COMMAND\ifl@d@Xtwolines	208
COMMAND\ifl@d@dash	208
COMMAND\ifl@d@elin	208
COMMAND\ifl@d@esl	208
COMMAND\ifl@d@pnum	208
COMMAND\ifl@d@ssub	208
COMMAND\ifl@dend@X	262
COMMAND\ifl@dmemoir	441
COMMAND\ifl@dpaging	448
COMMAND\ifl@dpairing	90, 443
COMMAND\ifl@dprintingpages	449
COMMAND\ifl@dskipnumber	169
COMMAND\ifl@dstartendok	357
COMMAND\ifl@imakeidx	82
COMMAND\ifledRcol	91, 443
COMMAND\ifledRcol@	91, 447
COMMAND\iflemmacommand@	448
COMMAND\ifnoend@	247
COMMAND\ifnoledgroup@	313
COMMAND\ifnoteschanged@	109
COMMAND\ifnumberedpar@	151
COMMAND\ifnumbering	92, 95
COMMAND\ifnumberingR	91, 443
COMMAND\ifnumberline	140, 169
COMMAND\ifpst@rted	443
COMMAND\ifpst@rtedL	92
COMMAND\ifresumenummering@start	95
COMMAND\ifseriesbefore	266

COMMAND\ifstopmsdata@inserted@	333
COMMAND\ifsublines@	105, 118
COMMAND\iftrue	451
COMMAND\ifvmode	286
COMMAND\ifxxx	366
COMMAND\ignorespaces	138
COMMAND\imki@wrindexentry	82
COMMAND\immediate	125, 126, 241
COMMAND\indent	18, 156, 390
COMMAND\index	321, 322, 457
COMMAND\indtl@wrindexentry	82
COMMAND\initnumbering@quote	363, 452
COMMAND\initnumbering@reg	442
COMMAND\initnumbering@sectcmd	452
COMMAND\inplaceoflemmaseparator	389
COMMAND\inplaceofnumber	389
COMMAND\insert	140, 174, 182, 185, 199, 215, 221
COMMAND\insert@Xtxtbeforenotes	179, 199
COMMAND\insert@count	122, 128, 138
COMMAND\insert@countR	138
COMMAND\insert@msdata	334, 340
COMMAND\insert@txtbeforenotesX	179
COMMAND\insertthangingsymbol	445
COMMAND\insertlines@list	106, 122, 123
COMMAND\insertparafootsepX	235
COMMAND\inserts@list	133, 151, 174, 175, 191
COMMAND\interAfootnotelinepenalty	443
COMMAND\interfootnotelinepenalty	443
COMMAND\interlinepenalty	182
COMMAND\interparanoteglue	387
COMMAND\justifying	48
COMMAND\l@advance@parledegroupp@beforenormalnotes	452
COMMAND\l@d@@wrindexhyp	448
COMMAND\l@d@add	143
COMMAND\l@d@end	242, 262
COMMAND\l@d@nums	137, 140, 142, 143, 208, 209
COMMAND\l@d@section	242
COMMAND\l@d@set	117, 131
COMMAND\l@dampcount	348
COMMAND\l@dbfnote	218, 442
COMMAND\l@dcheckstartend	357
COMMAND\l@dchset@num	117
COMMAND\l@dcolcount	348, 349
COMMAND\l@dcollect@@body	342
COMMAND\l@dcollect@body	341
COMMAND\l@dcsnote	447
COMMAND\l@dcsnotetext	162, 304
COMMAND\l@dcsnotetext@l	162, 304
COMMAND\l@dcsnotetext@r	162, 304
COMMAND\l@ddodoreinextrafeet	277, 441



COMMAND\l@dextrafeet	276, 277, 441
COMMAND\l@emptyd@ta	442
COMMAND\l@dend@close	241
COMMAND\l@dend@open	241
COMMAND\l@dend@stuff	242
COMMAND\l@denvbody	342
COMMAND\l@dfeetbeginmini	441
COMMAND\l@dfeetendmini	441
COMMAND\l@dgetline@margin	443
COMMAND\l@dgetlock@disp	443
COMMAND\l@dgetref@num	292
COMMAND\l@dgetsidenote@margin	301, 443
COMMAND\l@dgobbeloptarg	448
COMMAND\l@dgobblearg	448
COMMAND\l@dgobbleoptarg	347
COMMAND\l@dlabel@parse	292
COMMAND\l@dld@ta	169, 171
COMMAND\l@dlp@rbox	306
COMMAND\l@dlsn@te	443
COMMAND\l@dlsnote	447
COMMAND\l@dmake@labels	286, 300
COMMAND\l@dmake@labelsR	300
COMMAND\l@dnumpstartsL	92, 443
COMMAND\l@dp@rsefootspec	209
COMMAND\l@dparsefootspec	209
COMMAND\l@dpush@begins	342
COMMAND\l@drd@ta	169, 171
COMMAND\l@dref@undefined	291
COMMAND\l@drsn@te	443
COMMAND\l@drsnote	447
COMMAND\l@dtabaddcols	357
COMMAND\l@dtabnoexpands	441
COMMAND\l@dumboxmpfoot	452
COMMAND\l@dunboxmpfoot	442
COMMAND\l@dunhbox@line	157
COMMAND\l@dzeropenalties	443, 448
COMMAND\l@pb	378
COMMAND\l@prev@nopb	377
COMMAND\l@prev@pb	377
COMMAND\l@reg	385
COMMAND\l@secondoptarg	347
COMMAND\l@wrap@ifnotemptybox	89
COMMAND\label	19, 62, 64, 68, 284, 285, 293
COMMAND\label@refs	284
COMMAND\labelstarttrue	19, 392, 444, 459
COMMAND\labelref@list	284, 287
COMMAND\language	191
COMMAND\lastbox	156
COMMAND\lastskip	130
COMMAND\latex@makecol	279

COMMAND\leavevmode	18, 156
COMMAND\led@check@nopb	378
COMMAND\led@check@pb	378
COMMAND\led@nopb	377–379
COMMAND\led@nopbnum	377, 378
COMMAND\led@pb	377–379
COMMAND\led@pb@macro	378
COMMAND\led@pbnum	377, 378
COMMAND\led@reinit@index@fornote	322
COMMAND\led@set@index@fornote	321
COMMAND\ledRflag	319
COMMAND\ledchapter	387, 444
COMMAND\ledfootinsdim	388
COMMAND\ledinnernote	65, 302, 447, 456
COMMAND\ledinnote	316, 386, 451
COMMAND\ledinnotemark	67, 386, 450
COMMAND\ledleftnote	65, 302
COMMAND\ledlinenum	104, 443
COMMAND\ledllfill	162
COMMAND\ledlsnotefontsetup	456
COMMAND\ledlsnotesep	65
COMMAND\ledlsnotewidth	65
COMMAND\lednopb	74, 377
COMMAND\lednopbinverse	379
COMMAND\lednopbinversetrue	54, 74
COMMAND\lednopbnum	377
COMMAND\ledouternote	65, 302, 447, 456
COMMAND\ledpb	74, 377, 387
COMMAND\ledpbnum	377
COMMAND\ledpbsetting	74, 378, 453
COMMAND\ledrightnote	65, 302
COMMAND\ledrsnotefontsetup	456
COMMAND\ledrsnotesep	65
COMMAND\ledrsnotewidth	65
COMMAND\ledsection	387
COMMAND\ledsectnomark	365
COMMAND\ledsectnotoc	365
COMMAND\ledsetnormalparstuff	386, 387, 450
COMMAND\ledsetnormalparstuff@common	236
COMMAND\ledsetnormalparstuffX	386, 387, 450
COMMAND\ledsidenote	65, 302, 304
COMMAND\ledsubsection	387
COMMAND\ledsubsubsection	387
COMMAND\ledxxx	447
COMMAND\left	71
COMMAND\leftctab	358
COMMAND\leftheadline	103
COMMAND\leftlinenum	22, 103, 441, 443
COMMAND\leftltab	358
COMMAND\leftnoteupfalse	65

COMMAND\leftstartnum	173
COMMAND\lefttab	358
COMMAND\leftsidenote	304
COMMAND\leftskip	187, 190, 192
COMMAND\lemma	3, 25, 28, 30, 31, 33, 133, 137, 138, 141, 142, 144, 383, 443, 444, 451, 452, 454
COMMAND\lemmaseparator	389
COMMAND\let	29, 52, 135, 140, 262, 279, 329, 441
COMMAND\letboxXendendlinum	389
COMMAND\line	196, 200
COMMAND\line@list	106, 108, 123, 140
COMMAND\line@list@stuff	92, 109, 110, 126, 441, 443
COMMAND\line@list@version	112
COMMAND\line@margin	100, 170, 301
COMMAND\line@num	105–107, 169, 441
COMMAND\line@set	142, 143
COMMAND\lineannot	57, 61, 143
COMMAND\lineannot@set	143
COMMAND\lineation	21, 98, 459, 460
COMMAND\linebreak	39
COMMAND\lineneation	455
COMMAND\linenum	25, 28, 57, 61, 62, 133, 142, 143, 290, 292, 299, 383
COMMAND\linenum@out	125, 284, 287
COMMAND\linenumannotation	57, 167
COMMAND\linenumberannotation	459
COMMAND\linenumberlist	21, 90, 170, 441
COMMAND\linenumberstyle	23, 103, 441, 458
COMMAND\linenumincrement	21, 443
COMMAND\linenummargin	21, 100, 301
COMMAND\linenumr@p	103, 441, 443
COMMAND\linenumrep	103, 443
COMMAND\linenumsep	22, 65, 103, 302
COMMAND\linerangesep@	275
COMMAND\lineref	283, 290, 299, 389, 449
COMMAND\list@clear	97
COMMAND\list@clearing@reg	443
COMMAND\list@create	97
COMMAND\lock@disp	102
COMMAND\lock@off	120
COMMAND\lock@on	119
COMMAND\lockdisp	22, 102
COMMAND\loop	178, 325
COMMAND\ltab	358, 359, 363
COMMAND\ltabtext	363
COMMAND\m@mmf@prepare	217
COMMAND\mainmatter	459, 460
COMMAND\makeatletter	161
COMMAND\makehboxoffhboxes	193, 194
COMMAND\makeindex	66, 318
COMMAND\makelabel	299
COMMAND\managestanza@modulo	326

COMMAND\marginpar	65, 75, 300, 301, 442
COMMAND\marginparwidth	65, 302
COMMAND\markboth	24, 161
COMMAND\mathchardef	325
COMMAND\maxhXnotes	389
COMMAND\maxhnotesX	50, 76, 388, 446, 447, 449, 450
COMMAND\maxlinesinpar@list	110
COMMAND\measurebody	360
COMMAND\measuredbody	360
COMMAND\memorybreak	20
COMMAND\metaXXXvfootnote	215
COMMAND\microtype	157
COMMAND\morenoexpands	76, 77, 133, 135
COMMAND\morethantwolines	389
COMMAND\mpfnpos	38, 238, 446, 457
COMMAND\mpnormalfootgroup	441
COMMAND\mpnormalvfootnote	442
COMMAND\msdata	34, 35, 332–334, 458
COMMAND\msdataposition	36
COMMAND\multfootsep	36, 217
COMMAND\multiplefootnotemarker	217
COMMAND\musixtex	446
COMMAND\n@num	443, 450
COMMAND\n@num@ref	450
COMMAND\new@line	127, 442
COMMAND\new@series	135
COMMAND\newcommand	29, 78, 94, 217, 286
COMMAND\newcommandx	29
COMMAND\newhookarg@specific	275
COMMAND\newhookcommand@series	269, 270, 450
COMMAND\newhookcommand@series@reload	270
COMMAND\newhookcommand@toggle@reload	270, 448
COMMAND\newhooktoggle@series	269, 270, 450
COMMAND\newhooktoggle@specific	275
COMMAND\newif	450
COMMAND\newline	39
COMMAND\newlinechar	263
COMMAND\newseries	37, 384, 387, 388
COMMAND\newseries@	254, 267
COMMAND\newverse	55, 59, 387, 447, 458
COMMAND\next	325
COMMAND\next@action	110
COMMAND\next@actionline	110
COMMAND\next@insert	175
COMMAND\next@line@list@stuff	126, 459
COMMAND\nl@regR	113
COMMAND\no@expands	76, 135, 141, 262, 441
COMMAND\noXlemmaseparator	389
COMMAND\nobreak	208
COMMAND\nocritical	254

COMMAND\noeledsec	73, 387
COMMAND\noendnotes	387
COMMAND\noexpand	385
COMMAND\nofamiliar	272
COMMAND\noindent	18, 19, 55, 156, 453
COMMAND\noindentX	453
COMMAND\nomk@	275
COMMAND\nonbreakableafternumber	389
COMMAND\nonumberinfootnote	389
COMMAND\norelax	52
COMMAND\normal@footnotemarkX	221
COMMAND\normal@page@break	377
COMMAND\normal@pars	236
COMMAND\normalbfnoteX	443
COMMAND\normalbodyfootmarkX	221
COMMAND\normalfont	457
COMMAND\normalfootfmt	52, 187, 194, 203, 242, 456
COMMAND\normalfootfmtX	222
COMMAND\normalfootfootmarkX	222
COMMAND\normalfootgroup	188
COMMAND\normalfootgroupX	223
COMMAND\normalfootnoterule	183
COMMAND\normalfootstart	187, 190
COMMAND\normalfootstartX	222
COMMAND\normalvfootnote	185, 186
COMMAND\normalvfootnote@inserted	185, 186
COMMAND\normalvfootnoteX	221
COMMAND\notbool	366
COMMAND\notfontsetup	388
COMMAND\notfontsizeX	46, 388
COMMAND\notenumfont	388
COMMAND\notenumfontX	46, 388
COMMAND\notesXwidthliketwocolumns	389
COMMAND\noteswidthliketwocolumnsX	51, 389, 448, 450
COMMAND\num@lines	151, 177
COMMAND\numberlinefalse	20, 460
COMMAND\numberlinetrue	20, 460
COMMAND\numberonlyfirstinline	267, 389
COMMAND\numberonlyfirstintwolines	389
COMMAND\numberpstartfalse	19
COMMAND\numberpstarttrue	19, 42, 386, 443, 452, 458, 460
COMMAND\numberstanza	42
COMMAND\numberstanzafalse	55
COMMAND\numberstanzatrue	55
COMMAND\numlabfont	22, 52, 104
COMMAND\none@line	151
COMMAND\nonehalfspacing	453
COMMAND\nonlyXpstart	389
COMMAND\nonlysideX	261
COMMAND\page@action	107, 117

COMMAND\page@start	107, 443
COMMAND\pagecontents	107
COMMAND\pagenumbering	282
COMMAND\pageparbreak	387
COMMAND\pageref	62, 290
COMMAND\par	28, 39, 156, 236
COMMAND\par@line	151, 177
COMMAND\par@patch@pagenumbering	282
COMMAND\par@patch@thepage	127, 282
COMMAND\para@footgroup	190
COMMAND\para@footgroupX	234
COMMAND\para@footsetup	190, 441
COMMAND\para@footsetupX	231, 441, 448
COMMAND\para@vfootnoteX	232
COMMAND\parafootfmt	193, 194, 456
COMMAND\parafootfmtX	233
COMMAND\parafootftm	195
COMMAND\parafootftmX	235
COMMAND\parafootftmsep	388
COMMAND\parafootsep	389, 446, 451
COMMAND\parafootsepX	48, 49, 109, 388, 389, 456
COMMAND\parafootstart	190
COMMAND\parafootstartX	232
COMMAND\paravfootnote	191, 194
COMMAND\parfillskip	193
COMMAND\parindent	453
COMMAND\parindentX	47, 391, 453, 455
COMMAND\parse@annot	168
COMMAND\parshape	76
COMMAND\parskip	19, 55, 156
COMMAND\pause@numbering	95
COMMAND\pausenumbering	20, 95, 110, 112, 157, 446, 448, 455
COMMAND\pausenumbering@page@num	95
COMMAND\penalty	193
COMMAND\pend	2, 7, 17–21, 23, 73, 131, 133, 136, 143, 151, 153–157, 173, 174, 385, 446, 447, 456–459
COMMAND\preXnotes	450, 456
COMMAND\prenotesX	50, 240, 450
COMMAND\prepare@Xgroupbyline	215
COMMAND\prepare@Xprenotes	239
COMMAND\prev@nopb	378
COMMAND\prev@pb	378
COMMAND\prevlineX	109
COMMAND\prevpageX@num	109
COMMAND\print@Xfootnoterule	450
COMMAND\print@Xnotes	277, 278
COMMAND\print@Xnotes@forpages	449
COMMAND\print@eledsection	160
COMMAND\print@footnoteXrule	450
COMMAND\print@leftmargin@eledsection	366

COMMAND\print@lemma	204
COMMAND\print@line	158
COMMAND\print@notesX@forpages	449
COMMAND\print@rightmargin@eledsection	366
COMMAND\printendlines	247, 253, 297, 391, 441, 443
COMMAND\printlinefootnote	205, 207, 449
COMMAND\printlinefootnotearea	207, 208, 449
COMMAND\printlinefootnotenumbers	205
COMMAND\printlines	108, 187, 203, 208, 209, 247, 297, 391, 441, 443, 449, 454
COMMAND\printpnum	386, 391
COMMAND\printpstart	204
COMMAND\protect	135, 385
COMMAND\providecommand	216, 441
COMMAND\pstarL	151
COMMAND\pstart	2, 7, 17–21, 23, 72, 73, 117, 131, 136, 140, 143, 151, 154–156, 160, 174, 371, 385, 443–445, 447, 448, 450–452, 455–460
COMMAND\pstartinfootnote	389
COMMAND\pstartinfootnoteeverytime	389
COMMAND\pstartnum	173
COMMAND\pstartref	60, 283, 291, 446, 458
COMMAND\pstarts	443, 444
COMMAND\pstarts@readL	151
COMMAND\pstarts@typeset@L	151
COMMAND\raggedX	49
COMMAND\raggedleft	48
COMMAND\raggedright	48
COMMAND\raw@text	151
COMMAND\rbracket	45
COMMAND\read@linelist	109, 111
COMMAND\ref	62, 64, 68
COMMAND\reformatted@	297
COMMAND\reformattedwithpage	297
COMMAND\relax	19, 117, 140, 164, 175, 329, 347, 385
COMMAND\renewcommand	76, 388, 391
COMMAND\reset@current@annot	168
COMMAND\reset@msd@options@iffullpage	340
COMMAND\resetprevline@	109
COMMAND\resetprevpage@	109
COMMAND\resumenumbering	20, 91, 95, 96, 110, 112, 157, 443, 447, 448, 455
COMMAND\right	71
COMMAND\rightctab	358
COMMAND\rightlinenum	22, 103, 441, 443
COMMAND\rightltab	359
COMMAND\rightnoteupfalse	65
COMMAND\rightrtab	359
COMMAND\rightsidenote	304
COMMAND\rightright	187, 190, 192, 193
COMMAND\rightstartnum	173
COMMAND\rigidbalance	196, 198, 200, 391, 454
COMMAND\rigidbalanceX	196, 391, 454

COMMAND\robustify	40
COMMAND\roman	352, 454
COMMAND\rtab	358, 359, 363
COMMAND\rtabtext	360, 363
COMMAND\sameparallelpagenum	127
COMMAND\sameword	16, 29–31, 33, 125, 143–147, 149, 449, 451, 453, 456, 459
COMMAND\sameword@inedtext	145
COMMAND\saweword	144
COMMAND\scriptsize	104
COMMAND\section	72, 443, 457
COMMAND\section@num	91
COMMAND\sectionmark	365
COMMAND\select@lemmafont	52, 180
COMMAND\series	254
COMMAND\series@	254
COMMAND\seriesatbegin	37, 266, 450
COMMAND\seriesatend	37, 266, 450
COMMAND\set@Xtxtbeforenotes	178
COMMAND\set@continuousnumberingforl	96
COMMAND\set@line	140
COMMAND\set@line@action	108, 118
COMMAND\set@txtbeforenotesX	178
COMMAND\setSErefonlypageprefixmore	63, 296, 455, 458
COMMAND\setSErefonlypageprefixsingle	63, 296, 455, 458
COMMAND\setSErefprefixmore	63
COMMAND\setSErefprefixsingle	63
COMMAND\setapprefprefixmore	63, 388
COMMAND\setapprefprefixsingle	63, 388, 454
COMMAND\setcommand@series	268
COMMAND\sethangingsymbol	54, 324, 388, 453
COMMAND\sethanginsymbol	53
COMMAND\setistwofollowinglines	212
COMMAND\setl@dlprbox	306
COMMAND\setline	23, 108, 112, 117, 131, 135, 154, 452
COMMAND\setlinenum	23, 112, 117, 131, 441, 459
COMMAND\setmsdatalabel	35
COMMAND\setmsdataposition	458
COMMAND\setmsdataseries	35
COMMAND\setprintendlines	247, 250, 443
COMMAND\setprintlines	209, 212, 247, 443
COMMAND\setsidenotese	66
COMMAND\setsidenotsep	388
COMMAND\setstanzaindent	326
COMMAND\setstanzaindents	53, 326, 385
COMMAND\setstanzapenalties	326
COMMAND\setstanzavalues	326
COMMAND\settoggle@series	267, 444, 448
COMMAND\showlemma	134, 441, 442
COMMAND\showwordrank	33, 145
COMMAND\sidenote@margin	442



COMMAND\sidenotemargin	65, 442, 447
COMMAND\sidenoteseq	388
COMMAND\sidesstartnumtrue	19
COMMAND\skip	187
COMMAND\skipnumbering	23, 120, 121, 132, 443, 450, 451
COMMAND\skipnumbering@reg	450
COMMAND\small	46
COMMAND\somemacro	33
COMMAND\special	13
COMMAND\splitmaxdepth	183, 198
COMMAND\splitoff	196
COMMAND\splittopskip	182, 198, 200
COMMAND\stanza	22, 23, 55, 59, 329, 388, 390, 453, 458
COMMAND\stanza@hang	328
COMMAND\stanza@line	328
COMMAND\stanzaindent	53, 54, 326, 449
COMMAND\stanzaindent*	54
COMMAND\stanzaindentbase	325
COMMAND\stanzanumwrapper	56
COMMAND\startlock	22, 108, 132, 329
COMMAND\startstanzahook	388
COMMAND\startsub	22, 107, 130
COMMAND\stopmsd	334
COMMAND\stopmsdata	35, 332, 333
COMMAND\store@annot@to@absline	168
COMMAND\strip@pt	190
COMMAND\strutbox	198
COMMAND\sub@action	107, 118
COMMAND\sub@lock	106
COMMAND\sub@off	115, 287
COMMAND\sub@on	115, 287
COMMAND\subline@num	105–107
COMMAND\sublinenum@rep	441
COMMAND\sublinenumberstyle	23, 103, 441
COMMAND\sublinenumincrement	21
COMMAND\sublinenumr@p	103, 441, 443
COMMAND\sublinenumrep	103, 443
COMMAND\sublineref	60, 283, 290
COMMAND\subsectionmark	365
COMMAND\sw@inthisedtext	137
COMMAND\sw@list@inedtext	141, 149
COMMAND\sw@txt	146
COMMAND\swnoexpands	33, 459
COMMAND\symlinenum	389
COMMAND\symplinenum	388
COMMAND\sza@penalty	328
COMMAND>tag	449
COMMAND\text	383
COMMAND\text<language>	47
COMMAND\textbeforenotesX	178

COMMAND\textcolor	77
COMMAND\textheight	76
COMMAND\textsuperscript	58, 89, 167
COMMAND\the	127, 441
COMMAND\thefootnoteA	36
COMMAND\thefootnoteX	445
COMMAND\thelabidx	320
COMMAND\thepage	112, 127, 282
COMMAND\thepstart	19, 392
COMMAND\thepstartL	444
COMMAND\thepstartR	444
COMMAND\thestanza	56
COMMAND\this@line@list@version	125
COMMAND\this@section@next@page@num	95
COMMAND\this@section@page@num	108
COMMAND\thisfootnote	225
COMMAND\threecol@begin@insert	199
COMMAND\threecolfootfmt	199, 456
COMMAND\threecolfootfmtX	230
COMMAND\threecolfootgroup	198
COMMAND\threecolfootgroupX	230
COMMAND\threecolfootsetup	197
COMMAND\threecolfootsetupX	229
COMMAND\threecolvfootnote	198
COMMAND\threecolvfootnote@inserted	198
COMMAND\threecolvfootnoteX	229
COMMAND\toendnotes	27, 245, 456
COMMAND\twocolfootfmt	456
COMMAND\twocolfootfmtX	227
COMMAND\twocolfootgroupX	228
COMMAND\twocolfootsetupX	227
COMMAND\twocolvfootnoteX	227
COMMAND\twolines	267, 389
COMMAND\twolines@A	267
COMMAND\twolines@B	267
COMMAND\twolines@C	267
COMMAND\twolinesbutnotmore	389
COMMAND\twolinesonlyinsamepage	389
COMMAND\xtbeforeXnotes	389
COMMAND\xtbeforenotesX	49, 459
COMMAND\xtbeforenotesonlyonceX	49, 459
COMMAND\uline	36, 37
COMMAND\unhbox	157, 191
COMMAND\unpenalty	193, 194
COMMAND\unskip	193
COMMAND\unvxh	193, 390
COMMAND\unvxhX	390
COMMAND\upbracefill	357
COMMAND\usingcritext	384, 387
COMMAND\usingedtext	384, 387

COMMAND\VAfootnote	182
COMMAND\adjust	129
COMMAND\variant	29
COMMAND\ vbox	153, 156, 191, 192, 196, 238
COMMAND\tfootnote	182, 187, 191, 198
COMMAND\vl@dbfnote	218, 442
COMMAND\vl@disnote	305
COMMAND\vl@dlsnote	305
COMMAND\vl@dosnote	305
COMMAND\vl@drsnote	305
COMMAND\vnnumfootnoteX	443
COMMAND\vsizer	50, 76
COMMAND\vsplit	176
COMMAND\waklam	357
COMMAND\waklamec	357
COMMAND\wapunktel	357
COMMAND\wastricht	357
COMMAND\widthX	51, 391, 455
COMMAND\wrap@edcrossref	289, 448
COMMAND\wrapcontentX	47, 456
COMMAND\wrapped@bodyfootmarkX	235
COMMAND\wrapped@footfootmarkX	235
COMMAND\x...	61
COMMAND\xabslineref	291
COMMAND\xannotationref	61
COMMAND\xdef	97, 329
COMMAND\xflagref	61, 291, 391, 454
COMMAND\xleft@appenditem	98, 134
COMMAND\xlineref	61, 391, 454
COMMAND\xpageref	61
COMMAND\xparse	34
COMMAND\xpstartref	61, 446, 459
COMMAND\xr	64
COMMAND\xright@appenditem	97, 98
COMMAND\xsublineref	61
COMMAND\xxref	61, 292, 293, 299, 446, 449, 456
COMMAND\xxxmatter	282
ENVIRONMENTastanza	454
ENVIRONMENTedarrayc	363
ENVIRONMENTedarrayl	363
ENVIRONMENTedarrayr	363
ENVIRONMENTedtabularc	363
ENVIRONMENTedtabularl	363
ENVIRONMENTedtabularr	363
ENVIRONMENTledgroup	82, 311, 391, 454
ENVIRONMENTledgroupsized	312
PACKAGE(r)(e)ledmac	37
PACKAGEEledmac	107, 317, 386, 387, 449, 450
PACKAGEEledpar	450
PACKAGEEtoolbox	81

PACKAGEParallel	393
PACKAGEREledmac	390, 391
PACKAGEamsgen	341
PACKAGEamsmath	341
PACKAGEbabel	47, 77, 352, 454
PACKAGEbiblatex	75
PACKAGEbidi	46, 47, 82, 453
PACKAGEccaption	90
PACKAGEcolor	77
PACKAGEcsquotes	458
PACKAGEedmac	1, 6, 11–14, 78, 208, 216, 284, 326, 383, 393, 441
PACKAGEedstanza	1, 14, 324
PACKAGEeledmac	1, 10, 11, 14, 16, 67, 216, 313, 317, 344, 367, 380, 384, 386–388, 445, 447, 449
PACKAGEeledpar	183, 393, 443, 447–449
PACKAGEetex	453
PACKAGEetoolbox	97, 143, 254, 267, 276, 304, 366, 377
PACKAGEfancyhdr	279, 457
PACKAGEfloatrow	75, 391
PACKAGEfootmisc	36, 77, 82, 216, 393, 457
PACKAGEgeometry	15
PACKAGEglossaries	68, 322, 454
PACKAGEhandout	448
PACKAGEhyperlink	261
PACKAGEhyperref	61, 137, 235, 285, 319, 369, 379, 446–448, 455, 458
PACKAGEifluatex	81
PACKAGEifxetex	81
PACKAGEimakeidx	66, 75, 82, 313, 317, 387, 445–447, 449
PACKAGEindextols	321
PACKAGEindextool	387
PACKAGEindextools	66, 75, 82, 91, 313, 317, 321, 387, 449, 454, 457
PACKAGEinputenc	145
PACKAGEledarab	77
PACKAGEledmac	1, 11, 14, 77, 97, 317, 383, 384, 387, 390
PACKAGEledpar	77
PACKAGEMemoir	82, 317, 387, 393, 447
PACKAGEMorewrites	75
PACKAGEMusixtex	447
PACKAGEperpage	454
PACKAGEpolyglossia	45, 77, 181, 203, 454
PACKAGERagged2e	48, 81
PACKAGEREledmac	1, 2, 6, 11, 12, 14–17, 20, 21, 24, 26, 28–30, 34, 36, 37, 40, 41, 44, 47, 48, 50, 52, 54, 56, 59–62, 64, 66–69, 73, 75–79, 98, 101, 107, 108, 111, 112, 115, 125–127, 135, 138, 140, 143, 144, 174, 184, 187, 192, 203, 216, 242, 254, 259, 267, 276, 282, 289, 291, 294, 299, 300, 317, 337, 344, 365, 366, 379, 387, 388, 390–392, 452, 455, 457
PACKAGEREledpar	1, 4, 6, 8, 16, 19, 37, 51, 61, 63, 64, 73–75, 77, 79, 90, 98, 110, 115, 126, 127, 138, 140, 151, 184, 188, 237, 238, 254, 260, 275, 277, 278, 282, 313, 324, 365, 453, 454, 456, 457, 459, 460
PACKAGESuffix	81
PACKAGETabmac	1, 14, 393
PACKAGEulem	36

PACKAGEuninormalize .....	30
PACKAGExargs .....	29, 81
PACKAGExkeyval .....	79, 275
PACKAGExparse .....	81, 460
PACKAGExr .....	5, 64, 300, 455
PACKAGExref .....	299
PACKAGExstring .....	81, 319

## A

\absline@num .....	1
Abu Kamil Shuja' b. Aslam .....	13
\actionlines@list .....	1
\actions@list .....	1
\add@inserts .....	1
\add@inserts@next .....	1
\add@msd@ .....	1
\add@msdata .....	1
\add@msdata@firstlineofpage .....	1
\add@penalties .....	1
\add@Xgroupbyline .....	1
\addtol@denbody .....	1
Adelard II .....	13
\advancelabel@refs .....	1
\advanceline .....	1, 22
\Aendnote .....	26
\affixline@num .....	1
\affixpstart@num .....	1
\affixside@note .....	1
\Afootnote .....	25
\afternoteX .....	48
\afterruleX .....	50
\ampersand .....	1, 59
\annot@list .....	1
\annotationref .....	1, 60
\append@notesX .....	1
\append@Xnotes .....	1
\applabel .....	1, 62
\appref .....	1, 62
\apprefwithpage .....	1, 62
\arrangementX .....	1, 39
\arrangementX@normal .....	1
\arrangementX@threecol .....	1
\arrangementX@twocol .....	1
\at@every@pend .....	1
\AtEndEveryPend .....	1, 19
\AtEveryPend .....	1, 19
\AtEveryPstart .....	1, 19
\AtEveryStanza .....	1, 55
\AtEveryStopStanza .....	1, 55
\AtStartEveryPstart .....	1, 19

\AtStartEveryStanza	1, 55
\autopar	1, 18

**B**

\ballast	76
\ballast@count	1
Beeton, Barbara Ann Neuhaus Friend	18
\beforeeledchapter	1
\BeforeEveryStopStanza	1
\beforeinsertingX	47
\beforeinsertion@X	1
\beforenotesX	50
\beginnumbering	1, 17
\Bendnote	26
\Bfootnote	25
\bhookgroupX	49
\bhooknoteX	47
\bodyfootmarkA	36
\boxfootnotenumbers	1
Bredon, Simon	13
Breger, Herbert	14, 344
Brey, Gerhard	13
Busard, Hubert L. L.	13
\bypage@false	1
\bypage@true	1
\bypstart@false	1
\bypstart@true	1

**C**

\c@addcolcount	1
\c@ballast	1
\c@firstlinenum	1
\c@firstsublinenum	1
\c@labidx	1
\c@linenumincrement	1
\c@sublinenumincrement	1
\Cendnote	26
\Cfootnote	25
\ch@ck@l@ck	1
\ch@cksub@l@ck	1
\chapter	1
\check@pb@in@verse	1
Chester, Robert of	13
Claassens, Geert H. M.	13
\colalignX	48
\continuousnumberingwithcolumns@sync@linenumber@singletext	1
Copernicus, Nicolaus	13
\critext	383
\ctab	1
\ctabtext	1

\current@annot . . . . . 1

## D

Dekker, Dirk-Jan . . . . . 77

\Dendnote . . . . . 26

\Dfootnote . . . . . 25

\disable@endnotes . . . . . 1

\disable@familiarnotes . . . . . 1

\disable@notes . . . . . 1

\disable@sidenotes . . . . . 1

\disablel@dtabfeet . . . . . 1

\do@actions . . . . . 1

\do@actions@fixedcode . . . . . 1

\do@actions@next . . . . . 1

\do@ballast . . . . . 1

\do@feet@custom@order . . . . . 1

\do@feetX . . . . . 1

\do@insidelinehook . . . . . 1

\do@line . . . . . 1

\do@linehook . . . . . 1

\do@lockoff . . . . . 1

\do@lockoffL . . . . . 1

\do@lockon . . . . . 1

\do@lockonL . . . . . 1

\do@Xfeet . . . . . 1

\doedindexlabel . . . . . 1

\doendnotes . . . . . 1, 26

\doendnotesbysection . . . . . 1, 26

\doinsidelinehook . . . . . 1, 24

\doinsidethislinehook . . . . . 1, 24

\dolinehook . . . . . 1, 24

\dosplits . . . . . 1

Downes, Michael . . . . . 76, 191, 193

\dummy@edindex . . . . . 1

\dummy@edtext . . . . . 1

\dummy@edtext@showlemma . . . . . 1

\dummy@ref . . . . . 1

## E

\edaftertab . . . . . 1, 71, 358

edarrayc (environment) . . . . . 69

edarrayl (environment) . . . . . 69

edarrayr (environment) . . . . . 69

\edatleft . . . . . 1, 71

\edatright . . . . . 1, 71

\edbforetab . . . . . 1, 71, 358

\edfilldimen . . . . . 1

\edfont@info . . . . . 1

\edindex . . . . . 1, 66

\edindexlab . . . . . 1, 68

<code>\EDLABEL</code>	1
<code>\edlabel</code>	1, 60
<code>\edlabelE</code>	1, 62
<code>\edlabelS</code>	1, 62
<code>\edlabelSE</code>	1, 62
<code>\edlineref</code>	1, 60
<code>\edmakelabel</code>	1, 61
<code>\edpageref</code>	1, 60
<code>\edrowfill</code>	1, 70
<code>\EDTAB</code>	1
<code>\edtabcolsep</code>	1, 69
<code>\EDTABINDENT</code>	1
<code>\edtabindent</code>	1
<code>\EDTABtext</code>	1
<code>edtabularc (environment)</code>	69
<code>edtabularl (environment)</code>	69
<code>edtabularr (environment)</code>	69
<code>\EDTEXT</code>	1
<code>\edtext</code>	1, 25
<code>\edvertdots</code>	1, 71
<code>\edvertline</code>	1, 71
<code>\Eendnote</code>	26
<code>\Efootnote</code>	25
<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, 22
<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, 22
<code>\endsubline@num</code>	1



environments:	
edarrayc	69
edarrayl	69
edarrayr	69
edtabularc	69
edtabularl	69
edtabularr	69
ledgroup	59
ledgroupsize	59
minipage	59
Euclid	13
\extensionchars	1, 74

## F

\f@x@l@cks	1
Fairbairns, Robin	36
\first@linenum@out@false	1
\first@linenum@out@true	1
\firstlinenum	1, 21
\firstseriesX@	1
\firstsublinenum	1, 21
\firstXseries@	1
\flag@end	1
\flag@end@later	1
\flag@end@RTL	1
\flag@start	1
\flag@start@later	1
\flag@start@RTL	1
\flagstanza	1, 59
\flush@notes	1
\fnpos	1, 37
Folkerts, Menso	13
\footfootmarkA	36
\footfudgefiddle	1, 76
\footnote	1
\footnoteA	36
\footnoteB	36
\footnoteC	36
\footnoteD	36
\footnoteE	36
\footnotelang@lua	1
\footnotelang@poly	1
\footnoteoptions@	1
\footnoteXmark	36
\footnoteXtext	36
\footsplitskips	1
\fullstop	1

## G

Gädeke, Nora	14
--------------	----

<code>\get@edindex@hyperref</code> .....	1
<code>\get@edindex@ledinnote@command</code> .....	1
<code>\get@fnmark</code> .....	1
<code>\get@fnmarkX</code> .....	1
<code>\get@index@command</code> .....	1
<code>\get@linelistfile</code> .....	1
<code>\get@sw@txt</code> .....	1
<code>\get@thisfootnote</code> .....	1
<code>\get@thisfootnoteX</code> .....	1
<code>\getline@num</code> .....	1
<code>\gl@p</code> .....	1

## H

<code>\h@num</code> .....	1
<code>\hangindentX</code> .....	47
<code>\hidenumbering</code> .....	1, 23
<code>\hidenumberingonleftpage</code> .....	1, 23
<code>\hidenumberingonrightpage</code> .....	1
<code>\Hilfsbox</code> .....	1
<code>\hilfsbox</code> .....	1
<code>\hilfscount</code> .....	1
<code>\HILFSskip</code> .....	1
<code>\Hilfsskip</code> .....	1
<code>\hilfsskip</code> .....	1
<code>\hsizethreecolX</code> .....	48
<code>\hsizetwocolX</code> .....	48
<code>\Hy@raisedlink@left</code> .....	1
<code>\hyperlinkformat</code> .....	1
<code>\hyperlinkformatR</code> .....	1
<code>\hyperlinkR</code> .....	1

## I

<code>\if@addsw</code> .....	1
<code>\if@edtext@secondarg@</code> .....	1
<code>\if@eled@sectioning</code> .....	1
<code>\if@firstlineofpage</code> .....	1
<code>\if@led@nofoot</code> .....	1
<code>\if@ledgroup</code> .....	1
<code>\if@lemmacommand@</code> .....	1
<code>\if@msdata@insertedfrompreviouspage</code> .....	1
<code>\if@noeled@sec</code> .....	1
<code>\if@noneed@Footnote</code> .....	1
<code>\if@RTL</code> .....	1
<code>\ifafterendnumberingR</code> .....	1
<code>\ifat@every@pend@star@</code> .....	1
<code>\ifat@every@pstart@star@</code> .....	1
<code>\ifautopar@pause</code> .....	1
<code>\ifbypage@</code> .....	1
<code>\ifbypage@R</code> .....	1
<code>\ifbypstart@</code> .....	1

<code>\ifbypstart@R</code> .....	1
<code>\ifeledmaccompat@</code> .....	1
<code>\iffirst@linenum@out@</code> .....	1
<code>\ifindtl@innote</code> .....	1
<code>\ifindtl@notenumber</code> .....	1
<code>\ifinserthangingsymbol</code> .....	1
<code>\ifinstanza</code> .....	1
<code>\ifl@d@dash</code> .....	1
<code>\ifl@d@elin</code> .....	1
<code>\ifl@d@esl</code> .....	1
<code>\ifl@d@pnum</code> .....	1
<code>\ifl@d@ssub</code> .....	1
<code>\ifl@d@Xmorethantwolines</code> .....	1
<code>\ifl@d@Xtwolines</code> .....	1
<code>\ifl@dend@X</code> .....	1
<code>\ifl@dhidenumber</code> .....	1
<code>\ifl@dmemoir</code> .....	1
<code>\ifl@dpaging</code> .....	1
<code>\ifl@dpairing</code> .....	1
<code>\ifl@dprintingcolumns</code> .....	1
<code>\ifl@dprintingpages</code> .....	1
<code>\ifl@dskipnumber</code> .....	1
<code>\ifl@dskipversenumber</code> .....	1
<code>\ifl@dstartendok</code> .....	1
<code>\ifl@footmisc</code> .....	1
<code>\ifl@imakeidx</code> .....	1
<code>\ifl@indextools</code> .....	1
<code>\ifledfinal</code> .....	1, 74
<code>\ifledgroupnotesL@</code> .....	1
<code>\ifledgroupnotesR@</code> .....	1
<code>\iflednopbinverse</code> .....	1
<code>\ifledRcol</code> .....	1
<code>\ifledRcol@</code> .....	1
<code>\ifnocritical@</code> .....	1
<code>\ifnoend@</code> .....	1
<code>\ifnofamiliar@</code> .....	1
<code>\ifnoledgroup@</code> .....	1
<code>\ifnoquotation@</code> .....	1
<code>\ifnoteschanged@</code> .....	1
<code>\ifnumberedpar@</code> .....	1
<code>\ifnumbering</code> .....	1
<code>\ifnumberingR</code> .....	1
<code>\ifnumberline</code> .....	1
<code>\ifnumberstanza</code> .....	1
<code>\ifparapparatus@</code> .....	1
<code>\ifparledgroup</code> .....	1
<code>\ifprevpgnotnumbered</code> .....	1
<code>\ifpst@rtedL</code> .....	1
<code>\ifresumenumbering@start</code> .....	1
<code>\ifsameparallelpagenunder</code> .....	1

<code>\ifseriesbefore</code> .....	1
<code>\ifsidepstartnum</code> .....	1
<code>\ifstopmsdata@inserted@</code> .....	1
<code>\ifsublines@</code> .....	1
<code>\ifwidthliketwocolumns</code> .....	1
<code>\ifXendinsertsep@</code> .....	1
<code>\ifxindy@</code> .....	1
<code>\ifxindyhyperref@</code> .....	1
<code>\initnumbering@quote</code> .....	1
<code>\initnumbering@reg</code> .....	1
<code>\insert@count</code> .....	0, 1
<code>\insert@msdata</code> .....	1
<code>\insert@txtbeforenotesX</code> .....	1
<code>\insert@Xtxtbeforenotes</code> .....	1
<code>\inserthangingymbol</code> .....	1
<code>\insertlines@list</code> .....	1
<code>\insertparafootsepX</code> .....	1
<code>\inserts@list</code> .....	1

## J

Jayaditya .....	14
-----------------	----

## K

Kabelschacht, Alois .....	178
---------------------------	-----

## 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@ddodoreinxtrafeet</code> .....	1
<code>\l@ddoxtrafeet</code> .....	1
<code>\l@dedbeginmini</code> .....	1
<code>\l@dedendmini</code> .....	1
<code>\l@emptyd@ta</code> .....	1

<code>\l@dend@close</code>	1
<code>\l@dend@open</code>	1
<code>\l@dend@stuff</code>	1
<code>\l@dend@Xfalse</code>	1
<code>\l@dend@Xtrue</code>	1
<code>\l@denbody</code>	1
<code>\l@desnote</code>	1
<code>\l@dfambeginmini</code>	1
<code>\l@dfamendmini</code>	1
<code>\l@dfeetbeginmini</code>	1
<code>\l@dfeetendmini</code>	1
<code>\l@getline@margin</code>	1
<code>\l@getlock@disp</code>	1
<code>\l@getref@num</code>	1
<code>\l@getsidenote@margin</code>	1
<code>\l@gobbleoptarg</code>	1
<code>\l@disnote</code>	1
<code>\l@dlabel@parse</code>	1
<code>\l@dld@ta</code>	1
<code>\l@dlp@rbox</code>	1
<code>\l@dlsn@te</code>	1
<code>\l@dlsnote</code>	1
<code>\l@dmake@labels</code>	1
<code>\l@dmodforedtext</code>	1
<code>\l@nullfills</code>	1
<code>\l@dnumpstartsL</code>	1
<code>\l@doldold@footnotetext</code>	1
<code>\l@dp@rsefootspec</code>	1
<code>\l@dparsedendline</code>	1
<code>\l@dparsedendpage</code>	1
<code>\l@dparsedendsub</code>	1
<code>\l@dparsedstartline</code>	1
<code>\l@dparsedstartpage</code>	1
<code>\l@dparsedstartsub</code>	1
<code>\l@dparsefootspec</code>	1
<code>\l@dpush@begins</code>	1
<code>\l@drd@ta</code>	1
<code>\l@dref@undefined</code>	1
<code>\l@drestorefills</code>	1
<code>\l@drestoreforedtext</code>	1
<code>\l@drp@rbox</code>	1
<code>\l@drrsn@te</code>	1
<code>\l@drrsnote</code>	1
<code>\l@dsetmaxcolwidth</code>	1
<code>\l@dskipnumberfalse</code>	1
<code>\l@dskipnumbertrue</code>	1
<code>\l@dtabaddcols</code>	1
<code>\l@dtabnoexpands</code>	1
<code>\l@dunboxmpfoot</code>	1
<code>\l@dunhbox@line</code>	1

<code>\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>	62
<code>\labelpstartfalse</code>	1
<code>\labelpstarttrue</code>	1, 19
<code>\labelref@list</code>	1
<code>\labelrefsparseline</code>	1
<code>\labelrefsparsesubline</code>	1
<code>\last@page@num</code>	1
Lavagnino, John	12
<code>\led@check@nopb</code>	1
<code>\led@check@pb</code>	1
<code>\led@err@AutoparNotNumbered</code>	1
<code>\led@err@BadAction</code>	1
<code>\led@err@edtextoutsidepstart</code>	1
<code>\led@err@EdtextWithoutFootnote</code>	1
<code>\led@err@FootnoteNotInSecondArgEdtext</code>	1
<code>\led@err@HighEndColumn</code>	1
<code>\led@err@LineationInNumbered</code>	1
<code>\led@err@LowStartColumn</code>	1
<code>\led@err@ManyLeftnotes</code>	1
<code>\led@err@ManyRightnotes</code>	1
<code>\led@err@ManySidenotes</code>	1
<code>\led@err@NumberingNotStarted</code>	1
<code>\led@err@NumberingShouldHaveStarted</code>	1
<code>\led@err@NumberingStarted</code>	1
<code>\led@err@NumberingWithoutPstart</code>	1
<code>\led@err@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

\led@pb	1
\led@pb@setting	1
\led@pbnum	1
\led@reinit@index@fornote	1
\led@set@index@fornote	1
\led@toksa	1
\led@toksb	1
\led@warn@AppLabelOutSecondArgEdtext	1
\led@warn@BadAction	1
\led@warn@BadAdvancelineLine	1
\led@warn@BadAdvancelineSubline	1
\led@warn@BadLineation	1
\led@warn@BadLinenummargin	1
\led@warn@BadLockdisp	1
\led@warn@BadSetline	1
\led@warn@BadSetlinenum	1
\led@warn@BadSidenotemargin	1
\led@warn@BadSublockdisp	1
\led@warn@DuplicateLabel	1
\led@warn@edinde@outsidenumbering	1
\led@warn@LineFileObsolete	1
\led@warn@NoFile	1
\led@warn@NoIndexFile	1
\led@warn@NoMarginpars	1
\led@warn@RefUndefined	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)	59
ledgroupsize (environment)	59
\ledinnernote	1, 65
\ledinnote	1
\ledinnotehyperpage	1
\ledinnotemark	1
\ledleftnote	1, 65
\ledlinenum	1
\ledllfill	1
\ledlsnotefontsetup	1, 65
\ledlsnotesep	1, 65
\ledlsnotewidth	1, 65
\lednopb	1, 73
\lednopbinversetrue	74
\lednopbnum	1
\ledouternote	1, 65
\ledpb	1, 73
\ledpbnum	1
\ledpbsetting	1, 74
\ledrightnote	1, 65

<code>\ledrlfill</code>	1
<code>\ledrsnotefontsetup</code>	1, 65
<code>\ledrsnotesep</code>	1, 65
<code>\ledrsnotewidth</code>	1, 65
<code>\ledsectnomark</code>	1
<code>\ledsectnotoc</code>	1
<code>\ledsetnormalparstuff@common</code>	1
<code>\ledsetnormalparstuffX</code>	1
<code>\ledsidenote</code>	1, 65
<code>\leftctab</code>	1
<code>\leftlinenum</code>	1, 22
<code>\leftltab</code>	1
<code>\leftnoteupfalse</code>	65
<code>\leftpstartnum</code>	1
<code>\leftrtab</code>	1
<code>Leibniz</code>	14
<code>\lemma</code>	1, 28
<code>\letsforverteilen</code>	1
<code>\line@list</code>	1
<code>\line@list@stuff</code>	1
<code>\line@list@version</code>	1
<code>\line@margin</code>	1
<code>\line@num</code>	1
<code>\line@set</code>	1
<code>\lineannot</code>	1, 57
<code>\lineannot@set</code>	1
<code>\lineation</code>	1, 21
<code>\linenum</code>	1, 28
<code>\linenum@out</code>	1
<code>\linenumannotation</code>	1, 56
<code>\linenumberlist</code>	1, 21
<code>\linenumberstyle</code>	1, 23
<code>\linenumincrement</code>	1, 21
<code>\linenummargin</code>	1, 21
<code>\linenumr@p</code>	1
<code>\linenumrep</code>	1
<code>\linenumsep</code>	1, 22
<code>\linerrangesep@</code>	1
<code>\list@clear</code>	1
<code>\list@clearing@reg</code>	1
<code>\list@create</code>	1
<code>\lock@disp</code>	1
<code>\lock@off</code>	1
<code>\lock@on</code>	1
<code>\lockdisp</code>	1, 22
<code>Lorch, Richard</code>	13
<code>\ltab</code>	1
<code>\ltabtext</code>	1
<code>Luecking, Dan</code>	81



## M

<code>\m@mmf@check</code> .....	1
<code>\m@mmf@prepare</code> .....	1
<code>\M@sect</code> .....	1
<code>\makehboxofhboxes</code> .....	1
<code>\managestanza@modulo</code> .....	1
<code>\maxhnotesX</code> .....	50
Mayer, Gyula .....	14
<code>\measurebody</code> .....	1
<code>\measurecell</code> .....	1
<code>\measuremrow</code> .....	1
<code>\measuretbody</code> .....	1
<code>\measuretcell</code> .....	1
<code>\measuretrow</code> .....	1
Middleton, Thomas .....	14, 105
<code>minipage</code> (environment) .....	59
Mittelbach, Frank .....	13
<code>\morenoexpands</code> .....	1, 76
<code>\mp@append@notesX</code> .....	1
<code>\mp@append@Xnotes</code> .....	1
<code>\mpfnpos</code> .....	1, 37
<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, 34
<code>\msdata@c</code> .....	1
<code>\msdata@cR</code> .....	1
<code>\multfootsep</code> .....	1, 36
<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>	73
<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>	46
<code>\notenumfontX</code>	46
<code>\noteschanged@false</code>	1
<code>\noteschanged@true</code>	1
<code>\noteswidthliketwocolumnsX</code>	51
<code>\nulledindex</code>	1
<code>\nullsetzen</code>	1
<code>\num@lines</code>	1
<code>\numberedpar@false</code>	1
<code>\numberedpar@true</code>	1
<code>\numberingfalse</code>	1
<code>\numberingtrue</code>	1
<code>\numberlinefalse</code>	20
<code>\numberlinetrue</code>	20
<code>\numberpstartfalse</code>	1, 19
<code>\numberpstarttrue</code>	1, 19
<code>\numberstanzafalse</code>	55
<code>\numberstanzatrue</code>	55
<code>\numlabfont</code>	1, 52

## O

<code>\old@hsize</code>	1
-------------------------	---

\one@line	1
optionafter	58
optionauxdir	16, 456
optionbefore	58
optioncontinuousnumberingwithcolumns	91, 159, 166, 455, 460
optioninnnote	454
optioninnote	454
optionlinerrangesep	275
optionnocritical	454
optionnoeledsec	371, 456
optionnoend	454
optionnopenalties	76
optionnotenumber	454
optionsameparallelpagenum	460
optionswcaseinsensitive	29, 459

## P

\page@action	1
\page@num	1
\pagelinesep	1, 67
\pageref	62
\par@line	1
\par@patch@pagenumbering	1
\par@patch@thepage	1
\para@footgroupX	1
\para@footsetup	1
\para@footsetupX	1
\para@vfootnoteX	1
\parafootfmt	1
\parafootfmtX	1
\parafootgroup	1
\parafootsepX	48
\parafootstart	1
\parafootstartX	1
\paravfootnote	1
\parindentX	47
\parse@annot	1
\pausenumbering	1, 20
\pausenumbering@page@num	1
\pend	1, 17
Plato of Tivoli	13
\postbodyfootmark	1
\prebodyfootmark	1
\prenotesX	50
\prepare@edindex@fornote	1
\prepare@prenotesX	1
\prepare@Xgroupbyline	1
\prepare@Xprenotes	1
\prev@nopb	1
\prev@pb	1

<code>\prevpage@num</code> .....	1
<code>\print@eledsection</code> .....	1
<code>\print@footnoteXrule</code> .....	1
<code>\print@leftmargin@eledsection</code> .....	1
<code>\print@lemma</code> .....	1
<code>\print@line</code> .....	1
<code>\print@notesX</code> .....	1
<code>\print@rightmargin@eledsection</code> .....	1
<code>\print@Xfootnoterule</code> .....	1
<code>\print@Xnotes</code> .....	1
<code>\printendlines</code> .....	1
<code>\printlineendnote</code> .....	1
<code>\printlineendnotearea</code> .....	1
<code>\printlinefootnote</code> .....	1
<code>\printlinefootnotearea</code> .....	1
<code>\printlinefootnotenumbers</code> .....	1
<code>\printlines</code> .....	1
<code>\printnpnum</code> .....	1
<code>\printpstart</code> .....	1
<code>\printsymlineendnotearea</code> .....	1
<code>\printsymlinefootnotearea</code> .....	1
<code>\printXafternumber</code> .....	1
<code>\printXbeforenumber</code> .....	1
<code>\pstart</code> .....	1, 17
<code>\pstarteref</code> .....	1
<code>\pstartnum</code> .....	1
<code>\pstartref</code> .....	60
<code>\pstarts@read@L</code> .....	1
<code>\pstarts@typeset@L</code> .....	1

## Q

<code>\quotation</code> .....	1
<code>\quote</code> .....	1

## R

<code>\raggedX</code> .....	49
<code>\raw@text</code> .....	1
<code>\rbracket</code> .....	1
<code>\read@linelist</code> .....	1
<code>\ref</code> .....	62
<code>\ref@reg@later</code> .....	1
<code>\Relax</code> .....	1
<code>\reledmac@error</code> .....	1
<code>\reledmac@warning</code> .....	1
<code>\removehboxes</code> .....	1
<code>\reset@current@annot</code> .....	1
<code>\reset@msd@options@iffullpage</code> .....	1
<code>\resetprevline@</code> .....	1, 109
<code>\resetprevpage@</code> .....	1
<code>\resetprevpage@num</code> .....	109

\restore@endnotes	1
\restore@familiarnotes	1
\restore@notes	1
\restore@sidenotes	1
\resumenumbering	1, 20
\rightctab	1
\rightlinenum	1, 22
\rightltab	1
\rightnoteupfalse	65
\rightrtab	1
\rightstartnum	1
\rigidbalance	1
\rigidbalanceX	1
\rtab	1
\rtabtext	1

## S

Sacrobosco	14
\sameword	1, 29
\sameword@inedtext	1
Schöpf, Rainer	13
\section@num	1
\select@lemmafnt	1
\select@lemmafnt	1, 52
\Seref	1, 62
\Serefonlypage	62
\Serefwithpage	1, 62
\series	1
\seriesatbegin	1, 37
\seriesatend	1, 37
\set@continuousnumberingforL	1
\set@line	1
\set@line@action	1
\set@txtbeforenotesX	1
\set@Xtxtbeforenotes	1
\setapprefprefixmore	62
\setapprefprefixsingle	62
\setcommand@series	1
\sethangingsymbol	1, 54
\setistwofollowinglines	1
\setl@dlp@rbox	1
\setl@drpr@box	1
\setline	1, 22
\setlinenum	1, 23
\setmcellcenter	1
\setmcellleft	1
\setmcellright	1
\setmrowcenter	1
\setmrowleft	1
\setmrowright	1

<code>\setmsdatalabel</code>	1, 35
<code>\setmsdataposition</code>	1, 36
<code>\setmsdataseries</code>	1, 35
<code>\setnoteswidthliketwocolumnsX@</code>	1
<code>\setnotesXpositionliketwocolumns@</code>	1
<code>\setprintendlines</code>	1
<code>\setprintlines</code>	1
<code>\setSErefonlypageprefixmore</code>	63
<code>\setSErefonlypageprefixsingle</code>	63
<code>\setSErefprefixmore</code>	63
<code>\setSErefprefixsingle</code>	63
<code>\setsidenotesepp</code>	66
<code>\setstanzaindents</code>	1, 52
<code>\setstanzapenalties</code>	1, 54
<code>\setstanzavalues</code>	1
<code>\settccllcenter</code>	1
<code>\settccllleft</code>	1
<code>\settccllright</code>	1
<code>\settoggle@series</code>	1
<code>\setthrowcenter</code>	1
<code>\setthrowleft</code>	1
<code>\setthrowright</code>	1
<code>\setXnotespositionliketwocolumns@</code>	1
<code>\setXnoteswidthliketwocolumns@</code>	1
<code>\showlemma</code>	1, 74
<code>\showwordrank</code>	1, 33
<code>\sidenote@margin</code>	1
<code>\sidenotemargin</code>	1, 65
<code>\sidepstartnumtrue</code>	19
<code>\skip@lockoff</code>	1
<code>\skipnumbering</code>	1, 23
<code>\splitoff</code>	1
<code>\spreadmath</code>	1, 70
<code>\spreadtext</code>	1, 70
<code>\stanza</code>	1, 52
<code>\stanza@count</code>	1
<code>\stanza@chang</code>	1
<code>\stanza@line</code>	1
<code>\stanzaindent</code>	1, 53
<code>\stanzaindent*</code>	1, 53
<code>\stanzaindentbase</code>	1, 52
<code>\stanzanumwrapper</code>	1, 56
<code>\startlock</code>	1, 22
<code>\startsub</code>	1, 22
<code>\stepl@dcclcount</code>	1
<code>\stopmsdata</code>	1, 35
<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, 23
<code>\sublinenumincrement</code>	1, 21
<code>\sublinenumr@p</code>	1
<code>\sublinenumrep</code>	1
<code>\sublineref</code>	1, 60
<code>\sublines@false</code>	1
<code>\sublines@true</code>	1
<code>\sublock@disp</code>	1
<code>\sublockdisp</code>	1
Sullivan, Wayne	13, 14, 76, 90, 95, 191, 192, 284, 324
<code>\sw@noexpand</code>	1
<code>\swnoexpands</code>	33
<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>	36
Theodosius	14
<code>\thepageline</code>	1
<code>\thepar@page</code>	1
<code>\thepstart</code>	1, 19
<code>\thestanza</code>	1, 56
<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, 27
<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> .....	49
<code>\txtbeforenotesX</code> .....	49

## 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
<code>\vnumfootnoteX</code> .....	1

## W

Whitney, Ron .....	13
<code>\widthX</code> .....	51
<code>\wrap@edcrossref</code> .....	1
<code>\wrapcontentX</code> .....	47
<code>\wrapped@bodyfootmarkX</code> .....	1
<code>\wrapped@footfootmarkX</code> .....	1
Wujastyk, Dominik .....	12

## X

<code>\X@atbegininsertion</code> .....	1
<code>\X@beforeinsertion</code> .....	1
<code>\X@doreinfeet</code> .....	1
<code>\xabslineref</code> .....	1
<code>\Xafterlemmaseparator</code> .....	45
<code>\Xafternote</code> .....	48
<code>\Xafternumber</code> .....	43
<code>\Xafterrule</code> .....	50
<code>\Xaftersymlinenum</code> .....	43
<code>\xannotationref</code> .....	1, 60
<code>\Xarrangement</code> .....	1, 39
<code>\Xarrangement@normal</code> .....	1
<code>\Xarrangement@paragraph</code> .....	1
<code>\Xarrangement@threecol</code> .....	1
<code>\Xarrangement@twocol</code> .....	1
<code>\Xbeforeinserting</code> .....	47
<code>\Xbeforelemmaseparator</code> .....	45
<code>\Xbeforenotes</code> .....	50



\Xbeforenumber	41, 43
\Xbeforenumlinenum	43
\Xhookgroup	49
\Xhooknote	47
\Xboxlinenum	43
\Xboxlinenumalign	44
\Xboxsymlinenum	44
\Xcolalign	48
\xedindex	1
\xedlabel	1
\xedtext	1
\Xendafterlemmaseparator	45
\Xendafternote	51
\Xendafterpagenumber	41
\Xendaftersymlinenum	43
\Xendaftrenumber	43
\Xendahookinplaceofnumber	44
\Xendahooklinenumber	44
\Xendbeforelemmaseparator	45
\Xendbeforenumber	43
\Xendbeforepagenumber	41
\Xendbeforesymlinenum	43
\Xendbhookinplaceofnumber	44
\Xendbhooklinenumber	44
\Xendbhooknote	47
\Xendboxendlinenumalign	44
\Xendboxlinenum	44
\Xendboxlinenumalign	44
\Xendboxstartlinenumalign	44
\Xendboxsymlinenum	44
\Xendhangindent	47
\Xendinplaceofflemmaseparator	45
\Xendinplaceofnumber	43
\Xendinplaceofpagenumber	40
\Xendlemmadisablefontselection	46
\Xendlemmafont	46
\Xendlemmaseparator	45
\Xendlinenumannotationposition	57
\Xendlineprefixmore	41
\Xendlineprefixsingle	41
\Xendlinerangeseparator	41
\Xendmorethantwolines	42
\Xendnonumber	42
\Xendnotefontsize	46
\Xendnotenumfont	45
\Xendnumberonlyfirstinline	40
\Xendnumberonlyfirstintwolines	40
\Xendpagenumberonlyfirst	40
\Xendparagraph	51
\Xendsep	51

<code>\Xendsublinesep</code>	42
<code>\Xendsymmlinenumber</code>	40
<code>\Xendsympagenumber</code>	40
<code>\Xendtwolines</code>	42
<code>\Xendtwolinesbutnotmore</code>	42
<code>\Xendtxtbeforenotes</code>	49
<code>\Xendwrapcontent</code>	47
<code>\xflagref</code>	1
<code>\Xgroupbyline</code>	49
<code>\Xgroupbylineseparatetwolines</code>	49
<code>\Xhangindent</code>	47
<code>\Xhsizethreecol</code>	48
<code>\Xhsizetwocol</code>	48
<code>\Xinplaceoflemmaseparator</code>	45
<code>\Xinplaceofnumber</code>	43
<code>\Xinsertparafootsep</code>	1
<code>\Xledsetnormalparstuff</code>	1
<code>\xleft@appenditem</code>	1
<code>\Xlemmadisablefontselection</code>	46
<code>\Xlemmafont</code>	46
<code>\Xlemmaseparator</code>	45
<code>\Xlinenumannotationposition</code>	57
<code>\Xlinenumannotationposition@side</code>	1
<code>\Xlinrangeseparator</code>	41
<code>\xlineref</code>	1, 60
<code>\Xmaxhnotes</code>	50
<code>\Xmorethantwolines</code>	41
<code>\Xnolemmaseparator</code>	1, 45
<code>\Xnonbreakableafternumber</code>	43
<code>\Xnonumber</code>	42
<code>\Xnotefontsize</code>	46
<code>\Xnotenumfont</code>	45
<code>\Xnoteswidthliketwocolumns</code>	51
<code>\Xnumberonlyfirstinline</code>	39
<code>\Xnumberonlyfirstintwolines</code>	39
<code>\Xonlypstart</code>	42
<code>\Xpagelinesep</code>	43
<code>\xpageref</code>	1, 60
<code>\Xparafootsep</code>	48
<code>\Xparindent</code>	47
<code>\Xprenotes</code>	1, 50
<code>\Xprenotes@</code>	1
<code>\Xpstart</code>	42
<code>\Xpstarteverytime</code>	42
<code>\xpstartref</code>	1, 60
<code>\XR@test</code>	1
<code>\XR@test@mac</code>	1
<code>\XR@test@mac@test</code>	1
<code>\Xragged</code>	49
<code>\xright@appenditem</code>	1

<code>\Xrigidbalance</code> .....	<u>1</u>
<code>\Xstanza</code> .....	42
<code>\Xstanzaseparator</code> .....	42
<code>\Xstorelineinfo</code> .....	<u>1</u>
<code>\xsublineref</code> .....	<u>1</u> , 60
<code>\Xsublinesep</code> .....	22, 42
<code>\Xsublinesep side</code> .....	22
<code>\Xsymlinenum</code> .....	40
<code>\Xtoendnotes</code> .....	27
<code>\Xtwolines</code> .....	41
<code>\Xtwolinesonlyinsamepage</code> .....	41
<code>\Xtxtbeforenotes</code> .....	49
<code>\Xtxtbeforenotesonlyonce</code> .....	49
<code>\Xunvxh</code> .....	<u>1</u>
<code>\Xwidth</code> .....	51
<code>\Xwrapcontent</code> .....	47
<code>\Xwrapendlemma</code> .....	46
<code>\Xwraplemma</code> .....	46
<code>\Xwraplinenumannotation@ref</code> .....	<u>1</u>
<code>\Xwraplinenumannotation@side</code> .....	<u>1</u>
<code>\xxref</code> .....	<u>1</u> , 61

**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	82
\morenoexpands: Added \l@dtabnoexpands to \no@expands	135
\reledmac@error: Added \eledmac@error and replaced error messages	83
General: Added tabmac code, and extended indexing	1
v0.2.1.	
\@lab: Removed page setting from \@lab	287
\edlabel: Tweaked \edlabel to get correct page numbers	284
\l@ddodoreinextrafeet: Renamed \dodoreinextrafeet to \l@ddodoreinextrafeet	277
\l@ddoxtrafeet: Renamed \doxtrafeet to \l@ddoxtrafeet	276
\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	135
General: Added text about normal labeling	62
Bug fixes and match with mempatch v1.8	1
Major changes to insert code when memoir is loaded	279
v0.2.2.	
\footfudgefiddle: Added \footfudgefiddle	190
\next@line@list@stuff: Added initial write of page number in \line@list@stuff	126
\para@footsetup: Added \footfudgefiddle to \para@footsetup	190
\para@footsetupX: Added \footfudgefiddle to \para@footsetupX	231
General: Improved paragraph footnotes	1
New Dekker example	1
Used \providecommand for \@gobblethree to avoid clash with the amsfonts package	89
v0.3.0.	
\@lab: Replaced \the\line@num by \linenumr@p\line@num in \@lab, and similar for sub-lines	287
\@nl@reg: Added a bunch of code to \@nl for handling \setlinenum	113
\ledlinenum: Added \linenumr@p and \sublinenumr@p to \leftlinenum and \rightlinenum	104
\linenumberlist: Added \linenumberlist mechanism	90
\printendlines: Added \linenumr@p and \sublinenumr@p to \printendlines	250
\printlines: Added \linenumr@p and \sublinenumr@p to \printlines	214
\sublinenumr@p: Added \linenumberstyle and \sublinenumberstyle	103
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	310
\Xarrangement@normal: Added minpage footnote setup to \footnormal	184
\edtext: Added \showlemma to \edtext	136
\l@dfeetendmini: Added \l@dfeetbeginmini, \l@dfeetendmini and all their supporting code	308
\mpnormalfootgroup: Added \mpnormalfootgroup	188

\mpnormalvfootnote: Added \mpnormalvfootnote	186
\showlemma: Added \showlemma	90
General: Added final/draft options	79
Added ledgroup environment	311
Added ledgroupsize environment	312
Added minipage, etc., support	1
v0.4.1.	
\do@Xfeet: Changed \do@Xfeet code for easier extensions	277
\edindex: Let eldmac take advantage of memoir's indexing	317
\print@Xnotes: Added \opXfeet	277
General: Added code for changing \doclearpage	280
Not released. Minor editorial improvements and code tweaks	1
Only change \@footnotetext and \@footnotemark if memoir not used	217
v0.5.0.	
\@footnotetext: Enabled regular \footnote in numbered text	218
\@xympar: Eliminated \marginpar disturbance	300
General: Added left and right sidenotes	301
Added sidenotes, familiar footnotes in numbered text	1
v0.5.1.	
\affixline@num: Changed \affixline@num to cater for sidenotes	169
\l@getsidenote@margin: Added \sidenotemargin and \sidenote@margin	301
General: Added moveable sidenote	301
Fixed right line numbers killed in v0.5	1
Only change \hsize in ledgroupsize environment otherwise page number can be in wrong place	312
v0.6.0.	
\@lopR: Added \@pend, \@pendR, \@lopL and \@lopR in anticipation of parallel processing	115
\@nl@reg: Extended \@nl to include the page number	113
\get@thisfootnote: Changed \l@dbfnote and \vl@dbfnote as originals could give incorrect markers in the footnotes	218
\l@the@c@page: Extended \new@line to output page numbers	127
General: Fixed long paragraphs looping	1
Fixed minor typos	1
Prepared for eldpar package	1
v0.7.0.	
\@nl@reg: Added \@nl@reg	113
\@ref@reg: Added \@ref@reg	123
\affixline@num: Added skipnumering to \affixline@num	169
\do@actions@fixedcode: Added \do@actions@fixedcode	165
\do@actions@next: Added number skipping to \do@actions	164
\do@insidelinehook: Added \do@linehook for use in \do@line	161
\endnumbering: Changed \endnumbering for eldpar	94
\fx@l@cks: Added \ch@cksub@l@ck, \ch@ck@l@ck and \fx@l@cks	172
\footsplitskips: Added \footsplitskips for use in many footnote styles	183
\get@linelistfile: Added \get@linelistfile	111
\initnumbering@reg: Added \initnumbering@reg	92
\l@advance@parledgroup@beforenormalnotes: Added \l@dunboxmpfoot containing some common code	311
\l@dcsnotetext@r: Added \l@emptyd@ta	162

\l@dgetline@margin: Added \l@dgetline@margin	100
\l@dgetlock@disp: Added \l@dgetlock@disp	102
\l@dgetsidenote@margin: Added \l@dgetsidenote@margin	301
\l@dnumpstartsL: Added \l@dnumpstartsL, \ifl@dpairing and \ifpst@rted for/from eledpar	90
\l@drsn@te: Added \l@dlsn@te and \l@drsn@te for use in \do@line	162
\l@dzeropenalties: Added \l@dzeropenalties	156
\ledlinenum: Added \ledlinenum for use by \leftlinenum and \rightlinenum	104
\list@clearing@reg: Added \list@clearing@reg	111
\n@num: Added \n@num	120
\next@line@list@stuff: Deleted \page@start from \line@list@stuff	126
\normalbfnoteX: Removed extraneous space from \normalbfnoteX	224
\pausenumbering@page@num: Changed \resumenumbering for eledpar	95
\setprintendlines: Added \setprintendlines for use by \printendlines	247
\setprintlines: Added \setprintlines for use by \printlines	209
\skipnumbering: Added \skipnumbering and supports	132
\sublinenumincrement: Added \firstlinenum, \linenumincrement, \firstsublinenum and \linenumincrement	101
\sublinenumr@p: Using \linenumrep instead of \linenumr@p	103
Using \sublinenumrep instead of \sublinenumr@p	103
\vnumfootnoteX: Removed extraneous space from \vnumfootnoteX	225
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	83
Replaced all \interAfootnotelinepenalty, etc., by just \interfootnotelinepenalty	1
Tidying up for eledpar and ledarab packages	1
v0.8.0.	
General: Bug on endnotes fixed: in a // text, all endnotes will print and be placed at the ends of columns ()	1
v0.8.1.	
General: Bug on \edtext ; \critex ; \lemma fixed: we can now us non-switching commands	1
v0.9.0.	
General: No more ledpatch. All patches are now in the main file.	1
v0.9.1.	
General: Fix some bugs linked to integrating ledpatch on the main file.	1
v0.10.0.	
General: Corrections to \section and other titles in numbered sections	1
v0.11.0.	
General: Makes it possible to add a symbol on each verse's hanging, as in French typography. Redefines the command \hangingsymbol to define the character.	1
v0.12.0.	
\l@dnumpstartsL: Added \ifledRcol and \ifnumberingR for/from eledpar	90
General: For compatibility with eledpar, possibility to use \autopar on the right side.	1
Possibility to number \pstart.	19
Possibility to number the pstart with the commands \numberpstarttrue.	1
v0.12.1.	
General: Don't number \pstarts of stanza.	1

The numbering of <code>\pstart</code> s 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. . . . .	326
General: New <code>stanzaindentsrepetition</code> counter to repeat stanza indents every $n$ verses. . . . .	53
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. . . . .	284
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. . . . .	169
<code>\ifinserthangingsymbol</code> : New management of <code>hangingsymbol</code> insertion, preventing undesirable insertions. . . . .	324
<code>\printlines</code> : Line numbering can be reset at each <code>pstart</code> . . . . .	213
General: Line numbering can be reset at each <code>pstart</code> . . . . .	98
Possibility to print <code>\pstart</code> number inside. . . . .	19
v0.17.0.	
<code>\ifinserthangingsymbol</code> : New new management of <code>hangingsymbol</code> insertion, preventing undesirable insertions. . . . .	324
v1.0.0.	
<code>\morenoexpands</code> : Change to be compatible with new features . . . . .	135
General: <code>\lemma</code> can contain commands. . . . .	28
Debug in lineation command . . . . .	21
New generic commands to customize footnote display. . . . .	38, 267
Options <code>nonum</code> and <code>nosep</code> in <code>\Xfootnote</code> . . . . .	25
Options of <code>\Xfootnotes</code> . . . . .	180
Possibility to have commands in sidenotes. . . . .	65
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. . . . .	39
v1.1.0.	
<code>\Xprenotes</code> : New skip <code>\Xprenotes@</code> . . . . .	239
<code>\settoggle@series</code> : <code>\settoggle@series</code> switch the global value of the toggle, not only the local value. . . . .	267
General: Add <code>\labelpstarttrue</code> . . . . .	19
Add <code>\Xnumberonlyfirstintwolines</code> . . . . .	39
Add <code>\Xpstart</code> and <code>\Xonlypstart</code> . . . . .	42
New hook to add arbitrary code at the beginning of the notes . . . . .	47
New options for block of notes. . . . .	49
New package option: <code>parapparatus</code> . . . . .	1
New tools to change order of series . . . . .	266
Sectioning commands. . . . .	72
v1.2.0.	
<code>\Xprenotes</code> : Debug in familiar footnotes (bug introduced by v1.1). . . . .	239
<code>\endquote</code> : Compatibility of <code>\ledchapter</code> with the <i>memoir</i> class. . . . .	363

v1.3.0.		
\endquote:	<i>Quotation</i> and quote environment inside numbered sections. . . . .	363
v1.4.0.		
\edtext:	Compatibility of \edtext with the right-to-left direction (with Polyglossia). . . . .	136
\ledsetnormalparstuffX:	Direction of footnotes with polyglossia. . . . .	236
\newseries@:	Remembers the language of the lemma, in order to create a correct direction for the footnote separator. . . . .	256
\rbracket:	Switch the right bracket to a left bracket when the lemma is RTL (needs polyglossia or LuaTeX). . . . .	203
General:	Compatibility with LuaTeX of RTL notes. . . . .	1
v1.4.1.		
\affixside@note:	Remove spurious spaces. . . . .	307
\endquote:	New option <i>noquotation</i> . . . . .	363
\get@thisfootnote:	Compatibility of standard footnotes with eledmac when these footnotes contain any commands. . . . .	218
\labelrefsparsesubline:	Fix a bug with \edlabel. . . . .	286
v1.4.2.		
General:	Debug with some special classes. . . . .	1
v1.4.3.		
General:	Add \Xnonbreakableafternumber. . . . .	43
	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. . . . .	363
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. . . . .	224
v1.5.0.		
\do@insidelinehook:	Added \do@insidelinehook for use in \do@line . . . . .	161
\edindex:	Compatibility with imakeidx package, and possibility to use multiple index with \edindex. . . . .	317
General:	Correct indexing when the call is made in critical notes. . . . .	313
v1.5.1.		
\managestanza@modulo:	Correct stanzaindentsrepetition counter . . . . .	326
\normalvfootnoteX:	Fix a bug with normal familiar footnotes when mixing RTL and LTR text. . . . .	221
v1.6.0.		
\newverse:	Add \falseverse macro. . . . .	329
v1.6.1.		
\AtStartEveryPstart:	Spurious space in \pstart. . . . .	151
\ifinserthangingsymbol:	Hang verse is now not automatically flush right. . . . .	324
\l@dunhbox@line:	Move the call to \inserthangingsymbol to allow use \hfill inside. . . . .	157



\pend: Spurious space in \pend. . . . .	153
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. . . . .	73
v1.8.0.	
\endquote: Correction of sectioning commands in parallel texts. . . . .	363
\get@index@command: Debug \get@index@command and compatibility with hyperref package. . . . .	316
\newhookcommand@series@reload: Debug \beforenotesX and \maxhnotesX which did not work. . . . .	270
\prevpage@num: Correct \parafootsep when using with ledgroup. . . . .	195
General: Compatibility with parledgroup option ofeledpar package. . . . .	1
If imakeidx and hyperref are loaded, adds hyperref in the index. . . . .	313
v1.8.1.	
General: Debug endnotes when more than one series is used (change the position where tools for endnotes are defined). . . . .	241
v1.8.2.	
General: Debug compatibility problem with hebrew option of babel package. . . . .	1
v1.8.3.	
General: Fixes spurious spaces added by v1.7.0. . . . .	1
v1.8.5.	
General: Debug indexing in right column, witheledpar. . . . .	313
v1.9.0.	
\l@ddoxtrafeet: Add \fnpos to choice the order of footnotes. . . . .	276
\l@dfheetendmini: Add \mpfnpos to choice the order of footnotes in minipage / ledgroup. . . . .	308
v1.10.0.	
\endquote: Correction of sectioning commands in parallel texts. . . . .	363
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 environnment. . . . .	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. . . . .	46
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. . . . .	112

\AtStartEveryPstart: New optional argument for \pstart, to execute code before it.	151
\edindex: Use correctly default index when imakeidx is loaded.	317
\endquote: \ledxxx sectioning commands are deprecated and replaced by \eledxxx commands.	363
\initnumbering@reg: \beginnumbering is defined only on eledmac, not on eledpar.	92
\l@dgetsidenote@margin: \sidenotemargin is now directly defined in eledmac to be able to manage eledpar.	301
\l@disnote: \l@dlsnote, \l@drsnote and \l@dcnote defined only one time, in eledmac, including needs for eledpar case.	302
\l@dumpstartsL: Add \ifledRcol@ for eledpar	90
\l@dunhbox@line: \do@line is split in more little commands.	158
\newhookcommand@series@reload: Debug \beforenotesX and \maxhnotesX which did not work when called after \footparagraphX.	270
Debug \Xbeforenotes and \Xmaxhnotes which did not work when called after \footparagraph.	270
\pend: New optional argument for \pend, to execute code after it.	153
\stanza: &can have an optional argument: content to be printed after.	329
\Stanza can have an optional argument: content to be printed before.	329
Add \newverse macro, \falseverse deprecated.	329
General: Add \ledinnernote and \ledouternote commands.	65
Add \Xendparagraph and related settings.	51
Add hyperlink to crossref (needs hyperref package).	60
Compatibility with musixtex.	1
Debug eledmac sectioning command after using \resumenumbering.	1
Ensure that imakeidx is loaded <i>before</i> eledmac	313
New hooks: \Xafterrule and \afterruleX	50
New options for ragged-paragraph notes	49
New sectioning commands.	72
Optional arguments for \pstart and \pend.	18
v1.12.1.	
\wrap@edcrossref: Fix spurious spaces.	289
v1.12.2.	
\l@dunhbox@line: Fix a bug with critical notes at the tops of pages (added by v12.0.0)	157
v1.12.3.	
\flag@end: \flag@start and \flag@end are now defined only one time for eledmac and eledpar	128
\flag@start send a error message when a \edtext is done without insert (note)	128
\reledmac@error: Replaced error messages	83
General: Add macros for new messages since v0.7	83
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.	65
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	319
\@ssect: Debug \eledchapter in parallel with memoir	367
\doinsidelinehook: Added \dolinehook and \doinsidelinehook	161

\endnumbering: Allow to mix parallel columns and normal text when using	
\pausenumbering	94
\l@dgobbleoptarg: \l@dgobblearg becomes \l@dgobbloptarg	347
\l@drestoreforedtext: Debug optional arguments of \Xfootnote in tabular context	347
\pausenumbering@page@num: Debug \resumenumbering	95
v1.12.7.	
\wrap@edcrossref: \wrap@edcrossref is now robust	289
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	128
v1.13.0.	
\newhooktoggle@series@reload: Add \newhookcommand@toggle@reload	270
\para@footsetupX: In \para@footsetupX, use \columnwidth instead of \hsize	231
\settoggle@series: \settoggle@series can take an optional arguments to reload	
series setup.	267
General: Add \Xnoteswidthliketwocolumns and \noteswidthliketwocolumnsX	51
Added widthliketwocolumns option	79
v1.13.1.	
\ifat@every@pstart@star@: Add \l@dzeroopenalties in \pstart	151
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	90
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	317
General: Debug spurious characters before endnotes.	241
Delete previous override of \l@d@wrindexhyp at the beginning of a document	
when hyperref is not loaded.	321
Move gobbling command	89
Provide \@gobblefour	89
v1.14.1.	
\@ssect: Debug sectioning commands when using both handout and hyperref	
package.	369
v1.14.2.	
\@ssect: Debug \edtext after starred sectioning commands when using memoir class.	367
v1.15.0.	
\@edtext@level: New boolean \if@edtext@.	136
\arrangementX@threecol: Correct bug with paragraphed familiar footnotes setting.	231
\endsub: Restore subline feature (disabled by mistake in v1.8.0).	130
\if@lemmacommand@: New boolean \iflemmacommand@.	142
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	29

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. . . . .
	126
v1.16.0.	
General:	\edtext is now defined only in eledmac, not in eledpar. Debug wrong numbering when using \sameword + eledpar + \tag command. . . . .
	136
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. . . . .
	290
v1.17.0.	
\edtext:	Error message when calling \edtext outside of a numbered paragraph. . . .
	136
v1.18.0.	
\@edindex@hyperref:	Fix spurious space with \edindex when using imakeidx/indextools + hyperref. . . . .
	319
\edlabel:	\edlabel is now defined only one time for both eledmac and eledpar . . . .
	284
\l@d@section:	Option parapparatus works for endnotes. . . . .
	242
\l@dnumpstartsL:	Add \ifl@dprintingpages and \@dprintingcolumns for eledpar . . . . .
	90
\print@line:	Compatibility with Lua $\TeX$ RTL languages. . . . .
	158
\printlinefootnote:	Code refactoring in \printlinefootnote: the printing of the numbers are factorized in \printlinefootnotearea . . . . .
	205
\printpstart:	Debug \Xpstart with parallel pages and columns (eledpar) . . . . .
	204
General:	Add \Xpstarteverytime . . . . .
	42
Compatibility with Lua $\TeX$ RTL languages.	. . . . .
	1
Debug \Xonlypstart when using \Xnumberonlyfirstinline and the current line number differs from the previous.	. . . . .
	42
v1.19.0.	
\footplitskips:	\footplitskips doesn't set \floatingpenalty to \@MM when processing parallel pages. . . . .
	183
\xxref:	\xxref works also with right side numbers, when \@Rlineflag is not empty. . . .
	292
General:	\Xmaxhnotes and \maxhnotesX work now for both two-columns and three-columns setting. . . . .
	1
Compatibility with eledpar v.1.13.0.	. . . . .
	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 . . . . .
	214
\stanza:	& and \& can be preceded by spaces. . . . .
	329
\xxref:	Debug \xxref when not loading eledpar (Fix a bug added in 1.19.0). . . . .
	292
General:	Add \Xendboxlinenum . . . . .
	44
Add \Xtwolines and \Xmorethantwolines hooks	. . . . .
	41
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 <code>\Xfootnote</code> .	25
The <code>\newifs</code> are not followed by boolean values set to false, because it is the $\TeX$ default setting.	1
v1.21.0.	
<code>\@edindex@hyperref</code> : Look at the hyperindex option of hyperref before inserting hyperref	319
<code>\l@d@section</code> : <code>\endnotes</code> take five arguments.	242
<code>\ledinnotemark</code> : Add <code>\ledinnotemark</code> .	316
<code>\ledsetnormalparstuffX</code> : <code>\ledsetnormalparstuff</code> is deprecated and becomes <code>\ledsetnormalparstuffX</code> and <code>\Xledsetnormalparstuff</code> .	236
<code>\n@num</code> : <code>\n@num@ref</code> deleted	120
<code>\n@num</code> defined only one time for both <code>Eledmac</code> and <code>Eledpar</code>	120
<code>\newhookcommand@series</code> : <code>\newhookcommand@series</code> can take an optional argument.	269
<code>\newhooktoggle@series</code> : <code>\newhooktoggle@series</code> can take an optional argument.	270
<code>\print@footnotexrule</code> : Code refactoring: the spaces after the footnote rules are directly managed in <code>\print@Xfootnoterule</code> and <code>\print@footnotexrule</code>	238
<code>\seriesatend</code> : Fix spurious space in <code>\seriesatend</code>	266
<code>\skipnumbering</code> : <code>\skipnumbering</code> defined only one time for both <code>Eledmac</code> and <code>Eledpar</code> .	132
Correct <code>\skipnumbering</code> for stanza.	132
Delete <code>\skipnumbering@reg</code> .	132
General: <code>\AtEveryPstart</code> and <code>\AtEveryPend</code> are now compatible with <code>\autopar</code>	1
<code>\Xafterrule</code> and <code>\afterruleX</code> features no longer create problems of overflowing at the bottom of the page.	1
<code>\chapter</code> inside optional argument of <code>\pstart</code> works when typesetting parallel pages	1
<code>\preXnotes</code> and <code>\prenotesX</code> features no longer create problems of overflowing at the bottom of the page.	1
<code>\seriesatbegin</code> and <code>\seriesatbegin</code> more efficient	266
Add <code>\applabel</code> and related	62
Add <code>\beforenotesX</code> and <code>\Xbeforenotes</code> features for notes set in two and three column.	1
Add <code>\hidenumbering</code>	23
Add <code>\Xcolalign</code> and <code>\colalignX</code> hooks	48
Add <code>\Xendtwolines</code> , <code>\Xendmorethantwolines</code> , <code>\Xendtwolinesbutnotmore</code> and <code>\Xendtwolinesonlyinsamepage</code> .	42
Add <code>\Xparindent</code> and <code>\hangindentX</code>	47
Add <code>\Xtwolinesbutnotmore</code> and <code>\Xtwolinesonlyinsamepage</code> .	1
Add <code>nocritical</code> , <code>noend</code> , <code>nofamiliar</code> and <code>noledgroup</code> options.	1
Add <code>noeledsec</code> package option	1
Debug <code>\beforenotesX</code> <code>\maxhnotesX</code> <code>\noteswidthliketwocolumnsX</code> and <code>\afterruleX</code> with footnotes set in two and three columns.	1
Fix a bug when a <code>\Xfootnote</code> follows a <code>\Xendnote</code> in the second argument of <code>\edtext</code> (bug added in <code>eledmac 1.0.0</code> ).	1
Fix a bug with <code>\maxhnotesX</code> when using <code>\foottwocolX</code> or <code>\footthreecolX</code> .	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>	89
v1.22.0.	
<code>\ledinnote</code> : <code>\ledinnote</code> takes a first optional argument, which is the label for hyperlinks.	316
General: Add <code>\doendnotesbysection</code> command.	26
Add option for lemma separator inside endnotes	45
Adds hyperlink for references to notes in indices.	1
Fix conflict between <code>noend</code> package option and <code>edtabularx</code> environments	1
Provides support for <code>xindy</code> .	1
Standardize endnotes handbook.	26
When using <code>hyperref</code> package, internal links in index or with <code>\edlineref</code> are now targeted to the top and not longer to the bottom of the lines they refer to.	1
v1.22.1.	
<code>\prevpage@num</code> : Correct double symbol when using both <code>\parafootsep</code> and <code>\Xsymlinenum</code> .	195
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> .	136
<code>\Serefwithpage</code> : Debug <code>\Xendtwolines</code> , <code>\Xendmoreethantwolines</code> , <code>\Xendtwolinesbutnotmore</code> and <code>\Xendtwolinesonlyinsamepage</code> when using <code>\apprefwithpage</code> .	297
<code>\lemma</code> : Fix spurious space after <code>\lemma</code> command	141
<code>\newseries@</code> : Prevent spurious spaces when <code>\Afootnote</code> and similar commands are followed by spaces (bug added on 1.0.0).	256
<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.	147
General: Add <code>\Xboxlinenumalign</code> and <code>\Xendboxlinenumalign</code> .	44
Add <code>\Xboxstartlinenum</code> , <code>\Xendboxstartlinenum</code> , <code>\Xboxendlinenum</code> , <code>\Xendboxendlinenum</code> .	44
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.	29
Fix a bug with <code>\sameword</code> when the same lemma is used for multiple notes or for nested <code>\edtexts</code> .	29
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.	29
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. . . . .	310
<code>\@xympar</code> : Patching <code>\@xympar</code> instead of redefining it . . . . .	300
<code>\endminipage</code> : Patch <code>\endminipage</code> instead of redefining it. . . . .	310
<code>\initnumbering@quote</code> : <code>\initnumbering@sectcmd</code> becomes <code>\initnumbering@quote</code> . . . . .	363
<code>\l@advance@parledgroup@beforenormalnotes</code> : Some conde of <code>\l@dumboxmpfoot</code> moved to <code>\l@advance@parledegroupp@beforenormalnotes</code> . . . . .	311
<code>\newseries@</code> : One endnotes file by series. . . . .	262
General: <code>\makecol</code> , <code>\@reinserts</code> and <code>\@doclearpage</code> are patched instead of begin redefined . . . . .	279
<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> . . . . .	279
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 . . . . .	217
Suppress indexing command specific to <code>memoir</code> . . . . .	317
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. . . . .	236
General: <code>\Xhangindent</code> and <code>\hangindentX</code> work now with all the paragraphs in the note. . . . .	1
<code>\Xnoindent</code> and <code>\noindentX</code> work now again (broken in 2.0.0). . . . .	1
Change some internal code in order to provide compatibility with $\TeX$ release of october 2015 . . . . .	1
Fix a bug which inserted double space before paragraphed familiar notes. . . . .	1
Fix a bug with <code>\edindex</code> when using not-Latin characters without UTF-8 engines . . . . .	1
v2.2.0.	
General: Fix a bug with combination of <code>\onehalfspacing</code> and two columns and three columns notes typeset. . . . .	1
Fix a bug with some setting command and optimization option. . . . .	1
Fix spurious space with paragraphed critical notes when using $\text{Lua}\TeX$ . . . . .	1
Increase line list version number to ensure compatibility with new options of <code>reledpar</code> package. . . . .	1
New setting tools for endnotes: <code>\Xendnumberonlyfirstinline</code> , <code>\Xendnumberonlyfirstintwolines</code> , <code>\Xendsymmlinenumber</code> , <code>\Xendbeforenumber</code> , <code>\Xendafternumber</code> , <code>\Xendbeforemsymmlinenumber</code> , <code>\Xendaftersymmlinenumber</code> , <code>\Xendboxsymmlinenumber</code> , <code>\Xendhangindent</code> , <code>\Xendbhooklinenumber</code> , <code>\Xendahooklinenumber</code> , <code>\Xendbhookinplaceofnumber</code> , <code>\Xendahookinplaceofnumber</code> . . . . .	1
v2.2.1.	
General: Compatibility with $\text{TeX}\text{format}$ 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. . . . .	180
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> . . . . .	41



Option <code>linrangesep</code> for critical footnotes and endnotes. . . . .	41
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> . . . . .	296
<code>\edlabel</code> : Fix a bug when calling <code>\edlabel</code> in a footnotes of the right side . . . . .	284
<code>\l@d@section</code> : <code>\endnotes</code> take six arguments. . . . .	242
<code>\printlines</code> : <code>\printlines</code> takes an eighth argument: the line flag . . . . .	213
<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. . . . .	290
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>\hookgroupX</code> . . . . .	1
New tools to easily make cross-reference to a passage defined by a start and an end line	62
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> . . . . .	196
<code>\l@d@section</code> : <code>\endnotes</code> take seven arguments. . . . .	242
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> . . . . .	51
New hooks: <code>\Xlemmafont</code> and <code>\Xendlemmafont</code> . . . . .	46

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> . . . . .	51
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. . . . .	242
General: <code>reledmac</code> cross-referencing can take advantage of <code>xr</code> package. . . . .	1
More <code>\edgls...</code> commands. . . . .	1
No indentation for paragraphed notes in <code>ledgroup</code> . Can be changed with <code>\Xparindent</code> and <code>\parindentX</code> . . . . .	1
v2.8.1.	
General: Warnings for undefined labels are really parsable by latexmk . . . . .	1
v2.8.2.	
General: Fix a bug concerning indent in a paragraph immediately following a sectioning command (bug NOT fixed on <code>reledpar</code> ) . . . . .	1
Fix a bug with <code>\AtEveryPstart</code> added in version 2.0.0. . . . .	1
Fix a bug with vertical space after the between-sectioning command as optional argument of a <code>\pstart</code> and <code>\pstart</code> content . . . . .	1
v2.9.0.	
General: Allow continuing line numbering between normal text and parallel text, using <code>\pausenumbering</code> and <code>\resumenumbering</code> and the <code>continuousnumberingwithcolumns</code> option. . . . .	1
Fix a bug when using <code>\lineneation{page}</code> and <code>\pausenumbering... \resumenumbering</code> . . . . .	1
Fix a bug with three- and two-column footnote setting (added in v2.4.0). . . . .	1
Fix spurious space inside three-column familiar footnote. . . . .	1
Write correct metadata in numbered files when using <code>\pausenumbering... \resumenumbering</code> . . . . .	1
v2.9.1.	
General: Fix a bug when notes start with “plus” or “minus”. . . . .	1
v2.9.2.	
General: Fix a bug with <code>hyperref</code> package when a lemma starts with “plus” or “minus” (bug introduced in v. 2.7.0). . . . .	1
v2.9.3.	
General: Fix a bug with line number position and reset added by v. 2.9.0 . . . . .	1

v2.10.0.	
\print@lemma: Code refactoring between \parafootfmt, \twocolfootfmt, \threecolfootfmt and \normalfootfmt. . . . .	204
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. . . . .	47
New hooks: \Xwraplemma and \Xendwraplemma . . . . .	46
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 . . . . .	165
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>	279
Add <code>\Xendpagenumberonlyfirst</code> , <code>\Xendpagenumberonlyfirstifsingle</code> , <code>\Xendpagenumberonlyfirstintwo</code> , <code>\Xendinplaceofpagenumber</code> and <code>\Xendsympagenum</code> hooks.	1
Add <code>\Xpagelinesep</code> hook.	43
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.	279
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.	126
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>\linenumberannotation</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>\linenumberannotation</code> in parallel typesetting	1
Fix bug with <code>\Xgroupbyline</code> and <code>\Xparafootsep</code> .	1

Fix bug with <code>\Xgroupbyline</code> and optional argument of critical footnotes. . . . .	1
Fix bug with <code>\Xgroupbyline</code> for the series after the first one. . . . .	1
v2.26.3.	
General: Fix bug added by v2.26.0 with notes in parallel typesetting . . . . .	1
v2.26.4.	
General: Fix bug when using <code>\numberpstarttrue</code> and alternating between manual <code>\pstart</code> and automatic <code>\pstart</code> produced by <code>\autopar</code> . . . . .	1
v2.26.5.	
General: Check that users have a enough recent version of <code>xparse</code> . . . . .	1
Fix bug when using <code>\eledsection</code> with the <code>continuousnumberingwithcolumns</code> of <code>reledpar</code> . . . . .	1
Fix bug with <code>continuousnumberingwithcolumns</code> . . . . .	1
Fix bug with using <code>\eledsection</code> and related when alternate normal typesetting and parallel typesetting . . . . .	1
v2.26.6.	
General: Fix bug with <code>\lineation</code> by page and nexted <code>\edtext</code> between two pages (added in v. 2.24.0) . . . . .	1
v2.26.7.	
General: Fix page number in footnote and endnotes when using <code>sameparallelpagenumber</code> option of <code>reledpar</code> and typesetting critical text in single mode after critical text in double page mode . . . . .	1
v2.26.8.	
General: Fix bug in some specific cases, introduced by v.2.27.7 . . . . .	1
v2.26.9.	
General: Fix bug with critical footnotes line number when alternating between <code>\numberlinefalse</code> and <code>\numberlinetrue</code> . . . . .	1
Fix bug with page number in footnote after <code>\mainmatter</code> (bug introduced in v2.26.7) . . . . .	1
Workaround with a bug of <code>TeX2018/04/01</code> patch level 3 to generate the .sty file from a .ins file . . . . .	1
v2.27.0.	
General: Add <code>\doinsidethislinehook</code> . . . . .	1
v2.27.1.	
General: Fix spurious space in <code>\hidenumbering</code> . . . . .	1