

# reledmac

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

Maïeul Rouquette<sup>†</sup>

based on the ledmac by

Peter Wilson

Herries Press

with the EDMAC plugins tabmac and edstanza by

Herbert Breger and Wayne Sullivan, all based on EDMAC by

John Lavagnino and Dominik Wujastyk

### Abstract

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

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

reledmac provides many tools and options. Normally, they are all documented in this file. Also provided is a help folder, “examples”. The folder contains additional examples (although not for every possible case). Examples starting with “1-” are for basic uses, those starting with “2-” are for advanced uses.

To report bugs or request a new feature, please go to ledmac GitHub page and click on “New Issue”: <https://github.com/maieul/ledmac/issues/>. You must create an account on github.com to access my page (maieul/ledmac). GitHub accounts are free for open-source users. You can post messages in English or in French (preferred).

You can subscribe to the reledmac mail list at:

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

## Contents

<b>1 Introduction</b>	<b>11</b>
1.1 Aim of the package	11
1.2 History	12
1.2.1 EDMAC	12
1.2.2 ledmac	13
1.2.3 eledmac	14
1.2.4 reledmac	14
1.3 Bibliography	14
1.4 Acknowledgements	14

---

\*This file (reledmac.dtx) has version number v2.39.7, last revised 2023/08/21.

<sup>†</sup>maieul at maieul dot net

<b>2 How the package works – the problem of the number of L<sup>A</sup>T<sub>E</sub>X runs</b>	<b>14</b>
<b>3 Compatibility warning</b>	<b>15</b>
<b>4 Options</b>	<b>15</b>
4.1 Specific features . . . . .	15
4.2 Optimizing package performance . . . . .	16
<b>5 Text lines and paragraphs numbering</b>	<b>16</b>
5.1 Text lines numbering . . . . .	16
5.2 Paragraphs . . . . .	17
5.2.1 Basics . . . . .	17
5.2.2 Automatically producing <code>\pstart ... \pend</code> . . . . .	17
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> . . . . .	18
5.2.5 Numbering paragraphs ( <code>\pstart</code> ) . . . . .	19
5.2.6 Languages written in Right to Left . . . . .	19
5.2.7 Memory limits . . . . .	19
5.3 Lineation commands . . . . .	20
5.3.1 Disabling lineation . . . . .	20
5.3.2 Setting lineation start and step . . . . .	20
5.3.3 Setting lineation reset . . . . .	21
5.3.4 Setting line number margin . . . . .	21
5.3.5 Other settings . . . . .	21
5.4 Changing the line numbers . . . . .	21
5.4.1 Sublineation . . . . .	21
5.4.2 Locking lineation . . . . .	22
5.4.3 Setting and changing line number . . . . .	22
5.4.4 Line number style . . . . .	22
5.4.5 Skipping and hiding number . . . . .	23
5.5 Adding annotations to line numbers . . . . .	23
5.5.1 Resetting annotation . . . . .	24
5.5.2 Modifying annotation associated with a specific note . . . . .	25
5.5.3 Changing the position of the annotation . . . . .	25
5.5.4 Changing the macro in which annotations are wrapped . . . . .	25
5.5.5 Not printing the second identical annotations . . . . .	26
5.5.6 Separator between annotations . . . . .	26
5.5.7 Annotations in the apparatus . . . . .	26
5.6 Executing code at each line . . . . .	26
5.7 Executing code at a specific line . . . . .	26
<b>6 Apparatus commands</b>	<b>27</b>
6.1 Terminology . . . . .	27
6.2 Critical notes . . . . .	27
6.2.1 The lemma . . . . .	27
6.2.2 Footnotes . . . . .	28
6.2.3 Endnotes . . . . .	28
6.2.4 Paragraph in critical apparatus . . . . .	30
6.2.5 Change lemma and line number . . . . .	30
6.2.6 Changing the names of commands for critical apparatus . . . . .	31

6.3 Disambiguation of identical words in the apparatus	31
6.3.1 Basic use	31
6.3.2 Case setting	32
6.3.3 Notes about input encoding with UTF-8 processor	32
6.3.4 Notes about right-to-left texts	32
6.3.5 Use with <code>\lemma</code> command	32
6.3.6 Sameword for a group of words	34
6.3.7 Customizing	35
6.3.8 Problems with some macros	35
6.3.9 Automatic sameword annotation	36
6.4 Apparatus of manuscripts	36
6.4.1 Marking sections of text	36
6.4.2 Layout of the apparatus of manuscripts	37
6.4.3 Settings	37
6.5 Familiar notes	37
6.5.1 Basic use	37
6.5.2 Customizing mark	38
6.5.3 Separator for multiple footnotes	38
6.6 Printing the footnote mark without printing the footnote text	38
6.7 Changing series	39
6.7.1 Create a new series	39
6.7.2 Delete series	39
6.7.3 Series order	39
6.8 Position of critical and familiar footnotes	39
<b>7 Apparatus customization</b>	<b>40</b>
7.1 Introduction	40
7.2 Notes arrangement in a series	40
7.3 Disabling footnote	41
7.4 Control number printing	41
7.4.1 Print line number only at first time	41
7.4.2 Print page number only at first time	42
7.4.3 Arbitrary text before line number	42
7.4.4 Separator for line range	43
7.4.5 Abbreviate line range	43
7.4.6 Disabling line number printing	43
7.4.7 Disabling page number printing	43
7.4.8 Printing pstart number	44
7.4.9 Printing stanza number	44
7.4.10 Options for line number annotations	44
7.4.11 Separator between line and subline numbers	45
7.4.12 Separator between page and line numbers	45
7.4.13 Space around number	45
7.4.14 Space around line symbol	45
7.4.15 Space in place of number	46
7.4.16 Boxing line number and line symbol	46
7.5 Separator between the lemma and the note	47
7.5.1 For footnotes	47
7.5.2 For endnotes	47
7.6 Font style	48

7.6.1 For line number	48
7.6.2 For the lemma	48
7.6.3 For all notes	48
7.7 Wrapping notes	48
7.7.1 Wrapping lemmas	48
7.7.2 Wrapping contents	49
7.8 Indent of notes content	49
7.9 Arbitrary code	49
7.9.1 Arbitrary code around line number	49
7.9.2 Arbitrary code at the beginning of notes	49
7.9.3 Arbitrary code before inserting note	50
7.10 Options for footnotes in columns	50
7.10.1 Alignment	50
7.10.2 Size of the columns	50
7.11 Options for paragraphed footnotes and notes grouped by line	51
7.11.1 Mark separation of notes	51
7.11.2 Ragged text	51
7.12 Options for block of notes	51
7.12.1 Grouping notes by line	51
7.12.2 Text before notes	51
7.12.3 Code before notes	52
7.12.4 Spacing	52
7.12.5 Rule	52
7.12.6 Maximum height	52
7.12.7 Width	53
7.13 Footnotes and the <code>reledpar</code> columns	53
7.14 Endnotes in one paragraph	53
<b>8 Fonts</b>	<b>53</b>
<b>9 Verse</b>	<b>54</b>
9.1 Basic	54
9.2 Define stanza indents	54
9.3 Repeating stanza indents	55
9.4 Notes about the scope of the indent settings	55
9.5 Manual stanza indent	55
9.6 Stanza breaking	56
9.7 Hanging symbol	56
9.8 Long verse and page break	56
9.9 Content before/after verses	56
9.10 Numbering stanza	57
9.11 Antilabe	58
9.12 Various tools	58
9.13 Notes on empty lines	58
<b>10 Grouping</b>	<b>58</b>

<b>11 Cross referencing</b>	<b>59</b>
11.1 Basic use	59
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	60
11.4 Not automatic cross-referencing	61
11.5 Normal $\LaTeX$ cross-referencing	61
11.6 References to start and end lines	61
11.6.1 Reference to main text lines	61
11.6.2 References to lines that are commented on in the apparatus	62
11.6.3 Settings	62
11.6.4 Combining multiple references	64
11.7 Compatibility with <code>xr</code> package	64
<b>12 Sidenotes</b>	<b>64</b>
12.1 Basics	64
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	65
<b>13 Indexing</b>	<b>65</b>
13.1 Basics	65
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	66
13.5 Using <code>xindy</code>	67
13.6 Advanced setting	67
<b>14 Glossary</b>	<b>68</b>
14.1 Preamble setting	68
14.2 Commands	68
<b>15 Tabular material</b>	<b>68</b>
<b>16 Sectioning commands</b>	<b>71</b>
16.1 Sectioning commands without line numbers or critical notes	71
16.2 Sectioning commands with line numbering and critical notes	71
16.3 Optimization	72
<b>17 Quotation environments</b>	<b>72</b>
<b>18 Page breaks</b>	<b>73</b>
18.1 Control page breaking	73
18.2 Prevent page break in a long verses	73

<b>19 Miscellaneous</b>	<b>73</b>
19.1 Known and suspected limitations	74
19.1.1 Non-standard geometry	74
19.1.2 floatrow package compatibility	74
19.1.3 “No room for a new”	74
19.1.4 Marginal notes	74
19.1.5 Paragraph shape	75
19.1.6 Paragraphed footnotes	75
19.1.7 Use with other packages	75
19.2 Parallel typesetting	76
<b>I Implementation overview</b>	<b>77</b>
<b>II Preliminaries</b>	<b>77</b>
II.1 Links with original EDMAC	77
II.2 Package declaration	77
II.3 Package options	78
II.3.1 Options of reledpar	78
II.3.2 Options of reledmac	78
II.4 Loading packages	80
II.5 Compatibility with Lua $\TeX$	80
II.6 Boolean flags	81
II.7 Messages	82
II.8 Gobbling	89
II.9 Wrapping content	89
II.10 Miscellaneous commands	89
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>96</b>
<b>V Line counting</b>	<b>97</b>
V.1 Choosing the system of lineation	97
V.2 Line number margin	99
V.3 Line number initialization and increment	99
V.4 Line number locking	101
V.5 Line number style	102
V.6 Line number printing	102
V.7 Line number counters and lists	104
V.8 Line number locking counter	105
V.9 Line number associated to lemma	105
V.10 Reading the line-list file	108
V.11 Commands within the line-list file	110
V.12 Writing to the line-list file	123

<b>VI Marking text for notes</b>	<b>130</b>
VI.1 <code>\edtext</code> itself	131
VI.2 Substitute lemma	139
VI.3 Substitute line numbers	140
VI.4 Lemma disambiguation	141
<b>VII Paragraph decomposition and reassembly</b>	<b>147</b>
VII.1 Boxes, counters, <code>\pstart</code> and <code>\pend</code>	148
VII.2 Processing one line	154
VII.2.1 General process	154
VII.2.2 Process for “normal” line	155
VII.2.3 Process for line containing <code>\eledsection</code> command	156
VII.2.4 Hooks	157
VII.2.5 Sidenotes and marginal line number initialization	158
<b>VIII Line and page number computation</b>	<b>159</b>
VIII.1 Continuous line numbering between parallel typesetting and normal typesetting	162
<b>IX Line number annotation</b>	<b>163</b>
<b>X Line number printing</b>	<b>165</b>
<b>XI Pstart number printing inside</b>	<b>170</b>
<b>XII Restoring footnotes and penalties</b>	<b>171</b>
XII.1 Add insertions to the vertical list	171
XII.2 Penalties	173
XII.3 Printing leftover notes	174
XII.4 Text before notes	174
<b>XIII Critical footnotes</b>	<b>176</b>
XIII.1 Fonts	176
XIII.2 Individual note options	176
XIII.3 Notes language	177
XIII.4 General survey of the way we manage notes	178
XIII.5 General setup	178
XIII.6 Footnotes arrangement	179
XIII.6.1 User level macro	179
XIII.6.2 Normal footnote	179
XIII.6.3 Paragraphed footnotes	184
XIII.6.4 Columnar footnotes	191
XIII.7 Footnote paragraph indent	197
XIII.8 Footnote hanging indent	198
XIII.9 Critical notes presentation	198
XIII.9.1 Font tools	198
XIII.9.2 Pstart number in footnote	199
XIII.9.3 Lemma printing	200
XIII.9.4 Line number printing	200
XIII.9.5 Footnote grouped by line	213

<b>XIV Familiar footnotes</b>	<b>214</b>
XIV.1 Adjacent footnotes	214
XIV.2 Regular footnotes for numbered texts	215
XIV.3 Footnote formats	217
XIV.4 Footnote arrangement	218
XIV.4.1 User level macro	218
XIV.4.2 Normal footnotes	218
XIV.4.3 Two columns footnotes	224
XIV.4.4 Three columns footnotes	226
XIV.4.5 Paragraphed footnotes	228
XIV.5 Wrapping footnote marks in hyperlink	232
<b>XV Code common to both critical and familiar footnote in normal arrangement</b>	<b>233</b>
<b>XVI Footnotes' width for two columns</b>	<b>233</b>
<b>XVII Footnotes' order</b>	<b>235</b>
<b>XVIII Footnotes' rule</b>	<b>235</b>
<b>XIX Specific skip for first series of footnotes</b>	<b>235</b>
XIX.1 Overview	235
XIX.2 User level command	236
XIX.3 Internal commands	236
<b>XX Endnotes</b>	<b>237</b>
XX.1 Internal commands	238
XX.2 User level commands	241
XX.2.1 Inserting contents to endnotes	241
XX.2.2 Printing endnotes	242
<b>XXI Generate series of notes</b>	<b>251</b>
XXI.1 Test if series is still existing	251
XXI.2 Init specific to reledpar	251
XXI.3 For critical footnotes	251
XXI.3.1 Options	252
XXI.3.2 Create inserts, needed to add notes in foot	253
XXI.3.3 Create commands for critical apparatus, \Afootnote, \Bfootnote etc.	254
XXI.3.4 Set standard display	256
XXI.4 For familiar footnotes	256
XXI.4.1 Options	257
XXI.4.2 Create tools for familiar footnotes (\footnoteX)	257
XXI.5 The endnotes	260
XXI.5.1 The auxiliary file	260
XXI.5.2 The main macro	260
XXI.5.3 Tools	261
XXI.5.4 Internal commands	261
XXI.5.5 The options	261
XXI.6 Init standards series (A,B,C,D,E)	263



<b>XXII Setting series display</b>	<b>263</b>
XXII.1 Change series order	263
XXII.2 Test series order	264
XXII.2.1 Get the first series	264
XXII.3 Series setting	264
XXII.3.1 General way of working	264
XXII.3.2 Tools to set options	265
XXII.3.3 Tools to generate options commands	266
XXII.3.4 Options for critical notes	268
XXII.3.5 Options for familiar notes	270
XXII.3.6 Options for endnotes	270
XXII.4 Hooks for a particular footnote	272
XXII.5 Alias	274
<b>XXIII Output routine</b>	<b>274</b>
XXIII.1 Extra footnotes output	274
XXIII.2 Patching standard output's commands	277
<b>XXIV Page numbering in parallel typesetting</b>	<b>279</b>
<b>XXV Cross referencing</b>	<b>282</b>
XXV.1 Compatibility with xref	297
<b>XXVI Sidenotes</b>	<b>298</b>
<b>XXVII Minipages and such</b>	<b>306</b>
<b>XXVIII Indexing</b>	<b>310</b>
XXVIII.1 Looking on package order	310
XXVIII.2 Auxiliary macros for <code>\edindex</code>	311
XXVIII.3 Code specific to <code>\edindexin</code> critical footnotes	312
XXVIII.4 Analysis of command in indexed text	313
XXVIII.5 Code for the formatted index	313
XXVIII.6 Main code	314
XXVIII.7 Hyperlink	316
XXVIII.8 "innote" and "notenumber" option of <code>indextols</code> package	318
<b>XXIX Glossaries</b>	<b>319</b>
<b>XXX Verse</b>	<b>322</b>
XXX.1 Hanging symbol management	322
XXX.2 Using <code>&amp;</code> character	322
XXX.3 Code category setting	323
XXX.4 Stanza count and indent	323
XXX.5 Numbering stanza	324
XXX.6 Stanza number in note	325
XXX.7 Main work	326
XXX.8 Restore catcode and penalties	329
XXX.9 Dealing with antilabe	329

<b>XXXI Apparatus of Manuscripts</b>	<b>330</b>
XXXI.1 User level macro	330
XXXI.2 Setting macro	331
XXXI.3 Counters and lists	332
XXXI.4 Auxiliary file macros	332
XXXI.5 Action macro	333
XXXI.6 Inserting footnote	338
XXXI.7 Other	338
<b>XXXII Arrays and tables</b>	<b>339</b>
XXXII.1 Preamble: macro as environment	339
XXXII.2 Tabular environments	342
XXXII.2.1 Disabling and restoring commands	342
XXXII.2.2 Counters, boxes and lengths	346
XXXII.2.3 Tabular typesetting	350
XXXII.2.4 Environments	360
<b>XXXIII Quotation's commands</b>	<b>360</b>
<b>XXXIV Section's title commands</b>	<b>361</b>
XXXIV.1 Commands to disable some feature	361
XXXIV.2 General overview	362
XXXIV.3 <code>\beforeeledchapter</code> command	362
XXXIV.4 Auxiliary commands	363
XXXIV.5 Patching standard commands	364
XXXIV.6 Main code of <code>\eledxxx</code> commands	368
XXXIV.7 Macros written in the auxiliary file	371
<b>XXXV Page breaking or no page breaking depending on specific lines</b>	<b>373</b>
<b>XXXVI Long verse: prevents being separated by a page break</b>	<b>374</b>
<b>XXXVII Tools for hyperref package</b>	<b>375</b>
<b>XXXVIII Compatibility with eledmac</b>	<b>376</b>
<b>Appendix A Things to do when changing versions</b>	<b>379</b>
A.1 Migrating from EDMAC to ledmac	379
A.2 Migration from ledmac to eledmac	380
A.3 Migration to eledmac 1.5.1	381
A.4 Migration to eledmac 1.12.0	381
A.5 Migration to eledmac 17.1	381
A.6 Migration to eledmac 1.21.0	382
A.6.1 <code>\Xledsetnormalparstuff</code> and <code>\ledsetnormalparstuffX</code>	382
A.6.2 Endnotes	382
A.7 Migration to eledmac 1.22.0	382
A.8 Migration to eledmac 1.23.0	382
A.9 Migration from eledmac to reledmac	382
A.9.1 Risk of 'no room for a new'	382
A.9.2 Multiple indices with memoir	383
A.9.3 Deprecated commands and options	383

A.9.4 <code>\renewcommand</code> replaced by <code>command</code> . . . . .	383
A.9.5 Commands the names of which have been changed . . . . .	384
A.9.6 Endnotes . . . . .	385
A.9.7 Z Series . . . . .	385
A.9.8 Internal commands . . . . .	385
A.10 Migration to <code>reledmac</code> 2.1.0 . . . . .	385
A.11 Migration to <code>reledmac</code> 2.1.3 . . . . .	386
A.12 Migration to <code>reledmac</code> 2.3.0 . . . . .	386
A.13 Migration to <code>reledmac</code> 2.4.0 . . . . .	386
A.14 Migration to <code>reledmac</code> 2.5.0 . . . . .	386
A.15 Migration to <code>reledmac</code> 2.7.0 . . . . .	386
A.16 Migration to <code>reledmac</code> 2.7.2 . . . . .	387
A.17 Migration to <code>reledmac</code> 2.8.0 . . . . .	387
A.18 Migration to <code>reledmac</code> 2.13.1 . . . . .	387
A.19 Migration to <code>reledmac</code> 2.18.0 . . . . .	387
A.20 Migration to <code>reledmac</code> 2.21.0 . . . . .	387
A.21 Migration to <code>reledmac</code> 2.24.0 . . . . .	387
A.22 Migration to <code>reledmac</code> 2.26.0 . . . . .	388
A.23 Migration to <code>reledmac</code> 2.27.1 . . . . .	388
A.24 Migration to <code>reledmac</code> 2.30.0 . . . . .	388
A.25 Migration to <code>reledmac</code> 2.31.1 . . . . .	388
A.26 Migration to <code>reledmac</code> 2.32.7 . . . . .	388
A.27 Migration to <code>reledmac</code> 2.33.0 . . . . .	388
<b>Appendix B Auxiliary softwares</b> . . . . .	<b>388</b>
B.1 <i>samewords</i> . . . . .	388
B.2 critical-keys for <i>Emacs</i> . . . . .	389
B.3 critical-marks for <i>Emacs</i> . . . . .	389
B.4 Import from TEI . . . . .	389
B.5 Import from TEI . . . . .	389
<b>References</b> . . . . .	<b>390</b>
<b>Index</b> . . . . .	<b>390</b>
<b>Change History</b> . . . . .	<b>440</b>

## 1 Introduction

### 1.1 Aim of the package

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

- automatic stepped line numbering, by page, section or paragraph;
- sub-lineation within the main series of line numbers;
- variant readings automatically keyed to line numbers;
- caters to both prose and verse;

- multiple series of footnotes and endnotes;
- block or columnar formatting of the footnotes;
- simple tabular material may be line numbered;
- indexing keyed to page and line numbers.

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

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

Any suggestions for new features are welcome.

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

You do not need to read the source code for this package in order to use it; we provide this code primarily for reference, and many of our comments on it repeat material that is also found in earlier sections. But no documentation, however thorough, can cover every question that comes up and many can be answered quickly by consulting the code. On a first reading, we suggest that you read only the general documentation in sections 2, unless you are particularly interested in the innards of `reledmac`.

## 1.2 History

### 1.2.1 EDMAC

The original version of EDMAC was `TEXTED.TEX`, written by John Lavagnino in late 1987 and early 1988 for formatting critical editions of English plays.

John passed these macros on to Dominik Wujastyk who, in September–October 1988, added the footnote paragraphing mechanism, margin swapping and other changes to suit his own purposes, making the style more like that traditionally used for classical texts in Latin and Greek (e.g., the Oxford Classical Texts series). He also wrote some extra documentation and sent the files out to several people. This version of the macros was the first to be called EDMAC.

The present version was developed in the summer of 1990, with the intent of adding necessary features, streamlining and documenting the code, and further generalizing it to make it easily adaptable to the needs of editors in different disciplines. John did most of the general reworking and documentation, with the financial assistance of the Division of the Humanities and Social Sciences, California Institute of Technology. Dominik adapted the code to the conventions of Frank Mittelbach’s `doc` option, and added some documentation, multiple-column footnotes, cross-references, and crop marks.<sup>1</sup> A description by John and Dominik of this version of EDMAC was published as “An overview of EDMAC: a PLAIN  $\TeX$  format for critical editions”, *TUGboat* 11 (1990), pp. 623–643.

From 1991 through 1994, the macros continued to evolve, and were tested at a number of sites. We are very grateful to all the members of the (now defunct)

<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).

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 T<sub>E</sub>X and EDMAC. Another project Wayne has worked on is a DVI post-processor which works with an EDMAC that has been slightly modified to output `\specials`. This combination enables you to recover to some extent the text of each line as ASCII code, facilitating the creation of concordances, an *index verborum*, etc.

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

### 1.2.2 ledmac

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

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

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

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

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

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

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

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

<sup>7</sup>Menso Folkerts, “Die *Rithmachie* des Werinher von Tegernsee”, *ibid.*

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

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

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

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

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

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. 40). 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. 380.

### 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. 382.

## 1.3 Bibliography

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

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

## 1.4 Acknowledgements

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

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

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

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

### 3 Compatibility warning

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

Please read [7.12.6 p. 52](#).

If you use the Lua $\TeX$  engine, you need Lua $\TeX$  1.1.0 or later. A file may mix *numbered* and *unnumbered* text.

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

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

### 4 Options

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

#### 4.1 Specific features

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

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

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

**antilabe** enable `\antilabe` feature (see [9.11 p. 58](#));

**eledmac-compat** help to migrate from `eledmac` to `reledmac` (see [A.9.5 p. 384](#));

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

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

**noquotation** by default, the quotation environment is redefined within numbered text. You can disable this redefinition with `noquotation` (see [17 p. 72](#));

**noresetlinenumannotation** does not reset the annotations to line number at each line (see [5.5 p. 23](#));

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

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

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

**xindy** and `xindy+hyperref` select `xindy` as the index processor (13.5 p. 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. 71).

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

Note that a `\noindent` will be automatically added before this argument, and, consequently, a `\parskip` will be inserted. You can use a second optional argument in brackets to avoid that `\noindent`.

```

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

```

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

If you need to start a `\pstart` with brackets, or to add brackets after a `\pend`, just add a `\relax` between `\pstart ... \pend` and the brackets.

This feature is also useful when typesetting verses (see 9 p. 54) or `reledpar` (see 19.2 p. 76).

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 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:

`\AtEndEveryPend`

```

\AtStartEveryPstart{*}

```

---

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

Instead of manually doing

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

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

It is possible to insert a number at every `\pstart` command; you must use the

`\numberpstarttrue` `\numberpstarttrue` command to have it. You can stop the numbering with `\numberpstartfalse`.

`\numberpstartfalse` You can redefine the command `\thepstart` to change style. You can change the value

`\thepstart` of the `pstart` number by using *after* `\beginnumbering`:

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

On each `\beginnumbering` the numbering restarts.

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

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

### 5.2.6 Languages written in Right to Left

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

### 5.2.7 Memory limits

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

`\pausenumbering` `reledmac` stores a lot of information about line numbers and footnotes in memory 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` Line numbering can be disabled with `\numberlinefalse`. It can be enabled again with `\numberlinetrue` `\numberlinetrue`.

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

- It should not be use if you have critical footnotes after `\numberlinefalse`.
- It could be used in the following cases:
  - You want only familiar footnotes, not critical footnotes;
  - You want only parallel typesetting (using `reledpar`) but no critical footnotes.
  - You disable, for a portion of text, line numbering
- It must not be used if:
  - You do not want to have the line number in the margins, but do want to use critical footnotes. In this case, set `\firstlinenum` to a big value, such as 100,000 (5.3.2 p. 20).
  - You want to freeze the line numbering for some line, but still keep critical footnotes. In this case, use `\startlock` (5.4.2 p. 22).

A successor to `reledmac` may disable this feature.

### 5.3.2 Setting lineation start and step

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

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

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

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

in any order but it is easier to read if you put them in numerical order. Either omitting the definition of `\linenumberlist` or following the empty definition

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

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

### 5.3.3 Setting lineation reset

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

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

### 5.3.4 Setting line number margin

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

```
\linenummargin{left}
```

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

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

### 5.3.5 Other settings

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

## 5.4 Changing the line numbers

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

### 5.4.1 Sublineation

`\startsub` You insert the `\startsub` and `\endsub` commands in your text to turn sub-lineation  
`\endsub` on and off. For example, stage directions in plays are often numbered with sub-line

numbers: as line 10.1, 10.2, 10.3, rather than as 11, 12, and 13. Titles and headings are sometimes numbered with sub-line numbers as well.

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

`\Xsublinesep` You can change the separator between the line number and the subline number either by using `\Xsublinesep` without any optional argument (7.4.11 p. 45) or by using `\Xsublinesepside`. In the second case, it will change the separator only for line numbers in the margins, not in the footnotes.

#### 5.4.2 Locking lineation

`\startlock` The `\startlock` command, used in running text, locks the line number at its current value, until you insert `\endlock`. It can tell for itself whether you are in a patch of line or sub-line numbering. One use for line-number locking is in printing poetry: there the line numbers should be those of verse lines rather than of printed lines, even when a verse line requires several printed lines. But in this case you may use the `\stanza` mechanism, see 9 p. 54.

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

#### 5.4.3 Setting and changing line number

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

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

#### 5.4.4 Line number style

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

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

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

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

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

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

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

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

#### 5.4.5 Skipping and hiding number

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

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

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

## 5.5 Adding annotations to line numbers

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

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

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

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

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

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

- An annotation is associated with a line of text. It is reset with each new line of text, unless you use the `noresetlinenumannotation` option of `reledmac`.
- It is printed alongside the line number in the margin.

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

- It is printed following the line number in critical footnotes and endnotes.
- And of particular interest: the annotation is printed in the critical note *only* when the `\edtext` is issued after the `\linenumannotation`.
- If two or more annotations are called before an `\edtext`, the last one is used in associated notes.
- If an annotation is called within an `\edtext`, it is printed as an annotation for the end of the lemma.

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

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

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.

Note that the annotation must be called before the beginning of a `\edtext` to be taken into account as a starting annotation. So:

```
\linenumannotation{A}\edtext{something...\linenumannotation{B}}{\Afootnote{foo}}
```

will use A as starting annotation, while:

```
\edtext{\linenumannotation{A}something...\linenumannotation{B}}{\Afootnote{foo}}
```

won't.

### 5.5.1 Resetting annotation

You can manually reset annotation using `\resetlinenumannotation{}`. Note that this macro takes no argument, but you should use empty brackets or `\in` order to avoid gobbling spaces.<sup>15</sup>



### 5.5.2 Modifying annotation associated with a specific note

`\lineannot` The `\linenum` (6.2.5 p. 30) macro changes the line number passed to a note. The `\lineannot` macro does the same for the annotations. It takes one argument, containing the beginning and ending annotations separated by a pipe (`|`). As with `\linenum`, if one wishes to change some specific part of the annotation, one can use

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

### 5.5.3 Changing the position of the annotation

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

To change this behavior, 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.

With the `\linenumannotationothersidettrue`, the annotation will appear on the margin side opposite to the line number.

You can reset the default behavior with `\linenumannotationothersidefalse`.

### 5.5.4 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. 59).

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

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

---

<sup>15</sup>As for all TeX macro with no argument.

- `\Xwraplinenumannotationref{⟨cmd⟩}` for annotations referenced using the `\annotationref` command of the crossref mechanism (11 p. 59).

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>16</sup>

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

### 5.5.5 Not printing the second identical annotations

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

`\Xnoidenticallinenumannotation`  
`\Xendnoidenticallinenumannotation`

### 5.5.6 Separator between annotations

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

`\setlinenumannotationsep`

### 5.5.7 Annotations in the apparatus

Some other options for annotations in the apparatus are described in 7.4.10 p. 44

## 5.6 Executing code at each line

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

## 5.7 Executing code at a specific line

Sometime, you want to execute a code at a precise point in the flow of your text. For example, if the current header of the page corresponds to the position inside the text, you may want to use `\markboth` (or similar) command. The main problem is that you

<sup>16</sup>The `\@gobble` command takes one argument, and returns nothing.

need to execute `\markboth` when `reledmac` typesets the text, and not when it reads it, because that could create damage with vertical spacing.

`\doinsidethislinehook`

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

For example, with the following code:

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

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

See the file `examples/2-subdivision-number-in-header.tex` for an example. Especially, the use of this command with `ℒTEX` 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:
I saw my friend \edtext{Smith}{
\Afootnote{Jones C, D.}}
on Tuesday.

```

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

```

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

---

1 Smith ] Jones C, D.

---

1-2 I 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  
`\Bfootnote` `\Afootnote{<text>}`. When all of the five are used, the A notes appear in a layer just  
`\Cfootnote` below the main text, followed by the rest in turn, down to the E notes at the bottom.  
`\Dfootnote` These are the main macros that you will use to construct the critical apparatus of your  
`\Efootnote` text.

If you need more series of critical notes, please look at [6.7.1 p. 39](#).

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. [7.4.5 p. 43](#)).
- `nonum` disables line numbering for this note. A horizontal blank space is added instead. You can use `\Xinplaceofnumber` to set it ([7.4.15 p. 46](#)).
- `nosep` to disable the lemma separator for this note. A horizontal blank space is added instead. You can use `\Xinplaceoflemmaseparator` to set it ([7.5.1 p. 47](#)).
- `linerangesep=<c>` to change to `<c>` the separator between start line and end line for this particular note.

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

### 6.2.3 Endnotes

`\Aendnote` **Inserting endnotes** The package also maintains five separate series of endnotes.  
`\Bendnote`  
`\Cendnote`  
`\Dendnote`  
`\Eendnote`

If you do not need the endnotes facility, you should use `noend` option when loading `reledmac`.

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

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

*option* can contain a comma-separated list of values. Allowed values are:

- `fulllines` to disable `\Xendtwolines` and `\Xendmorethantwolines` features for this particular note (cf. 7.4.5 p. 43).
- `nonum` to disable line number for this particular note.
- `nosep` to disable the lemma separator for this particular note. A horizontal blank space is added instead. You can use `\Xendinplaceoflemmaseparator` to set it (7.5.2 p. 47).
- `linrangesep=c` to change to *c* the separator between start line and end line for this particular note.

`\doendnotes` **Printing endnotes** Normally, endnotes are not printed: you must use the `\doendnotes`{*s*}, where *s* is the letter of the series to be printed. Put this command where you want the corresponding set of endnotes printed. In this case, all the endnotes of the *s* series are printed, for all numbered sections.

`\doendnotesbysection` However, you may want to print the endnotes of one given series covering the first numbered section, then the endnotes of another given series covering the first numbered section, then the endnotes of the first given series covering the second numbered section, then the endnotes of the second given series covering the second numbered section, and so forth. In this case, use `\doendnotesbysection`{*s*}. For each value of *s*, the first call of the command will print the notes for the first series, the second call will print the notes for the second series etc. For example, do:

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

Note that by default inside endnotes no separator is used between the lemma and the content. However you can use the `\Xendlemmaseparator` macro to define one (7.5.2 p. 47).

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

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

`\Xtoendnotes` [*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>17</sup> version of the `<content>`, that is the version where the commands are expanded on the first level. This can be, for example, to get a counter value. Use the starred version in this case. For example:

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

#### 6.2.4 Paragraph in critical apparatus

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

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

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

#### 6.2.5 Change lemma and line number

`\lemma` If you want to change the lemma that gets passed to the notes, you can do this by using `\lemma{<alternative>}` within the second argument to `\edtext` and before the note commands. The most common use of this command is to abbreviate the lemma that’s printed in the notes. For example:

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

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

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

However, you can retain the value computed by `reledmac` for any number by simply omitting it; and you can omit a sequence of vertical bars at the end of the argument.

<sup>17</sup>The expansion mechanism of TeX is a quite complex problem, but fundamental. We have no place to explain it fully here. Read introduction to TeX to understand well.

For example, `\linenum{|||23}` changes only the ending page number of the current lemma.

This command does not change the marginal line numbers in any way; it just changes the numbers passed to the notes. Its use comes in situations that `\edtext` has trouble dealing with for whatever reason. If you need notes for overlapping passages that aren't nested, for instance, you can use `\lemma` and `\linenum` to generate such notes despite the limitations of `\edtext`. If the *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. 59) 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>18</sup>

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

## 6.3 Disambiguation of identical words in the apparatus

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

### 6.3.1 Basic use

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

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

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

<sup>18</sup>We use `\newcommand` and `\newcommandx` instead of classical `\let` command because the `edtabular` environments have to modify the notes definition, and we need to use the newest definition of notes. Read the handbook of `xargs` to know more about `\newcommandx`.

### 6.3.2 Case setting

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

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

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

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

For instance, in French, “é” has two possible Unicode codepoints:

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

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

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

- If you use  $X_{\text{L}}\text{TeX}$ , add this line in your preamble: `\XeTeXinputnormalization 1`.
- If you use  $\text{Lua}\text{TeX}$ , use the `uninormalize` package of Michal Hoftich<sup>19</sup> with the `buffer` option set to true.

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

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

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

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

### 6.3.5 Use with `\lemma` command

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

For example, in the following example:

```
some thing
  \edtext{\sameword{sw}
          and other \sameword{sw}
          and again \sameword{sw}}
```

<sup>19</sup><https://github.com/michal-h21/uninormalize>.



```

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

```

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

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

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

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

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

```

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

```

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

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

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

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

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

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

Here, the corresponding code:

```

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

```

### 6.3.6 Sameword for a group of words

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

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

```

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

```

1 per causam tamen scire causam est per causam cognoscere causam

---

1 causam<sup>2</sup> ] fnote  
1 per<sup>2</sup> causam<sup>3</sup> ] causam rei B  
1 causam<sup>4</sup> ] fnote

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

However, in the case that “per causam” is the lemma of the second note, there should be only one number for that whole lemma. In this case we can mark all “per causam” groups. But as “causam” is also called as lemma in note 1 and 3, we need to use nested

`\sameword` commands. Consequently, we need to use `\lemma` for the `\edtext` linked to “per causam”, as we don’t want to number each individual word.

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

```
1 per causam tamen scire causam est per causam cognoscere causam
```

---

```
1 causam2] fnote
```

```
1 per causam2] causam rei B
```

```
1 causam4] fnote
```

### 6.3.7 Customizing

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

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

### 6.3.8 Problems with some macros

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

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

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

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

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

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

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

### 6.3.9 Automatic sameword annotation

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

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

## 6.4 Apparatus of manuscripts

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

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

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

### 6.4.1 Marking sections of text

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

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

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

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

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

```
[...]
amet\stopmsdata{} \msdata{ABCD}sic transit [...]
```

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

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

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

### 6.4.2 Layout of the apparatus of manuscripts

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

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

### 6.4.3 Settings

As the apparatus of manuscripts technically consists of first-level critical notes (‘A’ series), any setting available for critical notes can be applied (7 p. 40). 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 behavior using `\msdataposition{msdata-regular}`

## 6.5 Familiar notes

### 6.5.1 Basic use

`\footnoteA` As well as the standard L<sup>A</sup>T<sub>E</sub>X footnotes generated via `\footnote`, the package also provides five series of additional footnotes called `\footnoteA` through `\footnoteE`. These `\footnoteC` have the familiar marker in the text, and the marked text at the foot of the page can be `\footnoteD` formatted using any of the styles described for the critical footnotes. Note that the “regular” footnotes have the series letter at the end of the macro name whereas the critical footnotes have the series letter at the start of the name.

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

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

### 6.5.2 Customizing mark

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

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

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

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

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

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

There are similar command triples for the other series.

### 6.5.3 Separator for multiple footnotes

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

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

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

and can be changed if necessary.

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

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

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

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

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

## 6.7 Changing series

### 6.7.1 Create a new series

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

### 6.7.2 Delete series

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

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

### 6.7.3 Series order

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

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

## 6.8 Position of critical and familiar footnotes

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

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

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

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

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

`<series><type>`

`<series>` is a series letter (A,B,C etc.), while `<type>` must be either “critical” or “familiar”.

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

```
\fnpos{%
  {A}{familiar},
  {A}{critical},%
  {B}{critical},%
```

```

{C}{critical},%
{D}{critical},%
{E}{critical},%
{B}{familiar},%
{C}{familiar},%
{D}{familiar},%
{E}{familiar}%
}

```

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

## 7 Apparatus customization

### 7.1 Introduction

Some commands can be used to change the way the footnotes are displayed.<sup>20</sup> All can have an optional argument [ $\langle s \rangle$ ], which is the letter of the series – or a list of letters separated by comma – depending on which option is applied. If the optional argument is omitted or empty, the setting will apply to the entire series.

When a length, noted  $\langle l \rangle$ , is used, it can be stretchable: a plus b minus c. The final length  $m$  is calculated by  $\LaTeX$  to have:  $a - c \leq m \leq a + b$ . If you use some relative unit<sup>21</sup>, it will be relative to font size of the footnote, except for commands concerning the place kept by the notes – including blank space.

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

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

There is also name convention:

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

### 7.2 Notes arrangement in a series

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

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

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

<sup>20</sup>In the code of `reledmac`, these commands are called “hooks”.

<sup>21</sup>Like `em` which is the width of an “m” in a given font.



- `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. 75.
- `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.

### 7.3 Disabling footnote

`\xnonote` Some time, you may want to export your edition without notes, but don't want to modify `\nonoteX` globally your code.

`\xnonote[⟨s⟩]` allow to disable critical notes for the series `⟨s⟩`.

`\nonoteX[⟨s⟩]` is the same for familiar notes.

### 7.4 Control number printing

#### 7.4.1 Print line number only at first time

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

`\Xnumberonlyfirstintwolines` Suppose you have a lemma on line 2 and a lemma between line 2 and line 3. With `\Xnumberonlyfirstinline`, the second lemma is considered to be on the same line as the first lemma. But if you use both `\Xnumberonlyfirstinline[⟨s⟩]` and `\Xnumberonlyfirstintwolines[⟨s⟩]`, a distinction is made.

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

`\Xendnumberonlyfirstinline` For endnotes, `\Xendnumberonlyfirstinline`; `\Xendnumberonlyfirstintwolines` and `\Xendsymlinenum` are the equivalents of

`\Xendsymlinenum` `\Xnumberonlyfirstinline`; `\Xnumberonlyfirstintwolines` and `\Xsymlinenum`.

### 7.4.2 Print page number only at first time

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

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

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

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

### 7.4.3 Arbitrary text before line number

`\Xtxtbeforenumber` **For critical notes** `\Xtxtbeforenumber[⟨s⟩]{⟨txt⟩}` allows the insertion of `⟨txt⟩` before the line number only when the line number is printed, so taking into account `\Xnumberonlyfirstinline` and the like. `\Xbeforepagenumber` `\Xbeforepagenumber[⟨s⟩]{⟨txt⟩}` allows the insertion of `⟨txt⟩` before the page number only when the page number is printed, so when a note spans across two or more pages and the lineation is by page.

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

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

defines the text before the line number in endnotes when there is only one line. Default value is empty. `\Xendlineprefixmore[⟨s⟩]{⟨text⟩}` defines the text before the line number in endnotes when there is more than one line. Default value is empty. If you don’t define it, it will use the value defined by `\Xendlineprefixsingle`.

#### 7.4.4 Separator for line range

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

#### 7.4.5 Abbreviate line range

`\Xtwolines` If a lemma is printed on two subsequent lines, `reledmac` will print the first and the last line numbers. Instead of this, it is also possible to print an abbreviation which stands for “line 1 and subsequent line(s)”.  
`\Xmorethantwolines` To achieve this, use `\Xtwolines [⟨s⟩]{⟨text⟩}` and `\Xmorethantwolines [⟨s⟩]{⟨text⟩}`. The `⟨text⟩` argument of `\Xtwolines` will be printed if the lemma is on two lines, and the `⟨text⟩` argument of `\Xmorethantwolines` will be printed if the lemma is on three or more lines. For example:

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

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

If you use `\Xtwolines` without setting `\Xmorethantwolines`, the `⟨text⟩` argument of `\Xtwolines` will be used for lemmas which fall on three or more lines.

However, if you want to use a short form (when the lemma overlaps two lines, but not more than two), use `\Xtwolinesbutnotmore [⟨series⟩]`.

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

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

You can disable `\Xtwolines` and related for a specific note by using the “[fulllines]” argument in the note macro cf. 6.2.2 p. 28.

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

#### 7.4.6 Disabling line number printing

`\Xnonumber` You can use `\Xnonumber [⟨s⟩]` if you do not want to have no line/page/pstart number in the footnotes, nor line number annotation.

`\Xendnonumber` `\Xendnonumber [⟨s⟩]` is the same for endnotes.

#### 7.4.7 Disabling page number printing

`\Xnopagenumberifcurrent` When we use a lineation by page, `reledmac` checks if a lemma spans across two page. In this case, it prints the starting end ending page number.

Many times, one of these two page numbers is also the current page number.

In this case, it is possible to disable page number printing, using `\Xnopagenumberifcurrent [⟨s⟩]`.

### 7.4.8 Printing pstart number

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

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

### 7.4.9 Printing stanza number

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

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

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

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

### 7.4.10 Options for line number annotations

`\Xlinenumannotationonlyfirst` By default, the line number annotation (5.5 p. 23) is printed in every note. If you want to print it only the first time for a given annotation you can use `\Xlinenumannotationonlyfirst[⟨s⟩]`.

Note the two following points:

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

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

`\Xsymlinenumannotation` To use a particular symbol in place of the line number annotation, you can use `\Xsymlinenumannotation[⟨s⟩]{⟨symbol⟩}` in combination with `\Xlinenumannotationonlyfirst[⟨s⟩]`. From the second lemma with the same line number annotation, the symbol will be used

instead of the annotation. Note that any command called in  $\langle symbol \rangle$  must be robust. Use `\robustify` to robustify a non-robust command.

`\endlinenumannotationonlyfirst` For endnotes, `\endlinenumannotationonlyfirst`; `\endlinenumannotationonlyfirstintwo`  
`\endlinenumannotationonlyfirstintwo` and `\endsymmlinenumannotation` are the equivalents of  
`\endsymmlinenumannotation` `\xlinenumannotationonlyfirst`; `\xlinenumannotationonlyfirstintwo` and `\xsymmlinenumannotation`.  
`\xlinenumannotationonlyfirst` The `\xlinenumannotationonlyfirst` disable line number printing only for critical  
`\xlinenumannotationonlyfirstintwo` notes when there are some annotations to the line number.  
`\xsymmlinenumannotation` The `\xendnolinenumannotation` is the same for endnotes.

#### 7.4.11 Separator between line and subline numbers

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

**Employed without optional argument, it also changes the separator in side numbering.**

`\Xendsublinesep` `\Xendsublinesep[\langle s \rangle]{\langle txt \rangle}` 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{\langle txt \rangle}` to do it.**

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

#### 7.4.12 Separator between page and line numbers

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

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

#### 7.4.13 Space around number

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

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

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

#### 7.4.14 Space around line symbol

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

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

`\Xendbeforemynum` `\Xendbeforemynum` and `\Xendaftersymynum` are the equivalents of  
`\Xaftersymynum` `\Xbeforemynum` and `\Xaftersymynum` for the endnotes.

### 7.4.15 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.

### 7.4.16 Boxing line number and line symbol

- `\Xboxlinenum` It could be useful to put the line numbers 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 numbers 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.
```

- `\Xnotboxingsubline` By default, for the start line number, these things are boxed:
- (start) line number annotation, if before line number;
  - (start) page and (start) line number;

- (start) line flag;
- (start) line number annotation, if after line number;
- (start) subline number.

You may exclude subline number and line number annotation after line number, using `\Xnotboxingsubline[⟨s⟩]`. `\Xendboxlinenum[⟨s⟩]{⟨l⟩}`, `\Xendboxlinenumalign[⟨s⟩]{⟨text⟩}`, `\Xendboxlinenumalign[⟨s⟩]{⟨l⟩}`, `\Xendnotboxingsubline[⟨s⟩]`, `\Xendboxendlinenum[⟨s⟩]{⟨l⟩}` are the same as, respectively, `\Xboxlinenum`, `\Xboxlinenumalign`, `\Xboxstartlinenum`, `\Xnotboxingsubline`, `\Xboxendlinenum` except in endnotes.

## 7.5 Separator between the lemma and the note

### 7.5.1 For footnotes

`\Xlemmaseparator` By default, in a footnote, the separator between the lemma and the note is a right bracket (`\rbracket`)<sup>22</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.

### 7.5.2 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.

<sup>22</sup>For `polyglossia`, when the lemma is RTL, the bracket automatically switches to a left bracket.



## 7.6 Font style

### 7.6.1 For line number

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

### 7.6.2 For the lemma

- `\Xlemmadisablefontselection` By default, font of the lemma in footnote is the same as font of the lemma in the main text. For example, if the lemma is in italic in the main text, it is also in italic in note. The `\Xlemmadisablefontselection` [*s*] command allows to disable it for a specific series.
- `\Xendlemmadisablefontselection` By default, font of the lemma in endnote is the same as font of the lemma in the main text. For example, if the lemma is in italic in the main text, it is also in italic in note. The command allows `\Xendlemmadisablefontselection` [*s*] to disable it for a specific series.
- `\Xlemmafnt` Use `\Xlemmafnt` [*s*] {*cmd*} to apply a T<sub>E</sub>X font command to the lemma. For example, to have boldface lemma:
- `\Xlemmafnt` `\Xlemmafnt`{`\bfseries`}

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

### 7.6.3 For all notes

- `\Xnotefontsize` `\Xnotefontsize` [*s*] {*command*} is used to define the font size of critical footnotes of the series. The default value is `\footnotesize`. The *command* must not be a size in pt, but a T<sub>E</sub>X macro, like `\small`.
- `\notefontsizeX` `\notefontsizeX` [*s*] {*command*} is used to define the font size of familiar footnotes of the series. The default value is `\footnotesize`. The *command* must not be a size in pt, but a T<sub>E</sub>X macro, like `\small`.
- `\Xendnotefontsize` `\Xendnotefontsize` [*s*] {*l*} is used to define the font size of end critical footnotes of the series. The default value is `\footnotesize`. The *command* must not be a size in pt, but a T<sub>E</sub>X macro, like `\small`.

## 7.7 Wrapping notes

### 7.7.1 Wrapping lemmas

- `\Xwraplemma` `\Xwraplemma` [*s*] {*cmd*} is used to wrap, in the footnote, the lemma in a T<sub>E</sub>X command. For example, with the  `bidi`  package, to ensure having a lemma written right to left, use `\Xwraplemma`{`\RL`}.
- `\Xwrapendlemma` `\Xwrapendlemma` [*s*] {*cmd*} is the same for endnotes.



### 7.7.2 Wrapping contents

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

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

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

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

## 7.8 Indent of notes content

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

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

`\Xhangindent` For critical notes NOT paragraphed you can define an indent with `\Xhangindent` [*s*] {*l*}, which will be applied in the second line of notes. It can help to make distinction between a new note and a break in a note. The default value is 0 pt.

`\hangindentX` For familiar notes NOT paragraphed you can define an indentation with `\hangindentX` [*s*] {*l*}, which will be applied in the second line of notes. It can help to make a distinction between a new note and a break in a note.

`\Xendhangindent` For critical endnotes NOT paragraphed you can define an indentation with `\Xendhangindent` [*s*] {*l*}, which will be applied in the second line of notes. It can help to make a distinction between a new note and a break in a note.

## 7.9 Arbitrary code

### 7.9.1 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 a space or symbol which replaces line number in endnotes. The code is executed before the `\Xendbeforemysynlinenum` space and before the `\Xendnotenumfont` font setting.

`\Xendahookinplaceofnumber` `\Xendahookinplaceofnumber` [*s*] {*code*} is used to execute code after a space or symbol which replaces line number in endnotes. The code is executed after the `\Xendaftersynlinenum` space.

### 7.9.2 Arbitrary code at the beginning of notes

The three next commands add arbitrary code at the beginning of notes. As the name's space is local to the notes, you can use it to redefine some style inside the notes. For example, if you don't want the `pstart` number to be in bold, use:

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

`\Xbhooknote` `\Xbhooknote[⟨s⟩]{⟨code⟩}` is to be used at the beginning of each critical footnote.  
`\bhooknoteX` `\bhooknoteX[⟨s⟩]{⟨code⟩}` is to be used at the beginning of each familiar footnote.  
`\Xendbhooknote` `\Xendbhooknote[⟨s⟩]{⟨code⟩}` is to be used at the beginning of each endnote.

### 7.9.3 Arbitrary code before inserting note

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

They allow one to add any arbitrary code just before the footnotes are added in the list of footnotes. The main use is to insert text direction code. For example, if you edit right-to-left text with `bidi`, but want your critical footnote be left-to-right, use `\Xbeforeinserting\LTR`. You should also use `\Xwraplemma` to ensure your lemmas are right-to-left in a left-to-right paragraph (7.7.1 p. 48).

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

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

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

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

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

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

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

### 7.10.2 Size of the columns

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

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

## 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` (7.4.1 p. 41).

`\Xgroupbylineseparatetwolines` Note that the `\Xafternote` and `\Xparafootsep` settings are used to determine space and content between footnotes (7.11 p. 51). 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 `\Xgroupbylineseparatetwolines[⟨s⟩]`.

In many cases, you might like to use it in combination with `\Xnumberonlyfirstintwolines` (7.4.1 p. 41).

### 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

`\Xhookgroup` While `\Xtxtbeforenotes` is for typesetting text before notes, `\Xhookgroup[⟨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 must be fixed (no stretch), 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. 50.

## 7.13 Footnotes and the `reledpar` columns

`\noteswidthliketwocolumns` 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 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. 9.9 p. 56)

### 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. 9.7 p. 56.

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. 381](#).**

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

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

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

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

### 9.4 Notes about the scope of the indent settings

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

### 9.5 Manual stanza indent

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

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

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



## 9.6 Stanza breaking

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

The command

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

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

The first entry “1” is a control value. If it is zero, then no penalties are passed on to  $\TeX$ , which is the default. Values between 0 and 10000 are penalty values; values between 10001 and 20000 have 10000 subtracted and the result is given as a negative penalty. The mechanism used for indentations and penalties requires unsigned values less than 32768. No penalty is placed after the last line, so the final ,0 in the example above could be omitted. A penalty of 10000 will prevent a page break; such a penalty is included automatically where there is stanza hanging indentation. A penalty of  $-10000$  (corresponding to the entry value 20000 in this context) forces a page break. Values in between act as suggestions as to the desirability of a page break at a given line. There is a subtle interaction between penalties and *glue*, so it may take some adjustment of skips and penalties to achieve the best results.

## 9.7 Hanging symbol

It is possible to insert a symbol in each line of hanging verse, as in French typography; for example, the opening bracket “[”. To insert it in `reledmac`, use macro `\sethangingsymbol`

`\sethangingsymbol{⟨h⟩}` with this code. In the example of French typography, do

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

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

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

## 9.8 Long verse and page break

If you want to prevent page breaks inside long verses, use the option `nopbinverse` when loading package, or use `\lednopbinversetrue`. Read [18.2 p. 73](#) for further details.

## 9.9 Content before/after verses

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

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

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



- `\AtEveryStanza` • Use `\AtEveryStanza{<arg>}` to automatically add content before stanzas (not in the same paragraph).  
Note that a `\noindent` will be inserted before the argument, and, consequently, a `\parskip`. You can use the starred version of `\AtEveryStanza` to avoid this `\noindent`.
- `\AtStartEveryStanza` • Use `\AtStartEveryStanza` to automatically add content at the beginning of stanzas (in the same paragraph).  
• `&` can be replaced by `\newverse` with two optional arguments (in brackets). The first will be printed after the current verse, the second before the next verse.  
A `\noindent` is automatically inserted before the contents of these optional arguments.  
Use a third and fourth optional argument to not add these `\noindents` (to add content respectively after the current verse / before the next verse).  
• Use `\AtEveryPend{<arg>}` to automatically add content after verses (including the final one) and `\AtEveryPstart{<arg>}` to automatically add content before verses (including the first one).  
• `\&` can take an optional argument (in brackets). Its content will be printed after the stanza.
- `\AtEveryStopStanza` • Use `\AtEveryStopStanza` to automatically add content after the end of stanzas (not in the same paragraph).  
Note that a `\noindent` will be inserted before the argument, and, consequently, a `\parskip`. You can use the starred version of `\AtEveryStopStanza` to avoid this `\noindent`.
- `\BeforeEveryStopStanza` • Use `\BeforeEveryStopStanza` to automatically add content at the end of stanzas (in the same paragraph).

## 9.10 Numbering stanza

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

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

You can use this feature in combination with `\Xstanza` (7.4.9 p. 44).

`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  $\LaTeX$ .

`\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}%
}
```

`\antilabe`

### 9.11 Antilabe

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

`Reledmac` provides tools to cope with this problem, but you must enable them with the `antilabe` option when loading package, as these tools are quite resource consuming.

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

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

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

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

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

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

### 9.12 Various tools

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

`\flagstanza` Putting `\flagstanza[<len>]{<text>}` at the start of a line in a stanza (or elsewhere) will typeset `<text>` at a distance `<len>` before the line. The default `<len>` is `\stanzaindentbase`.

### 9.13 Notes on empty lines

Since v2.3.0 of `reledmac`, empty lines when typesetting verses no longer produce new paragraphs, and consequently, do not insert vertical spaces. Use optional argument of `\stanza` or `\newverse` to insert vertical space (9.9 p. 56).

## 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.

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

`\begin{ledgroupsized}[\langle pos \rangle]{\langle width \rangle}`.

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

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

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

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

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

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

## 11 Cross referencing

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

### 11.1 Basic use

`\edlabel` First you place a label in the text using the command `\edlabel{\langle lab \rangle}`. `\langle lab \rangle` can be almost anything you like, including letters, numbers, punctuation, or a combination—anything but spaces; you might type `\edlabel{toves-3}`, for example.<sup>23</sup>

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

`\edlineref` its location with `\edpageref{\langle lab \rangle}`, `\edlineref{\langle lab \rangle}`, `\edsublineref{\langle lab \rangle}`,

`\sublineref` `\pstartref{\langle lab \rangle}` or `\annotationref{\langle lab \rangle}`, that will produce, respectively, the

`\pstartref` page, line, sub-line, `pstart`, the annotation, on which the `\edlabel{\langle lab \rangle}` command

`\annotationref` occurred.

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

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

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

<sup>23</sup>More precisely, you should stick to characters in the `TeX` categories of “letter” and “other”.

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 subtitle 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`  
`\xlineref`  
`\xsublineref`  
`\xpstartref`  
`\xannotationref`

However, there are situations in which you will want `reledmac` to return a number without displaying any warning messages about undefined labels or the like: if you want to use the reference in a context where  $\LaTeX$  is looking for a number, such a warning will lead to a complaint that the number is missing. This is the case for references used within the argument to `\linenum`, for example (see 6.2.5 p. 30).

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 on 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. 30) and `\lineannot` (5.5.2 p. 25) and sets the beginning page, line, subtitle 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}\Afootnot
\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 L<sup>A</sup>T<sub>E</sub>X cross-referencing

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

`\pageref`

## 11.6 References to start and end lines

### 11.6.1 Reference to main text lines

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

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

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

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

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

`\SErefwithpage` `\SErefwithpage` will make a cross-reference printed as a reference in critical end-notes.

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

### 11.6.2 References to lines that are commented on in the apparatus

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

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

If you use `\linenum` (6.2.5 p. 30) to refer to these labels, `reledmac` will use your line settings to refer to the passage.

`\appref` You can also use `\appref{<label>}` and `\apprefwithpage{<label>}` to refer to these lines. The first one will print the lines as they are printed in the critical footnotes, while the second will print the lines as they are printed in endnotes.

### 11.6.3 Settings

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

For example, you may use:

```
\setapprefprefixsingle{line~}
\setapprefprefixmore{lines~}
```

Note that if you do not use `\setapprefprefixmore`, the argument of `\setapprefprefixsingle` will be used in any case.

`\setSErefprefixsingle` `\setSErefprefixsingle` and `\setSErefprefixmore` are similar for `\SEref` command.  
`\setSErefonlypageprefixsingle` Use `\setSErefonlypageprefixsingle{<prefix>}` to set the page prefix for `\SErefonlypage` when there is only one page. Use `\setSErefonlypageprefixmore{<prefix>}` to set it when there is more than one page. For example:

```
\setSErefonlypageprefixsingle{p.~}
\setSErefonlypageprefixmore{pp.~}
```

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

**Also note that `\setSErefonlypageprefixsingle` is only a shortcut for `\Xendbeforepagenumber` (see 11.6.3 p. 62). So if you use `\Xendbeforepagenumber` without any optional argument, it will override this setting.**

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

These commands are the following:

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

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

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

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

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

**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. 28).

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. 29).

### 11.6.4 Combining multiple references

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

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

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

## 11.7 Compatibility with xr package

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

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

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

## 12 Sidenotes

### 12.1 Basics

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

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

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

`\ledrightnote` `\ledsidenote` 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  $\TeX$  run. If the note positions change in your `.tex` file, you need two runs to get the correction position in the output file. You are strongly encouraged to use tools like `latexmk`, to be sure to get the correct number of runs.**



## 12.2 Setting

### 12.2.1 Width

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

### 12.2.2 Vertical position

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

### 12.2.3 Distance to the main text

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

### 12.2.4 Font

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

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

These can of course be changed to suit.

### 12.2.5 Separator between notes

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

## 13 Indexing

### 13.1 Basics

`\edindex`  $\TeX$  provides the `\index{<item>}` command for specifying that `<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 & linenum  
 should be added to the raw index file.

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

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

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

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

The minimal version of `imakeidx` package to be used is the version 1.3a uploaded on CTAN on 2013/07/11.

Be careful with the order of package loading and index declaration. You must use this order:

1. Load `imakeidx` or `indextools`.
2. Load `reledmac`.
3. Declare the index with the macro `\makeindex` of `imakeidx` and `indextools`.

### 13.3 Referring to critical notes

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

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

If you add `\edindex` inside some `\Xfootnote` command, it will refer to that note, and a suffix *n* will be appended to the reference.

`\Xinnotemark`  
`\innotemarkX`

You can redefine the way the reference is made using:

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

In the `<definition>` argument, you can use the following codes:

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

For example:

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

### 13.4 Separator between page and line numbers

`\pagelinesep`

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

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

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

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

However, you also have to configure your `.ist` index style file. For example, if you use `:` as separator<sup>24</sup>.

```
page_compositor ":"
```

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

## 13.5 Using xindy

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

```
pagenumber-linenummer
```

An example of such a file is provided in the “`examples`” folder. Read the `xindy` handbook to learn how to use it.<sup>25</sup>

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

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

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

1. Use `xindy+hyperref` option when loading the `reledmac` package. When you run (Xe/Lua)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>26</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.

<sup>24</sup>For further detail, you can read <http://tex.stackexchange.com/a/32783/7712>.

<sup>25</sup>Or, for people who read French, read <http://geekographie.maieul.net/174>.

<sup>26</sup>These are the recommended lines to provide the best possible compatibility between `hyperref` and `xindy`, even without using `reledmac`.

## 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

When using `reledmac` with `glossaries`, you must use also `glossaries-extra`.

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. 66).

### 14.2 Commands

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

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

## 15 Tabular material

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

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

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

1	2	3
a	bb	ccc
AAA	BB	C

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

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

For example:

```
\beginnumbering
```



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

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}
```

1	2	3	4
a	⏟		d
A	B	C	D

`\edatleft` `\edatleft[ $\langle math \rangle$ ]{ $\langle symbol \rangle$ }{ $\langle halfheight \rangle$ }` typesets the math  $\langle symbol \rangle$  as  $\left\langle symbol \right\rangle$  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  $\right\langle symbol \right\rangle$  with  $\langle math \rangle$  centered after it.

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

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

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

For example:

```
\begin{edarrayl}
```

```

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

```

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

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

```

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

```

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

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

## 16 Sectioning commands

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

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

```

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

```

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

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

You have to use the following commands:

- `\eledchapter[⟨text⟩]{⟨critical text⟩}`,

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

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

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

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

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

### 16.3 Optimization

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

## 17 Quotation environments

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

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



## 18 Page breaks

### 18.1 Control page breaking

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

`\ledpb`  
`\lednopb`

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

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

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

`\ledpbsetting`

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

### 18.2 Prevent page break in a long verses

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

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

## 19 Miscellaneous

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

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

`\showlemma` The lemma within the text is printed via `\showlemma{lemma}`. Normally, or with the `final` option, the definition of `\showlemma` is:  
`\newcommand*{\showlemma}[1]{#1}`  
 so it just produces its argument. With the `draft` option it is defined as  
`\newcommand*{\showlemma}[1]{\textbf{#1}}`  
 so that its argument is bold, 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. 52 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. 72 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 7.2 p. 41, and described in more detail on XIII.6.3 p. 187, 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. 52), 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>27</sup> you will find L<sup>A</sup>T<sub>E</sub>X whinging “Missing { inserted”, and then things start to fall apart. The trick in this case is to specify either:

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

or

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

(\@secondoftwo is an internal L<sup>A</sup>T<sub>E</sub>X macro that takes two arguments and throws away the first one.) The first incantation lets color show in both the main text and footnotes whereas the second one shows color in the main text but kills it in the lemma and footnotes. On the other hand if you use \textcolor instead, like

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

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

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

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

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

## 19.2 Parallel typesetting

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

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

---

<sup>27</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 targeted for `LaTeX2e`.

```

1 %<*code>
2 \NeedsTeXFormat{LaTeX2e}
3 \ProvidesPackage{reledmac}[2023/08/21 v2.39.7 typesetting critical editions
4 ]%
5 %
```

### II.3 Package options

```

\ifparledgroup Use this to remember which option is used, set and execute the options with final as the
\ifsameparallelpagelnumber default. We use xkeyval in order to manage options with argument.
\ifprevpgnotnumbered \RequirePackage{xkeyval}
\ifledfinal %
\ifnocritical@
\if@noeled@sec
\ifnoend@
\ifnofamiliar@
\ifnoledgroup@
\ifparapparatus@
\ifnoquotation@ \newif\ifparledgroup
\iflednopbinverse \newif\ifsameparallelpagelnumber%
\ifwidthliketwocolumns \newif\ifprevpgnotnumbered%%
\ifxindy@ %
\ifxindyhyperref@
\ifeledmaccompat@
\ifnoresetlinenumnotation@
\ifantilabe@
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
34 \newif\ifnoquotation@
35 \DeclareOptionX{noquotation}{\noquotation@true}
36
37 \newif\ifledfinal
38 \DeclareOptionX{final}{\ledfinaltrue}
39 \DeclareOptionX{draft}{\ledfinalfalse}
40 \ExecuteOptionsX{final}
41
42 \newif\ifparapparatus@
43 \DeclareOptionX{parapparatus}{\parapparatus@true}
44

```

#### II.3.1 Options of reledpar

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

#### II.3.2 Options of reledmac

```

45 \newif\iflednopbinverse
46 \DeclareOptionX{nopbinverse}{\lednopbinversetrue}
47
48 \newif\ifwidthliketwocolumns%
49 \DeclareOptionX{widthliketwocolumns}{\widthliketwocolumnstrue}%
50
51 \newif\ifcontinuousnumberingwithcolumns
52 \DeclareOptionX{continuousnumberingwithcolumns}{\
  continuousnumberingwithcolumnstrue}%
53
54 \newif\ifxindy@
55 \DeclareOptionX{xindy}[eledmac-markup-attr.xdy]{%
56   \AtBeginDocument{\immediate\openout\eledmac@xindy@out=#1}%
57   \newwrite\eledmac@xindy@out%
58   \xindy@true%
59   \gdef\eledmacmarkuplocdepth{:depth 1}%
60   \AtEndDocument{\immediate\closeout\eledmac@xindy@out}%
61 }%
62
63 \newif\ifxindyhyperref@
64 \DeclareOptionX{xindy+hyperref}{%
65   \xindyhyperref@true%
66 }%
67
68 \newif\ifeledmaccompat@%
69 \DeclareOptionX{eledmac-compat}{%
70   \eledmaccompat@true%
71 }%
72 \DeclareOptionX{nopenalties}{%
73   \AtBeginDocument{\let\add@penalties\relax}%
74 }
75 \def\l@auxdir{}%
76 \DeclareOptionX{auxdir}{%
77   \xdef\l@auxdir{#1}/}%
78 }%
79
80 \newif\ifsw@caseinsensitive%
81 \DeclareOptionX{swcaseinsensitive}{%
82   \sw@caseinsensitivetrue%
83 }%
84 \newif\ifnoresetlinenumannotation@
85 \DeclareOptionX{noresetlinenumannotation}{%
86   \noresetlinenumannotation@true%
87 }%
88 \newif\ifantilabe@%
89 \DeclareOptionX{antilabe}{%
90   \antilabe@true%
91 }%
92 %

```

We use the starred form of `\ProcessOptionsX` which executes options in the order listed in the source file: class options, then listed package options, so a package option can override a class option with the same name. This was suggested by Dan Luecking in the `ctt` thread *Class/package option processing*, on 27 February 2004.

```

93 \ProcessOptionsX*\relax
94
95 %

```

## II.4 Loading packages

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

```

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

```

## II.5 Compatibility with LuaTeX

Here, we enable some primitives for LuaTeX.

```

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

```



## II.6 Boolean flags

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

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

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

```
127 \newif\if@ledgroup%
128 %
```

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

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

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

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

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

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

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

`\if@RTL` The `\if@RTL` is defined by the bidi package, which is sometimes loaded by *polyglossia*. But we define it as well if the bidi package is not loaded.

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

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

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

## II.7 Messages

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

`\reledmac@warning` Write a warning message.

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

`\reledmac@error` Write an error message.

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

`\led@err@NumberingStarted` `\newcommand*{\led@err@NumberingStarted}{%`

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

`\led@err@NumberingShouldHaveStarted` `\newcommand*{\led@err@NumberingNotStarted}{%`

```
154 \reledmac@error{Numbering was not started}{\@ehc}}
```

```
155 \newcommand*{\led@err@NumberingShouldHaveStarted}{%

```

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

```
157 %
```

`\led@err@samewordRTL` `\newcommand*{\led@err@samewordRTL}{%`

```
159 \reledmac@error{You can't use \string\sameword \MessageBreak with XeLaTeX
\MessageBreak when typesetting RTL text.\MessageBreak Please use LuaTeX
instead.}{\@ehc}%
```

```
160 }%
```

```
161 %
```

`\led@err@edtextoutsidestart` `\newcommand*{\led@err@edtextoutsidestart}{%`

`\led@err@PstartInEdtext` `\reledmac@error{\string\edtext\space outside numbered paragraph (\pstart\space ... \space\pend)}{\@ehc}}%`

`\led@err@PendInEdtext` `\reledmac@error{\string\pend\space in \string\edtext\space #1 argument}`

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

```
165 \reledmac@error{\string\pstart\space in \string\edtext\space #1 argument}
166 }{\@ehc}%
```

```
167 }%
```

```
168 %
```

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

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

```
171 }%
```

```
172 %
```

`\led@mess@NotesChanged` `\newcommand*{\led@mess@NotesChanged}{%`

```
174 \typeout{reledmac reminder: }%
```

```
175 \typeout{ The number of the footnotes in this section
176 has changed since the last run.}%
```

```
177 \typeout{ You will need to run LaTeX two more times
```

```

178         before the footnote placement}%
179     \typeout{ and line numbering in this section are
180         correct.}}
181 %

```

```

\led@mess@SectionContinued82 \newcommand*\led@mess@SectionContinued}[1]{%
183     \message{Section #1 (continuing the previous section)}}
184 %

```

```

\led@err@LineationInNumbered85 \newcommand*\led@err@LineationInNumbered}{%
186     \reledmac@error{You can't use \string\lineation\space within
187         a numbered section}{\@ehc}}
188 %

```

```

\led@warn@BadLineation89 \newcommand*\led@warn@BadLineation}{%
\led@warn@BadLinenummargin90 \reledmac@warning{Bad \string\lineation\space argument}}
\led@warn@BadLockdisp91 \newcommand*\led@warn@BadLinenummargin}{%
\led@warn@BadSublockdisp92 \reledmac@warning{Bad \string\linenummargin\space argument}}
193 \newcommand*\led@warn@BadLockdisp}{%
194     \reledmac@warning{Bad \string\lockdisp\space argument}}
195 \newcommand*\led@warn@BadSublockdisp}{%
196     \reledmac@warning{Bad \string\sublockdisp\space argument}}
197 %

```

```

\led@warn@NoFile98 \newcommand*\led@warn@NoFile}[1]{%
199     \reledmac@warning{File `#1' not found}}
200 %

```

```

\led@warn@LineFileObsolete01 \newcommand*\led@warn@Obsolete}[1]{%
202     \reledmac@warning{Line-list file #1 was obsolete. We have not read it.
Please run LaTeX again.}}
203 %

```

```

\led@warn@BadAdvancelineSubline04 \newcommand*\led@warn@BadAdvancelineSubline}{%
\led@warn@BadAdvancelineLine05 \reledmac@warning{\string\advanceline\space produced a sub-line
206         number less than zero.}}
207 \newcommand*\led@warn@BadAdvancelineLine}{%
208     \reledmac@warning{\string\advanceline\space produced a line
209         number less than zero.}}
210 %

```

```

\led@warn@BadSetline11 \newcommand*\led@warn@BadSetline}{%
\led@warn@BadSetlinenum12 \reledmac@warning{Bad \string\setline\space argument}}
213 \newcommand*\led@warn@BadSetlinenum}{%
214     \reledmac@warning{Bad \string\setlinenum\space argument}}
215 %

```

```

\led@err@PstartNotNumbered16 \newcommand*{\led@err@PstartNotNumbered}{%
\led@err@PstartInPstart17 \reledmac@error{\string\pstart\space must be used within a
\led@err@PendNotNumbered18 numbered section %
\led@err@PendNoPstart19 (\string\beginnumbering\space ... \space\string\
\led@err@AutoparNotNumbered endnumbering)}{\@ehc}}%
\led@err@NumberingWithoutPstart20 \newcommand*{\led@err@PstartInPstart}{%
221 \reledmac@error{\string\pstart\space encountered while another
222 \string\pstart\space was in effect}{\@ehc}}
223 \newcommand*{\led@err@PendNotNumbered}{%
224 \reledmac@error{\string\pend\space must be used within a
225 numbered section}{\@ehc}}
226 \newcommand*{\led@err@PendNoPstart}{%
227 \reledmac@error{\string\pend\space must follow a \string\pstart}{\@ehc}}
228 \newcommand*{\led@err@AutoparNotNumbered}{%
229 \reledmac@error{\string\autopar\space must be used within a
230 numbered section}{\@ehc}}
231 \newcommand*{\led@err@NumberingWithoutPstart}{%
232 \reledmac@error{\string\beginnumbering... \string\endnumbering\space
without \string\pstart}{\@ehc}}%
233 %

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

\led@warn@DuplicateLabel37 \newcommand*{\led@warn@DuplicateLabel}[1]{%
ed@warn@AppLabelOutSecondArgEdtext38 \reledmac@warning{Duplicate definition of label `#1'\@gobble}%
\led@warn@RefUndefined39 \@latex@warning@no@line{Label `#1' multiply defined}%
\led@warn@RefUndefined40 }%
241 \newcommand*{\led@warn@AppLabelOutSecondArgEdtext}[1]{%
242 \reledmac@warning{\string\applabel\space outside of the second argument
of an \string\edtext\space `#1' on page \thepage.}}%
243 \newcommand*{\led@warn@RefUndefined}[1]{%
244 \G@refundefinedtrue%
245 \reledmac@warning{Reference `#1' on page \thepage\space undefined.%
246 Using `000'.}%
247 \@latex@warning{Reference `#1' undefined\on@line}%
248 }%
249 \newcommand*{\led@warn@pairRefUndefined}[1]{%
250 \G@refundefinedtrue%
251 \reledmac@warning{Reference `#1:start' and/or `#1:end' on page \thepage\
space undefined.
252 Using `??'.}%
253 \@latex@warning{Reference `#1:start' and/or `#1:end' undefined\on@line}%
254 }
255 %

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

```

```

\led@warn@BadHookSeries259 \newcommand{\led@warn@BadHookSeries}[2]{%
260   \reledmac@warning{Bad series #1 for #2 hook. Skipped.}%
261 }%
262 %

\led@warn@BadSidenotemargin263 \newcommand*\led@warn@BadSidenotemargin{%
264   \reledmac@warning{Bad \string\sidenotemmargin\space argument}}
265 %

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

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

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

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

```

```

295 \fi%
296 }%
297 %

\led@err@TooManyColumns98 \newcommand*\led@err@TooManyColumns}{%
\led@err@UnequalColumns99 \reledmac@error{Too many columns}{\@ehc}}
\led@err@LowStartColumn00 \newcommand*\led@err@UnequalColumns}{%
\led@err@HighEndColumn01 \reledmac@error{Number of columns is not equal to the number
\led@err@ReverseColumns302 in the previous row (or \protect\\ \space forgotten?)}{\@ehc}}
303 \newcommand*\led@err@LowStartColumn}{%
304 \reledmac@error{Start column is too low}{\@ehc}}
305 \newcommand*\led@err@HighEndColumn}{%
306 \reledmac@error{End column is too high}{\@ehc}}
307 \newcommand*\led@err@ReverseColumns}{%
308 \reledmac@error{Start column is greater than end column}{\@ehc}}
309 %

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

\led@err@EdtextWithoutNote14 \newcommand*\led@err@EdtextWithoutNote}[1]{%
315 \reledmac@error{edtext ``\unexpanded{#1}` without Xfootnote or Xendnote.
Check syntaxis.}{\@ehc}%
316 }%
317 %

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

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

\led@error@require@glossariesextra26 \newcommand*\led@error@require@glossariesextra}{%
327 \reledmac@error{When using both reledmac and glossaries, you must also
use glossaries-extra}{\@ehc}%
328 }%
329 %

```

```

error@fail@patch@makecol330 \newcommand{\led@error@fail@patch@makecol}{%
331 \reledmac@error{Fail to patch \string\makecol\space command}{\@ehc}%
332 }%
333 %

```

```

r@fail@patch@reinserts334 \newcommand{\led@error@fail@patch@reinserts}{%
335 \reledmac@error{Fail to patch \string\reinserts\space command}{\@ehc}%
336 }%
337 %

```

```

ofail@patch@docclearpage338 \newcommand{\led@error@fail@patch@docclearpage}{%
339 \reledmac@error{Fail to patch \string\docclearpage\space command}{\@ehc}%
340 }%
341 %

```

```

ofail@patch@iiiminipage342 \newcommand{\led@error@fail@patch@iiiminipage}{%
343 \reledmac@error{Fail to patch \string@iiiminipage\space command}{\@ehc}%
344 }%
345 %

```

```

ofail@patch@endminipage346 \newcommand{\led@error@fail@patch@endminipage}{%
347 \reledmac@error{Failed to patch the \string\endminipage\space command}{\
348 @ehc}%
349 %

```

```

ofail@patch@endminipage350 \newcommand{\led@error@fail@patch@makeindex}{%
351 \reledmac@error{Failed to patch the \string\makeindex\space command}{\
352 @ehc}%
353 %

```

```

r@fail@patch@setcounter354 \newcommand{\led@error@fail@patch@setcounter}{%
355 \reledmac@error{Failed to patch the \string\setcounter\space command}{\
356 @ehc}%
357 %

```

```

error@fail@patch@stpelt358 \newcommand{\led@error@fail@patch@stpelt}{%
359 \reledmac@error{Failed to patch the \string\stpelt\space command}{\@ehc}
360 }%
361 %

```

```

\led@error@fail@patch@addtocounter 362 \newcommand{\led@error@fail@patch@addtocounter}{%
363   \reledmac@error{Failed to patch the \string\addtocounter\space command}{\
@ehc}%
364 }%
365 %

366 \newcommand{\led@error@fail@patch@stepcounter}{%
367   \reledmac@error{Failed to patch the \string\stepcounter\space command}{\
@ehc}%
368 }%
369 %

\led@error@fail@patch@mempnum 70 \newcommand{\led@error@fail@patch@mempnum}{%
371   \reledmac@error{Fail to patch \string@mempnum\space command}{\@ehc}%
372 }%
373 %

\led@warn@edinde@outsidenumbering 74 \newcommand{\led@warn@edinde@outsidenumbering}{%
375   \reledmac@warning{\string\edindex\space called outside of \string\
beginnumbering\space ... \space \string\endnumbering. \MessageBreak
Automatically switched to \string\index.}%
376 }%
377 %

\led@warning@hsizeX@deprecated 78 \newcommand{\led@warning@hsizeX@deprecated}{%
379   \reledmac@warning{\string\hsizeX\space command deprecated, use \string\
widthX\space instead.}%
380 }%
381 %

\led@warning@Xhsize@deprecated 82 \newcommand{\led@warning@Xhsize@deprecated}{%
383   \reledmac@warning{\string\Xhsize\space command deprecated, use \string\
Xwidth\space instead.}%
384 }%
385 %

\led@warning@msdatawithoutstop 86 \newcommand{\led@warning@msdatawithoutstop}{%
387   \reledmac@warning{\string\msdata\space without corresponding \string\
stopmsdata}%
388 }%
389 %

\led@warning@preXnotes@deprecated 90 \newcommand{\led@warning@preXnotes@deprecated}{%
391   \reledmac@warning@preXnotes@deprecated%
392 }%
393 %

```



## II.8 Gobbling

Here, we define some commands which gobble their arguments.

```
\@gobblethree94 \providecommand*\@gobblethree}[3]{}
\@gobblefour95 \providecommand*\@gobblefour}[4]{}
\@gobbleseven96 \providecommand*\@gobbleseven}[7]{}
397 %
```

## 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>28</sup> The `\l@wrapcs@ifnotemptybox` does the same, but take as argument a control sequence name, so without backslash.

```
398 \newcommand{\l@wrap@ifnotemptybox}[2]{%
399   \setbox0=\hbox{#2}%
400   \ifdim\wd0=\z@\else%
401     #1{#2}%
402   \fi%
403 }%
404 \newcommand{\l@wrapcs@ifnotemptybox}[2]{%
405   \l@wrap@ifnotemptybox{\csname #1\endcsname}{#2}%
406 }%
407 %
```

## II.10 Miscellaneous commands

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

```
408 \ifldefinal
409   \global\notbool{parapparatus@}\newcommand*{\newcommand}\showlemma
410   }[1]{#1}%
411 \else
412   \global\notbool{parapparatus@}\newcommand*{\newcommand}\showlemma
413   }[1]{\textbf{#1}}%
414 \fi
%
```

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

Initialize it as `\empty`.

```
415 \let\linenumberlist=\empty
416
417 %
```

<sup>28</sup>Compare the distance between letters in the first case (`ab`) and in the second one (`ab`).

`\@l@tempcnta` In imitation of  $\LaTeX$ , we create a couple of scratch counters.  
`\@l@tempcntb`  $\LaTeX$  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).

```
418 \newcount\@l@tempcnta \newcount\@l@tempcntb
419 %
```

## II.11 Prepare `reledpar`

`\ifl@dpairing` In preparation for the `reledpar` package, these are related to the “right” text of parallel  
`\ifl@dpaging` texts (when `\ifl@dpairing` is TRUE). They are explained in the `reledpar` manual.  
`\ifl@dprintingpages`

```
\ifl@dprintingcolumns \newif\ifl@dpairing
\ifpst@rtedL          \newif\ifl@dpaging%
\l@dnumpstartsL      \newif\ifl@dprintingpages%
420 \newif\ifl@dprintingcolumns%
421 \newif\ifpst@rtedL
422 \newif\ifl@dnumpstartsL
423 \newif\ifl@dprintingcolumns%
424 \newif\ifpst@rtedL
425 \newcount\l@dnumpstartsL
426 %
```

`\ifledRcol` `\ifledRcol` is set to true in the Rightside environment. It must be not confused  
`\ifledRcol@` with `\ifledRcol@` which is set to true when a right line is processed, in `\Pages` or  
`\Columns`.

```
427 \newif\ifledRcol
428 \newif\ifledRcol@
429 %
```

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

```
430 \newif\ifnumberingR
431 %
```

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

```
432 \newif\ifafterendnumberingR%
433 %
```

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

```
434 \newif\ifXnote@%
435 %
```

## II.12 Booleans provided by other optional packages which are required in any case

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

```
436 \providebool{indtl@innote}%
437 \providebool{indtl@notenumber}%
438 %
```

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

```
439 \newif\ifinstanzaL%
440 \newif\ifinstanzaR%
441 %
```

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

```
442 \newif\ifinastanzaL%
443 \newif\ifinastanzaR%
444 %
```

## III Sectioning commands

`\section@num` You use `\beginnumbering` and `\endnumbering` to begin and end a line-numbered section of the text; the pair of commands may be used as many times as you like within one document to start and end multiple, separately line-numbered sections.  $\TeX$  will maintain and display a “section number” as a count named `\section@num` that counts how many `\beginnumbering` and `\resumnumbering` 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.

```
445 \newcount\section@num
446 \section@num=0
447 \let\extensionchars=\empty
448 %
```

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

```

449 \newif\ifnumbering
450 %

```

`\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.

```

451 \newcommand*{\beginnumbering}{%
452   \ifnumbering
453     \led@err@NumberingStarted
454     \endnumbering
455   \fi
456   \global\numberingtrue
457   \global\afterendnumberingRfalse%
458   \global\advance\section@num \@ne
459   \initnumbering@reg
460   \message{Section \the\section@num }%
461   \line@list@stuff{\jobname.\extensionchars\the\section@num}%
462   \ifcontinuousnumberingwithcolumns%
463     \unless\ifafterendnumberingR%
464       \unless\ifl@dpairing%
465         \ledRcoltrue%
466         \global\advance\section@numR \@ne%
467         \message{Section \the\section@numR R (continuousnumbering)}%
468         \line@list@stuffR{\jobname.\extensionchars\the\section@numR R}%
469         \ledRcolfalse%
470       \fi%
471     \fi%
472   \fi%
473   \l@dend@stuff
474   \setcounter{pstart}{1}
475   \ifl@dpairing
476     \global\l@dnumpstartsL \z@
477     \global\pst@rtedLfalse
478 %

```

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.

```

479 \else
480 \begingroup
481 \global\@afterindenttrue%In order to reestablish normal feature if the \
begingroup was not here
482 \initnumbering@quote
483 \ifwidthliketwocolumns%
484 \setwidthliketwocolumns%
485 \cuse{setpositionliketwocolumns@\columns@position}%
486 \fi%
487 \fi
488 \gdef\eled@sections@{ }%
489 \if@noeled@sec\else%
490 \makeatletter%
491 \InputIfFileExists%
492 {\l@auxdir\jobname.eledsec\the\section@num}%
493 {}%
494 {\led@warn@NoFile{\l@auxdir\jobname.eledsec\the\section@num}}%
495 \makeatother%
496 \immediate\openout\eled@sectioning@out=\l@auxdir\jobname.eledsec\the\
section@num\relax%
497 \fi%
498 }
499 \newcommand*{\initnumbering@reg}{%
500 \global\pst@rtedLfalse
501 \global\l@dnumpststartsL \z@
502 \global\absline@num \z@
503 \gdef\normal@page@break{}
504 \gdef\l@prev@pb{}
505 \gdef\l@prev@nopb{}
506 \global\line@num \z@
507 \global\subline@num \z@
508 \global\@lock \z@
509 \global\sub@lock \z@
510 \global\sublines@false
511 \global\let\next@page@num=\relax
512 \global\let\this@section@next@page@num=\relax%
513 \global\let\sub@change=\relax
514 \global\last@page@num=-10000%
515 \ifdefined\line@numR%
516 \line@numR=\z@%
517 \last@page@numR=\z@%
518 \fi%
519 \resetprevline@
520 \resetprevpage@num
521 \global\stopmsdata@inserted@true%
522 \global\let\@msdata@list\relax%
523 \global\csundef{\msdata@\add@msd@c @data}%
524 }
525 %
526 %

```

`\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.

```

527 \def\endnumbering{%
528   \ifnumbering
529     \global\numberingfalse
530     \normal@pars
531     \ifnum\l@dnumpstartsL=0%
532       \led@err@NumberingWithoutPstart%
533     \fi%
534 %

```

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`.

```

535   \global\page@num=\this@section@page@num%
536   \global\last@page@num=\this@section@last@page@num%
537   \global\let\next@page@num\this@section@next@page@num%
538   \ifl@dpairing
539     \global\pst@rtedLfalse
540   \else
541     \ifx\insertlines@list\empty\else
542       \global\noteschanged@true
543     \fi
544     \ifx\line@list\empty\else
545       \global\noteschanged@true
546     \fi
547   \fi
548   \ifnoteschanged@
549     \led@mess@NotesChanged
550   \fi
551 \else
552   \led@err@NumberingNotStarted
553 \fi
554 \autoparfalse
555 \ifnoeled@sec\else%
556   \immediate\closeout\eled@sectioning@out%
557 \fi%
558 \ifl@dpairing\else
559   \global\l@dnumpstartsL=\z@%
560 \endgroup
561 \fi
562 }
563 %

```

`\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>29</sup> The `\ifresumenumbering@start` `\pausenumbering@page@num` counter stores the `\this@section@next@page@num` when the `\pause@numbering` is called.

```

564 \newcount\pausenumbering@page@num%
565 \newcommand{\pausenumbering}{%

```

<sup>29</sup>Peter Wilson’s thanks to Wayne Sullivan, who suggested the idea behind these macros.

```

566 \ifx\this@section@next@page@num\relax%
567 \global\pausenumbering@page@num=0%
568 \else%
569 \global\pausenumbering@page@num=\this@section@next@page@num%
570 \fi%
571 \ifautopar\global\autopar@pausetrue\fi%
572 \endnumbering\global\numberingtrue}
573 %

```

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.

```

574 \newif\ifresumenumbering@start%
575 \newcommand*{\resumenumbering}{%
576 \ifnumbering
577 \ifautopar@pause\autopar\fi
578 \global\pst@rtedLtrue
579 \global\advance\section@num \@ne
580 \global\resumenumbering@starttrue%
581 \led@mess@SectionContinued{\the\section@num}%
582 \set@continuousnumberingforL%
583 \line@list@stuff{\jobname.\extensionchars\the\section@num}%
584 \ifcontinuousnumberingwithcolumns%
585 \unless\ifafterendnumberingR%
586 \unless\ifl@dpairing%
587 \ledRcoltrue%
588 \global\advance\section@numR \@ne%
589 \message{Section \the\section@numR R (continooousnumbering)}%
590 \line@list@stuffR{\jobname.\extensionchars\the\section@numR R}%
591 \ledRcolfalse%
592 \fi%
593 \fi%
594 \fi%
595 \l@dend@stuff
596 \ifl@dpairing\else%
597 \begingroup%
598 \initnumbering@quote%
599 \ifwidthliketwocolumns%
600 \setwidthliketwocolumns%
601 \csuse{setpositionliketwocolumns@\columns@position}%
602 \fi%
603 \fi%
604 \else
605 \led@err@NumberingShouldHaveStarted
606 \endnumbering
607 \beginnumbering
608 \fi}
609
610
611 %

```

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

```

612 \newcommand{\set@continuousnumberingforL}{%
613   \ifcontinuousnumberingwithcolumns%
614     \ifl@dpairing%
615       \unless\ifl@dpaging%
616         \global\c@pstartL=\c@pstart%
617         \fi%
618       \fi%
619     \fi%
620   }%
621   %

```

## 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 on what we need, we use one or other.

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

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

```

622 \newcommand*{\list@create}[1]{%
623   \global\let#1=\empty%
624 }%
625 %

```

`\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 .

```

626 \newcommand*{\list@clear}[1]{%
627   \global\let#1=\empty%
628 }
629 %

```

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



```

630 \newtoks\led@toksa \newtoks\led@toksb
631 \global\led@toksa={\}
632 \long\def\xright@appenditem#1\to#2{%
633   \global\led@toksb=\expandafter{#2}%
634   \xdef#2{\the\led@toksb\the\led@toksa\expandafter{#1}}%
635   \global\led@toksb={}}
636 %

```

`\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`.

```

637 \long\def\xleft@appenditem#1\to#2{%
638   \global\led@toksb=\expandafter{#2}%
639   \xdef#2{\the\led@toksa\expandafter{#1}\the\led@toksb}%
640   \global\led@toksb={}}
641 %

```

`\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.

```

642 \def\gl@p#1\to#2{\expandafter\gl@poff#1\gl@poff#1#2}
643 \long\def\gl@poff\#1#2\gl@poff#3#4{\gdef#4{#1}\gdef#3{#2}}
644
645 %

```

## V Line counting

### V.1 Choosing the system of lineation

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

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

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

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

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

`\bypage@false`

```

646 \newif\ifbypage@
647 \newif\ifbypstart@
648 %

```

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

```

\ifbypage@R49 \newif\ifbypage@R
\ifbypstart@R50 \newif\ifbypstart@R
651 %

```

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

```
652 \newcommand*{\lineation}[1]{
653 %
```

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

```
654 \ifnumbering
655 \led@err@LineationInNumbered
656 \else
657 %
```

If the argument is page.

```
658 \def\@tempa{#1}\def\@tempb{page}%
659 \ifx\@tempa\@tempb
660 \global\bypage@true
661 \global\bystart@false
662 \unless\ifnocritical@%
663 \Xpstart[] [false]%
664 \fi%
665 %
```

If the argument is pstart.

```
666 \else
667 \def\@tempb{pstart}%
668 \ifx\@tempa\@tempb
669 \global\bypage@false
670 \global\bystart@true
671 \unless\ifnocritical@%
672 \Xpstart%
673 \fi%
674 %
```

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

```
675 \else
676 \def\@tempb{section}
677 \ifx\@tempa\@tempb
678 \global\bypage@false
679 \global\bystart@false
680 \unless\ifnocritical@%
681 \Xpstart[] [false]%
682 \fi%
683 %
```

In other case, it is an error.

```
684 \else
685 \led@warn@BadLineation
686 \fi
687 \fi
688 \fi
689 \fi}}
690 %
```

## V.2 Line number margin

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

```

691 \newcount\line@margin%
692 \newcount\line@margin@columns%Only for parallel typesetting
693 \line@margin@columns=\m@ne%
694
695 \newcommand*\linenummargin}[1]{%
696   \l@dgetline@margin{#1}%
697   \ifnum\@l@tempcntb>\m@ne
698     \ifledRcol
699       \global\line@marginR=\@l@tempcntb
700       \led@warn@setting@in@rightside{\linenummargin}%
701     \else
702       \global\line@margin=\@l@tempcntb
703     \fi
704   \fi}}
705
706 \newcommand*\l@dgetline@margin}[1]{%
707   \def\@tempa{#1}\def\@tempb{left}%
708   \ifx\@tempa\@tempb
709     \@l@tempcntb \z@
710   \else
711     \def\@tempb{right}%
712     \ifx\@tempa\@tempb
713       \@l@tempcntb \@ne
714     \else
715       \def\@tempb{outer}%
716       \ifx\@tempa\@tempb
717         \@l@tempcntb \tw@
718       \else
719         \def\@tempb{inner}%
720         \ifx\@tempa\@tempb
721           \@l@tempcntb \thr@@
722         \else
723           \led@warn@BadLinenummargin
724           \@l@tempcntb \m@ne
725         \fi
726       \fi
727     \fi
728   \fi}
729
730 %

```

## V.3 Line number initialization and increment

`\c@firstlinenum` `\c@linenumincrement` 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.

```

731 \newcounter{firstlinenum}
732   \setcounter{firstlinenum}{5}
733 \newcounter{linenumincrement}
734   \setcounter{linenumincrement}{5}
735 %

```

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

```

736 \newcounter{firstsublinenum}
737   \setcounter{firstsublinenum}{5}
738 \newcounter{sublinenumincrement}
739   \setcounter{sublinenumincrement}{5}
740
741 %

```

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

```

\linenumincrement
\firstsublinenum
\sublinenumincrement
742
743 \newcommand*\firstlinenum}[1]{%
744   \ifledRcol%
745     \setcounter{firstlinenumR}{#1}%
746     \led@warn@setting@in@rightside{\firstlinenum}%
747   \else%
748     \setcounter{firstlinenum}{#1}%
749   \fi%
750 }
751 \newcommand*\linenumincrement}[1]{%
752   \ifledRcol%
753     \setcounter{linenumincrementR}{#1}%
754     \led@warn@setting@in@rightside{\linenumincrement}%
755   \else%
756     \setcounter{linenumincrement}{#1}%
757   \fi%
758 }
759 \newcommand*\firstsublinenum}[1]{%
760   \ifledRcol%
761     \setcounter{firstsublinenumR}{#1}%
762     \led@warn@setting@in@rightside{\firstsublinenum}%
763   \else%
764     \setcounter{firstsublinenum}{#1}%
765   \fi%
766 }
767 \newcommand*\sublinenumincrement}[1]{%
768   \ifledRcol%
769     \setcounter{sublinenumincrementR}{#1}%
770     \led@warn@setting@in@rightside{\sublinenumincrement}%
771   \else%
772     \setcounter{sublinenumincrement}{#1}%
773   \fi%
774 }
775

```

```
776 %
```

## 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.

```
777 \newcount\lock@disp
778 \newcommand{\lockdisp}[1]{%
779   \l@dgetlock@disp{#1}%
780   \ifnum\@l@dtempcntb>\m@ne
781     \global\lock@disp=\@l@dtempcntb
782   \else
783     \led@warn@BadLockdisp
784   \fi}}
785 \newcommand*{\l@dgetlock@disp}[1]{
786   \def\@tempa{#1}\def\@tempb{first}%
787   \ifx\@tempa\@tempb
788     \@l@dtempcntb \z@
789   \else
790     \def\@tempb{last}%
791     \ifx\@tempa\@tempb
792       \@l@dtempcntb \@ne
793     \else
794       \def\@tempb{all}%
795       \ifx\@tempa\@tempb
796         \@l@dtempcntb \tw@
797       \else
798         \@l@dtempcntb \m@ne
799       \fi
800     \fi
801   \fi}
802
803 %
```

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

```
804 \newcount\sublock@disp
805 \newcommand{\sublockdisp}[1]{%
806   \l@dgetlock@disp{#1}%
807   \ifnum\@l@dtempcntb>\m@ne
808     \global\sublock@disp=\@l@dtempcntb
809   \else
810     \led@warn@BadSublockdisp
811   \fi}}
812
813 %
```

## V.5 Line number style

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

```
814 \newcommand*{\linenumberstyle}[1]{%
815   \def\linenumrep##1{\@nameuse{#1}{##1}}
816 \newcommand*{\sublinenumberstyle}[1]{%
817   \def\sublinenumrep##1{\@nameuse{#1}{##1}}
818 %
```

Initialise the number styles to arabic.

```
819 \linenumberstyle{arabic}
820 \let\linenumr@p\linenumrep
821 \sublinenumberstyle{arabic}
822 \let\sublinenumr@p\sublinenumrep
823
824 %
```

## V.6 Line number printing

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

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

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

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

```
825 \newlength{\linenumsep}
826 \setlength{\linenumsep}{1pc}
827 \newcommand*{\numlabfont}{\normalfont\scriptsize}
828 \newcommand*{\ledlinenum}{%
829   \bgroup%
830   \ifluatex%
831     \textdir TLT%
832   \fi%
833   \numlabfont%
834   \unless\iflinenumannotationotherside%
835     \ifdefstring{\Xlinenumannotationposition@side}{before}{%
836       \l@wrap@ifnotemptybox{\Xwraplinenumannotation@side}{%
```

```

837     \csuse{annot@the\absline@num @the\section@num}%
838     }%
839   }%
840   \fi%
841   \linenumrep{\line@num}%
842   \ifsublines@
843     \ifnum\subline@num>0\relax
844       \unskip%
845       \Xsublinesep@side%
846       \sublinenumrep{\subline@num}%
847     \fi
848   \fi%
849   \unless\iflinenumannotationotherside%
850     \ifdefstring{\Xlinenumannotationposition@side}{after}{%
851       \l@wrap@ifnotemptybox{\Xwraplinenumannotation@side}{%
852         \csuse{annot@the\absline@num @the\section@num}%
853       }%
854     }%
855   \fi%
856   \egroup%
857 }%
858
859 \newcommand*{\leftlinenum}{%
860   \ledlinenum%
861   \kern\linenumsep%
862 }%
863 \newcommand*{\rightlinenum}{%
864   \kern\linenumsep%
865   \ledlinenum%
866 }%
867
868 %

```

`\leftlinenumannotation` and `\rightlinenumannotation` prints the line number annotation; when user-defined it must be printed on the side opposite the main line number.

```

869 \newcommand*{\leftlinenumannotation}{%
870   \l@wrap@ifnotemptybox{\Xwraplinenumannotation@side}{%
871     \csuse{annot@the\absline@num @the\section@num}%
872   }%
873   \kern\linenumsep%
874 }%
875 \newcommand*{\rightlinenumannotation}{%
876   \kern\linenumsep%
877   \l@wrap@ifnotemptybox{\Xwraplinenumannotation@side}{%
878     \csuse{annot@the\absline@num @the\section@num}%
879   }%
880 }%
881 %

```

## 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`.

```
882 \newcount\line@num
883 %
```

`\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.

```
884 \newcount\subline@num
885 %
```

`\ifsublines@` We maintain an associated flag, `\ifsublines@`, to tell us whether we're within a sub-line range or not (and the equivalent for right side: `\ifsublines@R`)

`\ifsublines@R`  
`\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.

```
886 \newif\ifsublines@
887 \newif\ifsublines@R%
888 %
```

`\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.

```
889 \newcount\absline@num
890 %
```



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

## V.8 Line number locking counter

`\@lock` `\sub@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.

```

891 \newcount \@lock
892 \newcount \sub@lock
893 %

```

## V.9 Line number associated to lemma

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

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

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

```
23|35|0|24|3|0|OT1/cmr/m/n.
```

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

- `\insertlines@list`: the line numbers of lines that have footnotes or other insertions. These are the absolute numbers where the corresponding lemmas begin. This list contains one entry for every footnote in the section; one lemma may contribute no footnotes or many footnotes. This list is used by `\add@inserts` within `\do@line`, to tell it where to insert notes.
- `\actionlines@list`: a list of absolute line numbers at which we are to perform special actions; these actions are specified by the `\actions@list` list defined below.
- `\actions@list`: action codes corresponding to the line numbers in `\actionlines@list`. These codes tell `reledmac` what action it is supposed to take at each of these lines. One action, the page-start action, is generated behind the scenes by `reledmac` itself; the others, for specifying sub-lineation, line-number locking, and line-number alteration, are generated only by explicit commands in your input file.

The page-start and line-number-alteration actions require arguments, to specify the new values for the page or line numbers; instead of storing those arguments in another list, we have chosen the action-code values so that they can encode both the action and the argument in these cases. Action codes greater than  $-1000$  are page-start actions, and the code value is the page number; action codes less than  $-5000$  specify line numbers, and the code value is a transformed version of the line number; action codes between these two values specify other actions which require no argument.

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

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

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

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

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

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

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

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

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

An action code of  $-5000$  or less sets the current visible line number (either the line number or the sub-line number, whichever is currently being advanced) to a specific positive value. The value of the code is  $-(5000 + n)$ , where  $n$  is the value (always  $\geq 0$ ) assigned to the current line number. Action codes of this type are added to the list by the `\set@line@action` macro, as called to implement the `\advance@line` and `\set@line` macros: this action only occurs when the user has specified some change to the line numbers using those macros. Normally `reledmac` computes the visible line numbers from the absolute line numbers with reference to the other action codes and the settings they invoke; it does not require an entry in the action-code list for every line.

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

Here are the commands to create these lists:

```
894 \list@create{\line@list}
895 \list@create{\insertlines@list}
896 \list@create{\actionlines@list}
897 \list@create{\actions@list}
898 \list@create{\annot@list}%
899 %
```

`\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
900 \newcount\page@num
901 \newcount\endpage@num
902 \newcount\endline@num
903 \newcount\endsubline@num
904 %
```

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

```
905 \newcount\this@section@page@num%
906 %
```

`\ifnoteschanged@` 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 `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 `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.

```
907 \newif\ifnoteschanged@
908 %
```

`\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@
09 \newcommand*\resetprevline@{%
910 \def\do##1{\global\csundef{prevline##1}}%
911 \dolistloop{\@series}%
912 }
913 %
```

`\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@14 \newcommand*{\resetprevpage@num}{%
915   \def\do##1{%
916     \ifcsdef{prevpage##1@num}{%
917       \global\csname prevpage##1@num\endcsname=\z@%
918       \global\csname prevpage##1@numR\endcsname=\z@%
919     }%
920   }%
921   \ifcsdef{##1prevpage@num}{%
922     \global\csname ##1prevpage@num\endcsname=\z@%
923     \global\csname ##1prevpage@numR\endcsname=\z@%
924   }%
925   }%
926 }%
927 \dolistloop{\@series}%
928 }
929 %

```

## 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.

```

930 \newread\@inputcheck
931 \newcommand*{\read@linelist}[1]{%
932   \ifledRcol%
933     \list@clearing@regR%
934   \else%
935     \list@clearing@reg%
936   \fi%
937 %

```

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).

```

938   \list@clear{\maxlinesinpar@list}
939 %

```

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 `\pausenumbers` was executed.

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

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

```

940 \get@linelistfile{#1}%
941 \@stopmsd%Security if last \endms{} is forgotten
942 \unless\ifledRcol%Get the last line of the last page
943   \csnumgdef{@lastabsline@forpage@the\page@num}{\the\absline@num}%
944   \csnumgdef{@lastline@forpage@the\page@num}{\the\line@num}%
945   \csnumgdef{@lastsubline@forpage@the\page@num}{\the\subline@num}%
946 \else%
947   \csnumgdef{@lastabsline@forpageR@the\page@numR}{\the\absline@numR}%
948   \csnumgdef{@lastline@forpageR@the\page@numR}{\the\line@numR}%
949   \csnumgdef{@lastsubline@forpageR@the\page@numR}{\the\subline@numR}%
950 \fi%
951 \endgroup
952 %

```

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.

```

953 \ifledRcol
954   \global\page@numR=\m@ne
955   \ifx\actionlines@listR\empty
956     \gdef\next@actionlineR{1000000}%
957   \else
958     \gl@p\actionlines@listR\to\next@actionlineR
959     \gl@p\actions@listR\to\next@actionR
960   \fi
961 \else
962   \global\page@num=\m@ne
963   \ifx\actionlines@list\empty
964     \gdef\next@actionline{1000000}%
965   \else
966     \gl@p\actionlines@list\to\next@actionline
967     \gl@p\actions@list\to\next@action
968   \fi
969 \fi
970 }
971 %

```

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

```

972 \newcommand*{\list@clearing@reg}{%
973   \list@clear{\line@list}%
974   \list@clear{\insertlines@list}%
975   \list@clear{\actionlines@list}%
976   \list@clear{\actions@list}%
977   \list@clear{\linesinpar@listL}%
978   \list@clear{\linesonpage@listL}%
979 }%

```

```
980 %
```

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

```
981 \newcommand*\get@linelistfile}[1]{%
982   \InputIfFileExists{\l@auxdir#1}{%
983     \global\noteschanged@false
984     \begingroup
985       \catcode`\[=1 \catcode`\]=2
986       \makeatletter \catcode`\^M=9}{%
987     \led@warn@NoFile{\l@auxdir#1}%
988     \global\noteschanged@true
989     \begingroup}%
990 }
991
992 %
```

This version of `\read@linelist` creates list macros containing data for the entire section, so they could get rather large. It would be no more difficult to read the line-list file incrementally rather than all at once: we could read, at the start of each paragraph, only the commands relating to that paragraph. But this would require that we have two line-lists open at once, one for reading, one for writing, and on systems without version numbers we would have to do some file renaming outside of  $\LaTeX$  for that to work. We have retained this slower approach to avoid that sort of hacking about, but have provided the `\pausenumbering` and `\resumenumbering` macros to help you if you run into macro memory limitations (see 5.2.7 p. 19 above).

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

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

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

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

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

```
993 \newcommand{\line@list@version}[1]{%
994   \IfStrEq{#1}{\this@line@list@version}%
995   }%
996   {\ifledRcol%
997     \led@warn@Obsoleter{\jobname.\extensionchars\the\section@num}%
```

```

998     \else%
999         \led@warn@Obsolete{\jobname.\extensionchars\the\section@num}%
1000     \fi%
1001     \endinput%
1002 }%
1003 }%
1004 %

```

`\@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`.

```

1005
1006 \newcommand*{\@nl}[2]{%
1007     \@page{#1}%
1008     \ifledRcol%
1009         \@nl@regR%
1010         \csnumgdef{@page@forlineR@the\section@numR @the\absline@numR}{#1}%
1011     \else%
1012         \@nl@reg%
1013     \fi%
1014 }
1015 \newcommand*{\@nl@reg}{%
1016     \ifx\l@dchset@num\relax \else
1017         \advance\absline@num \@ne
1018         \csgdef{l@dchset@num@the\absline@num}{}%To remember this line have
1019         \set@line@action
1020         \let\l@dchset@num=\relax
1021         \advance\absline@num \m@ne
1022         \advance\line@num \m@ne
1023     \fi
1024 %

```

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

```

1025     \reset@current@annot%
1026     \advance\absline@num \@ne
1027         \ifx\next@page@num\relax \else
1028             \page@action
1029             \let\next@page@num=\relax
1030         \fi

```

```

1031     \ifx\sub@change\relax \else
1032       \ifnum\sub@change>\z@
1033         \sublines@true
1034       \else
1035         \sublines@false
1036       \fi
1037       \sub@action
1038       \let\sub@change=\relax
1039     \fi
1040 %

```

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

```

1041     \ifcase\@lock
1042       \or
1043         \@lock \tw@
1044       \or \or
1045         \@lock \z@
1046     \fi
1047     \ifcase\sub@lock
1048       \or
1049         \sub@lock \tw@
1050       \or \or
1051         \sub@lock \z@
1052     \fi
1053 %

```

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

```

1054     \ifsublines@
1055       \ifnum\sub@lock<\tw@
1056         \advance\subline@num \@ne
1057       \fi
1058     \else
1059       \ifnum\@lock<\tw@
1060         \advance\line@num \@ne \subline@num \z@
1061       \fi
1062     \fi}
1063 %
1064 %

```

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

`\this@section@last@page@num`

`\@page`

```

1065 \newcount\last@page@num
1066 \last@page@num=-10000
1067
1068 \newcount\this@section@last@page@num%
1069 \this@section@last@page@num=-10000%
1070
1071 \newcommand*{\@page}[1]{%
1072   \ifledRcol
1073     \ifnum #1=\last@page@numR
1074     \else
1075       \csnumgdef{\lastabsline@forpageR@the\page@numR}{\the\absline@numR}%

```



```

1076     \csnumgdef{@lastline@forpage@the\page@numR}{\the\line@numR}%
1077     \csnumgdef{@lastsubline@forpage@the\page@numR}{\the\subline@numR}%
1078     \ifbypage@R
1079     \letcs{\prevsection@last@page@num}{last@page@num@the\numexpr\
section@num-1\relax}%
1080     \ifundef{\prevsection@last@page@num}{\def\prevsection@last@page@num
{-999}}{}%
1081     \ifx\l@dchset@num\relax%Not resetting if preceded by a \setlinenum
1082     \ifboolexpr{%
1083     (bool{continuousnumberingwithcolumns} or bool{
resumenumberingR@start})%
1084     and test {\ifnumequal{\last@page@numR}{-10000}}%
1085     and not (%
1086     bool{continuousnumberingwithcolumns}%
1087     and test {\ifnumcomp{\prevsection@last@page@num}{<}{#1}}%
1088     )%
1089     }%
1090     {}%
1091     {%
1092     \line@numR \z@%
1093     \subline@numR \z@%
1094     \global\csdef{reset@lineR\the\numexpr\absline@numR+1\relax @\
the\section@numR}{}%
1095     }%
1096     \fi%
1097     \fi
1098     \page@numR=#1%
1099     \global\this@section@page@numR=#1%
1100     \global\last@page@numR=#1%
1101     \global\this@section@last@page@numR=#1%
1102     \def\next@page@numR{#1}%
1103     \gdef\this@section@next@page@numR{#1}%
1104     \fi
1105     \else
1106     \ifnum #1=\last@page@num
1107     \else
1108     \csnumgdef{@lastabsline@forpage@the\page@num}{\the\absline@num}%
1109     \csnumgdef{@lastline@forpage@the\page@num}{\the\line@num}%
1110     \csnumgdef{@lastsubline@forpage@the\page@num}{\the\subline@num}%
1111     \ifbypage@
1112     \ifx\l@dchset@num\relax%Not resetting if preceded by a \setlinenum
1113     \line@num \z@%
1114     \subline@num \z@%
1115     \global\csdef{reset@line\the\numexpr\absline@num+1\relax @the\
section@num}{}%
1116     \fi%
1117     \fi
1118     \page@num=#1%
1119     \global\this@section@page@num=#1%
1120     \last@page@num=#1%
1121     \global\this@section@last@page@num=#1%
1122     \global\csdef{last@page@num@the\section@num}{#1}%
1123     \gdef\next@page@num{#1}%
1124     %

```

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

```

1125 \gdef\this@section@next@page@num{#1}%
1126 \listxadd{\normal@page@break}{\the\absline@num}
1127 \fi
1128 \fi}
1129 %

```

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

```

1130 \newcommand*\@pend}[1]{%
1131 \newcommand*\@pendR}[1]{%
1132 \newcommand*\@lopL}[1]{%
1133 \newcommand*\@lopR}[1]{%
1134
1135 %

```

`\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.

```

1136 \newcommand*\sub@on{%
1137 \ifboolexpr{%
1138   bool{sublines@}%
1139   or%
1140   (bool{sublines@R} and bool{ledRcol})%
1141 }{%
1142   \let\sub@change=\relax
1143 }{%
1144   \def\sub@change{-1}%
1145 }%
1146 }%
1147 \newcommand*\sub@off{%
1148 \ifboolexpr{%
1149   bool{sublines@}%
1150   or%
1151   (bool{sublines@R} and bool{ledRcol})%
1152 }{%
1153   \def\sub@change{-1}%
1154 }{%
1155   \let\sub@change=\relax
1156 }%
1157 }%
1158
1159 %

```

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

```

1160
1161 \newcommand*\@adv}[1]{%

```

```

1162 \ifsublines@
1163 \ifledRcol
1164 \advance\subline@numR by #1\relax
1165 \ifnum\subline@numR<\z@
1166 \led@warn@BadAdvancelineSubline
1167 \subline@numR \z@
1168 \fi
1169 \else
1170 \advance\subline@num by #1\relax
1171 \ifnum\subline@num<\z@
1172 \led@warn@BadAdvancelineSubline
1173 \subline@num \z@
1174 \fi
1175 \fi
1176 \else
1177 \ifledRcol
1178 \advance\line@numR by #1\relax
1179 \ifnum\line@numR<\z@
1180 \led@warn@BadAdvancelineLine
1181 \line@numR \z@
1182 \fi
1183 \else
1184 \advance\line@num by #1\relax
1185 \ifnum\line@num<\z@
1186 \led@warn@BadAdvancelineLine
1187 \line@num \z@
1188 \fi
1189 \fi
1190 \fi
1191 \set@line@action}
1192
1193 %

```

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

```

1194
1195 \newcommand*{\@set}[1]{%
1196 \ifledRcol
1197 \ifsublines@R%
1198 \subline@numR=#1\relax
1199 \else
1200 \line@numR=#1\relax
1201 \fi
1202 \set@line@action
1203 \else
1204 \ifsublines@
1205 \subline@num=#1\relax
1206 \else
1207 \line@num=#1\relax
1208 \fi
1209 \set@line@action
1210 \fi}
1211

```

1212 %

`\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.

```
1213
1214 \newcommand*{\l@d@set}[1]{%
1215   \ifledRcol
1216     \line@numR=#1\relax
1217     \advance\line@numR \@ne
1218     \def\l@dchset@num{#1}
1219   \else
1220     \line@num=#1\relax
1221     \advance\line@num \@ne
1222     \def\l@dchset@num{#1}
1223   \fi}
1224 \let\l@dchset@num\relax
1225
1226 %
```

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

```
1227
1228 \newcommand*{\page@action}{%
1229   \ifledRcol
1230     \xright@appenditem{\the\absline@numR}\to\actionlines@listR
1231     \xright@appenditem{\next@page@numR}\to\actions@listR
1232   \else
1233     \xright@appenditem{\the\absline@num}\to\actionlines@list
1234     \xright@appenditem{\next@page@num}\to\actions@list
1235   \fi}
1236 %
```

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

```
1237
1238 \newcommand*{\set@line@action}{%
1239   \ifledRcol
1240     \xright@appenditem{\the\absline@numR}\to\actionlines@listR
1241     \ifsublines@R%
1242       \@l@tempcnta=-\subline@numR
1243     \else
1244       \@l@tempcnta=-\line@numR
1245     \fi
1246     \advance\@l@tempcnta by -5000\relax
1247     \xright@appenditem{\the\@l@tempcnta}\to\actions@listR
1248   \else
1249     \xright@appenditem{\the\absline@num}\to\actionlines@list
1250     \ifsublines@
1251       \@l@tempcnta=-\subline@num
1252     \else
```

```

1253     \@l@dtempcnta=-\line@num
1254     \fi
1255     \advance\@l@dtempcnta by -5000\relax
1256     \xright@appenditem{\the\@l@dtempcnta}\to\actions@list
1257     \fi}
1258 %

```

**\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.

```

1259 \newcommand*\sub@action){%
1260 \ifledRcol
1261 \xright@appenditem{\the\absline@numR}\to\actionlines@listR
1262 \ifsublines@R%
1263 \xright@appenditem{-1001}\to\actions@listR
1264 \else
1265 \xright@appenditem{-1002}\to\actions@listR
1266 \fi
1267 \else
1268 \xright@appenditem{\the\absline@num}\to\actionlines@list
1269 \ifsublines@
1270 \xright@appenditem{-1001}\to\actions@list
1271 \else
1272 \xright@appenditem{-1002}\to\actions@list
1273 \fi
1274 \fi}
1275 %
1276

```

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

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

```

1277 \newcommand*\lock@on){\futurelet\next\do@lockon}
1278
1279 \newcommand*\do@lockon){%
1280 \ifx\next\lock@off
1281 \global\let\lock@off=\skip@lockoff
1282 \else
1283 \ifledRcol
1284 \do@lockonR
1285 \else
1286 \do@lockonL
1287 \fi
1288 \fi}
1289
1290
1291 \newcommand*\do@lockonL){%
1292 \xright@appenditem{\the\absline@num}\to\actionlines@list
1293 \ifsublines@

```

```

1294 \xright@appenditem{-1005}\to\actions@list
1295 \ifnum\sub@lock=\z@
1296 \sub@lock \@ne
1297 \else
1298 \ifnum\sub@lock=\thr@@
1299 \sub@lock \@ne
1300 \fi
1301 \fi
1302 \else
1303 \xright@appenditem{-1003}\to\actions@list
1304 \ifnum\@lock=\z@
1305 \@lock \@ne
1306 \else
1307 \ifnum\@lock=\thr@@
1308 \@lock \@ne
1309 \fi
1310 \fi
1311 \fi}
1312 %
1313 %

```

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

```

\do@lockoff
\do@lockoffL
\skip@lockoff
1314 \newcommand*{\do@lockoffL}{%
1315 \xright@appenditem{\the\absline@num}\to\actionlines@list
1316 \ifsublines@
1317 \xright@appenditem{-1006}\to\actions@list
1318 \ifnum\sub@lock=\tw@
1319 \sub@lock \thr@@
1320 \else
1321 \sub@lock \z@
1322 \fi
1323 \else
1324 \xright@appenditem{-1004}\to\actions@list
1325 \ifnum\@lock=\tw@
1326 \@lock \thr@@
1327 \else
1328 \@lock \z@
1329 \fi
1330 \fi}
1331
1332 \newcommand*{\do@lockoff}{%
1333 \reset@current@annot%
1334 \ifledRcol
1335 \do@lockoffR
1336 \else
1337 \do@lockoffL
1338 \fi}
1339 \newcommand*{\skip@lockoff}{\global\let\lock@off=\do@lockoff}
1340 \global\let\lock@off=\do@lockoff
1341
1342 %

```

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

```

1343 \newcommand*{\n@num}{%
1344   \ifledRcol%
1345     \xright@appenditem{\the\absline@numR}\to\actionlines@listR
1346     \xright@appenditem{-1007}\to\actions@listR
1347   \else%
1348     \xright@appenditem{\the\absline@num}\to\actionlines@list%
1349     \xright@appenditem{-1007}\to\actions@list%
1350   \fi%
1351 }%
1352
1353 %

```

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

```

1354 \newcommand*{\n@num@stanza}{%
1355   \ifledRcol%
1356     \xright@appenditem{\the\absline@numR}\to\actionlines@listR%
1357     \xright@appenditem{-1008}\to\actions@listR%
1358   \else%
1359     \xright@appenditem{\the\absline@num}\to\actionlines@list%%
1360     \xright@appenditem{-1008}\to\actions@list%
1361   \fi%
1362 }
1363 %

```

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

```

\hidenumberingonleftpage
\hidenumberingonrightpage
1364 \newif\ifl@dhiddenumber
1365 \newcommand*{\hidenumbering}{%
1366   \ifledRcol%
1367     \write\linenum@outR{\string\hide@num}%
1368   \else%
1369     \write\linenum@out{\string\hide@num}%
1370   \fi%
1371 }%
1372 \newcommand*{\hide@num}{%
1373   \ifledRcol%
1374     \xright@appenditem{\the\absline@numR}\to\actionlines@listR%
1375     \xright@appenditem{-1009}\to\actions@listR%
1376   \else%
1377     \xright@appenditem{\the\absline@num}\to\actionlines@list%%
1378     \xright@appenditem{-1009}\to\actions@list%
1379   \fi%
1380 }
1381 \newcommand*{\hidenumberingonleftpage}{%
1382   \ifledRcol%
1383     \write\linenum@outR{\string\hide@num@left}%
1384   \else%
1385     \write\linenum@out{\string\hide@num@left}%
1386   \fi%

```

```

1387 }%
1388
1389 \newcommand*{\hide@num@left}{%
1390   \ifledRcol%
1391     \ifodd\page@numR\else%
1392       \xright@appenditem{\the\absline@numR}\to\actionlines@listR%
1393       \xright@appenditem{-1009}\to\actions@listR%
1394     \fi%
1395   \else%
1396     \ifodd\page@num\else%
1397       \xright@appenditem{\the\absline@num}\to\actionlines@list%%
1398       \xright@appenditem{-1009}\to\actions@list%
1399     \fi%
1400   \fi%
1401 }%
1402
1403 \newcommand*{\hidenumberingonrightpage}{%
1404   \ifledRcol%
1405     \write\linenum@outR{\string\hide@num@right}%
1406   \else%
1407     \write\linenum@out{\string\hide@num@right}%
1408   \fi%
1409 }%
1410
1411 \newcommand*{\hide@num@right}{%
1412   \ifledRcol%
1413     \ifodd\page@numR%
1414       \xright@appenditem{\the\absline@numR}\to\actionlines@listR%
1415       \xright@appenditem{-1009}\to\actions@listR%
1416     \fi%
1417   \else%
1418     \ifodd\page@num%
1419       \xright@appenditem{\the\absline@num}\to\actionlines@list%%
1420       \xright@appenditem{-1009}\to\actions@list%
1421     \fi%
1422   \fi%
1423 }%
1424
1425 %

```

`\@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`.

```

1426 \newcount\insert@count
1427 %

```

- #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.

```
1428 \newcommand*\dummy@ref}[2]{#2}
1429 %
```

`\@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.

```
1430 \newcommand*\@ref}[2]{%
1431   \ifledRcol%
1432     \@ref@regR{#1}{#2}%
1433   \else%
1434     \@ref@reg{#1}{#2}%
1435   \fi%
1436 }%
1437 \newcommand*\@ref@reg}[2]{%
1438   \global\insert@count=#1\relax
1439   \global\advance\@edtext@level by 1%
1440   \loop\ifnum\insert@count>\z@
1441     \xright@appenditem{\the\absline@num}\to\insertlines@list
1442     \global\advance\insert@count \m@ne
1443   \repeat
1444 %
```

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.

```
1445 \begingroup
1446   \let\@ref=\dummy@ref
1447   \let\@lopL@\gobble
1448   \let\page@action=\relax
1449   \let\sub@action=\relax
1450   \let\set@line@action=\relax
1451   \let\@lab=\relax
1452   \let\@lemma=\relax%
1453   \let\@sw@\gobblethree%
1454   \let\store@annot@to@absline@\gobble%
1455   #2
1456   \global\endpage@num=\page@num
1457   \global\endline@num=\line@num
1458   \global\endsubline@num=\subline@num
1459   \global\let\endcurrent@annot=\current@annot%
1460 \endgroup
1461 %
```

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

```
1462 \xright@appenditem%
1463   {\the\page@num|\the\line@num|%
1464     \ifsublines@ \the\subline@num \else 0\fi|%
1465     \the\endpage@num|\the\endline@num|%
1466     \ifsublines@ \the\endsubline@num \else 0\fi}\to\line@list
```

```

1467 \xright@appenditem%
1468   {\current@annot|\endcurrent@annot}\to\annot@list%
1469 %

```

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

```

1470   \@ref@reg@parse{#2}%
1471 %

```

Decrease edtext level counter.

```

1472 \global\advance\@edtext@level by -1%
1473 }
1474 %

```

`\@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.

```

1475 \newcommand{\@ref@reg@parse}[1]{%
1476   \expandafter\list@create\expandafter{\csname sw@list@edtext@tmp@the\
@edtext@level\endcsname}%
1477   \providebool{lemmacommand@the\@edtext@level}%
1478   \boolfalse{lemmacommand@the\@edtext@level}%
1479 %

```

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

```

1480   #1%
1481 %

```

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.

```

1482   \ifnum\@edtext@level>0%
1483     \def\create@this@edtext@level{\expandafter\list@create\expandafter{\
csname sw@list@edtext@the\@edtext@level\endcsname}}%
1484     \ifcsundef{sw@list@edtext@the\@edtext@level}{\create@this@edtext@level
}{}%
1485     \letcs{\@tmp}{sw@list@edtext@the\@edtext@level}%
1486     \letcs{\@tmpp}{sw@list@edtext@tmp@the\@edtext@level}
1487     \xright@appenditem{\expandonce\@tmpp}\to\@tmp%
1488     \global\cslet{sw@list@edtext@the\@edtext@level}{\@tmp}%
1489   \fi%
1490 %
1491 }
1492 %
1493 %

```

`\@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.

```

1494 \newcommand{\@ref@later}[1]{%
1495   \global\advance\@edtext@level by \@ne%
1496   \ifledRcol%
1497     \@ref@reg@parseR{#1}%
1498   \else%
1499     \@ref@reg@parse{#1}%
1500   \fi%
1501   \global\advance\@edtext@level by -\@ne%
1502 }%
1503 %

```

## 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`.

```

1504 \newwrite\linenum@out
1505 %

```

`\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 elegant 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`.

```

1506 \newif\iffirst@linenum@out
1507 \iffirst@linenum@out@true
1508 %

```

`\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 a older version, that means the commands used inside are deprecated, and we can't use them.

```

1509 \newcommand{\this@line@list@version}{8}%
1510 %

```

`\line@list@stuff` The `\line@list@stuff{<file>}` macro, which is called by `\beginnumbering`, performs all the line-list operations needed at the start of a section. Its argument is the name of the line-list file.

`\next@line@list@stuff`

```

1511 \let\next@line@list@stuff\relax%
1512 \newcommand*{\line@list@stuff}[1]{%
1513 %

```

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

```

1514 \global\newtoggle{notfirstrun@#1}%
1515 \IfFileExists{\l@auxdir#1}%
1516   {\global\toggletrue{notfirstrun@#1}}%
1517   {\global\togglefalse{notfirstrun@#1}}%
1518 %

```

A internal hook (not used yet).

```

1519 \next@line@list@stuff%
1520 \global\let\next@line@list@stuff\relax%
1521 %

```

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

```

1522 \read@linelist{#1}%
1523 %

```

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  $n$ .

```

1524 \iffirst@linenum@out@
1525   \global\first@linenum@out@false%
1526   \immediate\openout\linenum@out=\l@auxdir#1\relax%
1527   \immediate\write\linenum@out{\string\line@list@version{\
this@line@list@version}}%
1528   \ifl@dpaging%
1529     \immediate\write\linenum@out{\string\@par@sync@option{\
@par@this@sync@option}}%
1530   \fi%
1531 \else
1532 %

```

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.

```

1533   \if@minipage%
1534     \leavevmode%
1535   \fi%
1536   \closeout\linenum@out%
1537   \openout\linenum@out=\l@auxdir#1\relax%
1538   \write\linenum@out{\string\line@list@version{\this@line@list@version}}
1539 %
1539   \ifl@dpaging%
1540     \write\linenum@out{\string\@par@sync@option{\@par@this@sync@option}}
1541 %
1542   \fi%
1543 }%

```

```
1544 %
1545 %
```

`\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 `\sameparallelpagelnumber` 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.

```
1546 \newcommand*{\new@line}{%
1547   \ifnumberline%
1548   \IfStrEq{\led@pb@setting}{after}%
1549     {\xifinlist{\the\absline@num}{\l@prev@nopb}%
1550       {\xifinlist{\the\absline@num}{\normal@page@break}%
1551         {\numgdef{\@next@page}{\c@par@page+\@ne}%
1552           \write\linenum@out{\string\@nl[\@next@page] [\@next@page]}}%
1553         }%
1554         {\write\linenum@out{\string\@nl[\the\@this@page] [\thepage]}}}%
1555     }%
1556     {\write\linenum@out{\string\@nl[\the\@this@page] [\thepage]}}}%
1557   {}%
1558   \IfStrEq{\led@pb@setting}{before}%
1559     {\numdef{\next@absline}{\the\absline@num+\@ne}%
1560     \xifinlist{\next@absline}{\l@prev@nopb}%
1561       {\xifinlist{\the\absline@num}{\normal@page@break}%
1562         {\numgdef{\nc@page}{\c@par@page+\@ne}%
1563           \write\linenum@out{\string\@nl[\nc@page] [\nc@page]}}%
1564         }%
1565         {\write\linenum@out{\string\@nl[\the\@this@page] [\thepage]}}}%
1566       }%
1567       {\write\linenum@out{\string\@nl[\the\@this@page] [\thepage]}}}%
1568     }%
1569   {}%
1570   \IfStrEqCase{\led@pb@setting}%
1571     {%
1572     {before}{\relax}%
1573     {after}{\relax}%
1574   }[%
1575     \write\linenum@out{\string\@nl[\the\@this@page] [\thepage]}}%
1576   ]%
1577   \fi%
1578 }
```

```

1579 \newcount\@this@cc@page%%
1580 \newcommand{\set@this@cc@page}{%
1581   \ifboolexpr{%
1582     bool{sameparallelpagelnumber}%
1583     or bool{prevpgnotnumbered}%
1584   }%
1585   {\global\@this@cc@page=\c@par@page}%
1586   {\global\@this@cc@page=\c@page}%
1587 }%
1588 %
1589 %

```

`\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.

```

1590 \newcommand*{\flag@start}{%
1591   \ifledRcol%
1592     \edef\next{\write\linenum@outR{%
1593       \string\@ref[\the\insert@countR] []}%
1594     }%
1595   \else%
1596     \edef\next{\write\linenum@out{%
1597       \string\@ref[\the\insert@count] []}%
1598     }%
1599   \fi%
1600   \next%
1601 }%
1602 \newcommand*{\flag@end}{%
1603   \ifledRcol%
1604     \write\linenum@outR{]}%
1605   \else%
1606     \write\linenum@out{]}%
1607   \fi%
1608 }%
1609 %
1610 %
1611 %

```

`\flag@start@RTL` With  $X_{\text{L}}^{\text{L}}\text{T}_{\text{E}}\text{X}$ , 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>30</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

<sup>30</sup>This problem is caused by the way  $X_{\text{L}}^{\text{L}}\text{T}_{\text{E}}\text{X}$  manages right-to-left typesetting. David Carlisle explains it on <http://tex.stackexchange.com/a/333373/7712> and provides a potential solution, using `\vadjust`. However in some cases this adds spurious vertical spaces in `reledmac`. That is why we are using the solution explained below.

content of the first argument, and we call `\flag@end` at the point we normally call `\flag@start`, at the end of the content of the first argument.

- If the first argument of `\edtext` is typeset on only two lines, we use the normal order.

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

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

```

1612 \newcommand{\flag@start@RTL}{%
1613   \edlabel{edtext:start:\csuse{thisedtext@the\@edtext@level}}%
1614   \IfStrEq{\xabslineref{edtext:start:\csuse{thisedtext@the\@edtext@level}}%
1615     {000}%
1616     {}%
1617     {%
1618       \ifnumequal%
1619         {\xabslineref{edtext:start:\csuse{thisedtext@the\@edtext@level}}}%
1620         {\xabslineref{edtext:end:\csuse{thisedtext@the\@edtext@level}}}%
1621         {\flag@end}%
1622         {\flag@start}%
1623       }%
1624     }%
1625 }%
1626 \newcommand{\flag@end@RTL}{%
1627   \edlabel{edtext:end:\csuse{thisedtext@the\@edtext@level}}%
1628   \IfStrEq{\xabslineref{edtext:start:\csuse{thisedtext@the\@edtext@level}}%
1629     {000}%
1630     {}%
1631     {%
1632       \ifnumequal%
1633         {\xabslineref{edtext:start:\csuse{thisedtext@the\@edtext@level}}}%
1634         {\xabslineref{edtext:end:\csuse{thisedtext@the\@edtext@level}}}%
1635         {\flag@start}%
1636         {\flag@end}%
1637       }%
1638     }%
1639   %

```

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

```

1640 \newcommand*{\flag@start@later}{%
1641   \ifledRcol%
1642     \write\linenum@outR{\string\@ref@later []}%
1643   \else%
1644     \write\linenum@out{\string\@ref@later []}%
1645   \fi%
1646 }%
1647 \newcommand{\flag@end@later}{%
1648   \ifledRcol%
1649     \write\linenum@outR[]%
1650   \else%

```

```

1651 \write\linenum@out{]}%
1652 \fi%
1653 }
1654 %

```

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

We tinker with `\lastskip` because a command of either sort really needs to be attached to the last word preceding the change, not the first word that follows the change. This is because sub-lineation will often turn on and off in mid-line—stage directions, for example, often are mixed with dialogue in that way—and when a line is mixed we want to label it using the system that was in effect at its start. But when sub-lineation begins at the very start of a line we have a problem, if we don't put in this code.

```

1655
1656
1657 \newcommand*{\startsub}{%
1658 \dimen0\lastskip%
1659 \ifdim\dimen0>0pt%
1660 \unskip%
1661 \fi%
1662 \ifledRcol%
1663 \write\linenum@outR{\string\sub@on}%
1664 \else%
1665 \write\linenum@out{\string\sub@on}%
1666 \fi%
1667 \ifdim\dimen0>0pt%
1668 \hskip\dimen0%
1669 \fi%
1670 }%
1671
1672 \def\endsub{%
1673 \dimen0\lastskip%
1674 \ifdim\dimen0>0pt%
1675 \unskip%
1676 \fi%
1677 \ifledRcol%
1678 \write\linenum@outR{\string\sub@off}%
1679 \else%
1680 \write\linenum@out{\string\sub@off}%
1681 \fi%
1682 \ifdim\dimen0>0pt%
1683 \hskip\dimen0%
1684 \fi%
1685 }%
1686
1687 %

```

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



```

1688 \newcommand*{\advanceline}[1]{\leavevmode%
1689   \ifledRcol \write\linenum@outR{\string\@adv[#1]}%
1690   \else      \write\linenum@out{\string\@adv[#1]}%
1691   \fi%
1692 }
1693 %

```

**\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.

```

1694
1695 \newcommand*{\setline}[1]{%
1696   \leavevmode%
1697   \ifnum#1<\z@
1698     \led@warn@BadSetline%
1699   \else%
1700     \ifledRcol%
1701       \write\linenum@outR{\string\@set[#1]}%
1702     \else%
1703       \write\linenum@out{\string\@set[#1]}%
1704     \fi%
1705   \fi%
1706 }%
1707
1708 %

```

**\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.

```

1709
1710 \newcommand*{\setlinenum}[1]{%
1711   \ifnum#1<\z@%
1712     \led@warn@BadSetlinenum%
1713   \else%
1714     \ifledRcol%
1715       \write\linenum@outR{\string\l@d@set[#1]}%
1716     \else%
1717       \write\linenum@out{\string\l@d@set[#1]}%
1718     \fi%
1719   \fi%
1720 }%
1721
1722 %

```

**\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.

```

1723
1724 \newcommand*{\startlock}{%
1725   \ifledRcol \write\linenum@outR{\string\lock@on}%
1726   \else      \write\linenum@out{\string\lock@on}%
1727   \fi}
1728 \def\endlock{}

```

```

1729 \ifledRcol \write\linenum@outR{\string\lock@off}%
1730 \else      \write\linenum@out{\string\lock@off}%
1731 \fi}
1732 %

```

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

```

\l@dskipnumbertrue
\l@dskipnumberfalse
\skipnumbering
1733 \newif\ifl@dskipnumber
1734 \newif\ifl@dskipversenumber%
1735 \newcommand*{\skipnumbering}{%
1736   \leavevmode%
1737   \ifledRcol%
1738     \ifinstanza%
1739       \write\linenum@outR{\string\n@num@stanza}%
1740     \else%
1741       \write\linenum@outR{\string\n@num}%
1742     \fi%
1743     \advanceline{-1}%
1744   \else%
1745     \ifinstanza%
1746       \write\linenum@out{\string\n@num@stanza}%
1747     \else%
1748       \write\linenum@out{\string\n@num}%
1749     \fi%
1750     \advanceline{-1}%
1751   \fi%
1752 }%
1753 %
1754 %

```

## 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. 154). 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.

```
1755 \list@create{\end@lemmas}
1756 %
```

`\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.

```
1757 \newcommand{\dummy@edtext}[2]{#1}
1758 %
```

`\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`.

```
1759 \newcommand{\dummy@edtext@showlemma}[2]{\showlemma{#1}}%
1760 %
```

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>31</sup> This is done because we want to pass into our notes the generic commands to change to roman or whatever, and not their expansions that will ask for a particular style at a specified size. The notes may well be in a smaller font, so the command should be expanded later, when the note’s environment is in effect.

A second sort to turn off includes a few of the accent macros. Most are not a problem: an accent that is expanded to an `\accent` command may be harder to read but it works just the same. The ones that cause problems are: those that use alignments— $\TeX$  seems to get confused about the difference between alignment parameters and macro parameters; those that use temporary control sequences; and those that look carefully at what the current font is.

(The `\copyright` macro defined in PLAIN  $\TeX$  has this sort of problem as well, but is not used enough to bother with. That macro, and any other that causes trouble, will get by all right if you put a `\protect` in front of it in your file.)

We also need to eliminate all `reledmac` macros like `\edlabel` and `\setline` that write things to auxiliary files: that writing should be done only once. And we make `\edtext` itself, if it appears within its own argument, do nothing but copy its first argument.

Finally, we execute `\morenoexpands`. The version of `\morenoexpands` defined here does nothing; but you may define a version of your own when you need to add more expansion suppressions as needed with your macros. That makes it possible to make such additions without needing to copy or modify the standard `reledmac` code. If you define your own `\morenoexpands`, you must be very careful about spaces: if the macro adds any spaces to the text when it runs, extra space will appear in the main text when `\edtext` is used.

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

(A related problem, not addressed by these two macros, is that of characters whose category code are changed by any of the macros used in the arguments to `\edtext`. Since the category codes are set when the arguments are scanned, macros that depend

<sup>31</sup>Since “control sequences equivalent to characters are not expandable”—*The TeXbook*, answer to Exercise 20.14.

on changing them will not work. We have most often encountered this with characters that are made “active” within text in some, but not all, of the languages used within the document. One way around the problem, if it takes this form, is to ensure that those characters are *always* active. Within languages that make no special use of them, their associated control sequences should simply return the proper character. A simpler solution is to avoid active characters, using Lua<sub>TEX</sub> or Xe<sub>TEX</sub>.)

```

1761 \newcommand*{\noexpands}{%
1762   \let\select@lemmafnt=0%
1763   \let\startsub=\relax \let\endsub=\relax
1764   \let\startlock=\relax \let\endlock=\relax
1765   \let\edlabel=\@gobble
1766   \let\setline=\@gobble \let\advanceline=\@gobble
1767   \let\sameword\sameword@inedtext%
1768   \let\edtext=dummy@edtext
1769   \let\edindex\dummy@edindex%
1770   \l@dtabnoexpands
1771   \l@noexpands@edgl%
1772   \let\linenumnotation=\@gobble%
1773   \morenoexpands}
1774 \let\morenoexpands=\relax
1775
1776 %

```

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

```

1777 \newcommand{\@tag}{}
1778 %

```

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

```

1779 \newcount\@edtext@level%
1780 \@edtext@level=0%
1781 %

```

`\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.

```

1782 \newif\if@edtext@secondarg@%
1783 %

```

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

```

1784 \newcounter{edtext}
1785 \renewcommand{\theedtext}{edtxt@the\c@edtext}%
1786 %

```

`\edtext@notes` The `\edtext@notes` is increased each time we have a footnote required in the second argument of `\edtext`, that is a `\Xfootnote` or a `\Xendfootnote`.

The `\edtext@notes@start` is let to `\edtext@notes` at the beginning of a `\edtext`

```

1787 \newcount\edtext@notes%
1788 \newcount\edtext@notes@start%
1789 %

```

`\if@noneed@notes` The `\if@noneed@notes` macro is (locally) set to true when we read a critical note for a series with `set` to true.

```

1790 \newif\if@noneed@notes%
1791 %

```

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

```

1792 \newcommand{\edtext}[2]{\leavevmode%
1793 %

```

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

```

1794 \ifnumberedpar%
1795 %

```

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

```

1796 \check@edtext@args{#1}{#2}%
1797 %

```

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

```

1798 \global\advance\@edtext@level by 1%
1799 %

```

We let `\edtext@notes@start` equal to `\edtext@notes` to check later the number of notes associated to the edtext.

```

1800 \global\edtext@notes@start=\edtext@notes%
1801 %

```

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 `X2TeX`.

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`.

```

1802 \stepcounter{edtext}%
1803 \csxdef{thisedtext@the\@edtext@level}{\theedtext}%
1804 %

```

By default, we do not use `\lemma`

```

1805 \global\@lemmacommand@false%
1806 %

```

```

1807 \begingroup%
1808 %

```

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`.

```

1809     \ifledRcol%
1810         \ifcsvoid{sw@list@edtextR@the\@edtext@level}%
1811             {\global\let\sw@inthisedtext\empty}%
1812             {\expandafter\gl@p\csname sw@list@edtextR@the\
@edtext@level\endcsname\to\sw@inthisedtext}%
1813         \else%
1814             \ifcsvoid{sw@list@edtext@the\@edtext@level}%
1815                 {\global\let\sw@inthisedtext\empty}%
1816                 {\expandafter\gl@p\csname sw@list@edtext@the\@edtext@level
\endcsname\to\sw@inthisedtext}%
1817         \fi%
1818 %

```

`\@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.

```

1819     \global\renewcommand{\@tag}{%
1820         \no@expands #1%
1821     }%
1822 %

```

`\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`.

```

1823     \set@line%
1824 %

```

`\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`.

```

1825     \ifledRcol \global\insert@countR \z@%
1826     \else      \global\insert@count \z@ \fi%
1827 %

```

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.

```

1828     \@edtext@secondarg@true%
1829     \ignorespaces #2\relax%
1830     \@edtext@secondarg@false%
1831 %

```

Check if we indeed insert footnotes or critical notes.

```

1832     \ifnum\edtext@notes=\edtext@notes@start%
1833         \unless\if@noneed@notes%

```

```

1834     \led@err@EdtextWithoutNote{#1}%
1835     \fi%
1836     \fi%
1837 %

```

With  $X_{\text{La}}\text{TeX}$ , 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.

```

1838     \if@RTL%
1839     \IfStrEq{\xabslineref{edtext:start:\csuse{thisedtext@the\
@edtext@level}}}%
1840     {000}%
1841     }%
1842     {%
1843     \ifnumequal%
1844     {\xabslineref{edtext:start:\csuse{thisedtext@the\
@edtext@level}}}%
1845     {\xabslineref{edtext:end:\csuse{thisedtext@the\@edtext@level
}}}%
1846     }%
1847     {\write@lemma}%
1848     }%
1849     \flag@start@RTL%
1850     \IfStrEq{\xabslineref{edtext:start:\csuse{thisedtext@the\
@edtext@level}}}%
1851     {000}%
1852     }%
1853     {%
1854     \ifnumequal%
1855     {\xabslineref{edtext:start:\csuse{thisedtext@the\
@edtext@level}}}%
1856     {\xabslineref{edtext:end:\csuse{thisedtext@the\@edtext@level
}}}%
1857     {\write@lemma}%
1858     }%
1859     }%
1860     \else%
1861     \flag@start%
1862     \write@lemma%
1863     \fi%
1864 %

```

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

```

1865     \endgroup%
1866     \ifdef{\hypertarget}%
1867     {%
1868     \Hy@raisedlink@left{\hypertarget{\csuse{thisedtext@the\
@edtext@level}:start}}}%

```



```

1869     \showlemma{#1}%
1870     \Hy@raisedlink{\hypertarget{\csuse{thisedtext@the\@edtext@level}:
end}{}}}%
1871     }%
1872     {%
1873     \showlemma{#1}%
1874     }%
1875 %

```

If using `hyperref` package, we put the final `hypertarget` for `\applabel`.

```

1876     \ifdef{\hypertarget}%
1877     {%
1878     \def\do##1{\Hy@raisedlink{\hypertarget{##1:end}{}}}%
1879     \ifcsdef{applabel@thisedtext@the\@edtext@level}{%
1880     \dolistcsloop{applabel@thisedtext@the\@edtext@level}%
1881     \global\cslet{applabel@thisedtext@the\@edtext@level}{\
empty}%
1882     }{}%
1883     }%
1884     {}%
1885 %

```

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.

```

1886     \ifx\end@lemmas\empty \else%
1887     \glp\end@lemmas\to\x@lemma%
1888     \x@lemma%
1889     \global\let\x@lemma=\relax%
1890     \fi%
1891     \ifRTL%
1892     \flag@end@RTL%
1893     \else%
1894     \flag@end%
1895     \fi%
1896 %

```

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`.

```

1897     \global\advance\@edtext@level by -1%
1898     \global\@lemmacommand@false%
1899 %

```

We also reset `\@beforeinsertofthisedtext`

```

1900     \global\let\@beforeinsertofthisedtext\relax%
1901 %

```

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

```

1902 \else%
1903 \showlemma{#1} (\textbf{\textsc{Edtext outside numbered paragraph}})\
led@err@edtextoutsidestart%
1904 \fi%
1905 }%
1906
1907
1908 %

```

`\@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.

```

1909 \newcommand{\@check@edtext@args}[2]{%
1910 \begingroup%
1911 \noexpandarg%
1912 \IfSubStr{#1}{\pstart}{\led@err@PstartInEdtext{first}}{}%
1913 \IfSubStr{#1}{\pend}{\led@err@PendInEdtext{first}}{}%
1914 \IfSubStr{#2}{\pstart}{\led@err@PstartInEdtext{second}}{}%
1915 \IfSubStr{#2}{\pend}{\led@err@PendInEdtext{second}}{}%
1916 \endgroup%
1917 }%
1918 %
1919 %

```

`\@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`.

```

1920 \let\@beforeinsertofthisedtext\relax
1921 %

```

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

```

1922 \newif\ifnumberline
1923 \numberlinetrue
1924 %

```

`\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.

```

1925 \newcommand*{\set@line}{%
1926 \ifledRcol
1927 \ifx\line@listR\empty
1928 \global\noteschanged@true

```

```

1929 \xdef\l@d@nums{000|000|000|000|000|000|000|\edfont@info}%
1930 \else
1931 \glp\line@listR\to\@tempb
1932 \xdef\l@d@nums{\@tempb|\edfont@info}%
1933 \global\let\@tempb=\undefined
1934 \fi
1935 \ifx\annot@listR\empty%
1936 \xdef\l@current@annotR{||}%
1937 \else%
1938 \glp\annot@listR\to\@tempb%
1939 \xdef\l@current@annotR{\@tempb}%
1940 \global\let\@tempb=\undefined%
1941 \fi%
1942 \else
1943 \ifx\line@list\empty
1944 \global\noteschanged>true
1945 \xdef\l@d@nums{000|000|000|000|000|000|000|\edfont@info}%
1946 \else
1947 \glp\line@list\to\@tempb
1948 \xdef\l@d@nums{\@tempb|\edfont@info}%
1949 \global\let\@tempb=\undefined
1950 \fi
1951 \ifx\annot@list\empty%
1952 \xdef\l@current@annot{||}%
1953 \else%
1954 \glp\annot@list\to\@tempb%
1955 \xdef\l@current@annot{\@tempb}%
1956 \global\let\@tempb=\undefined%
1957 \fi%
1958 \fi}
1959
1960 %

```

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

```

1961 \newcommand*\edfont@info{\f@encoding/\f@family/\f@series/\f@shape}
1962
1963 %

```

## 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. 135).

```

1964 \newcommand*\lemma}[1]{%
1965 \global\@lemmacommand>true%
1966 \global\renewcommand{\@tag}{%
1967 \no@expands #1%
1968 }%
1969 \ignorespaces%
1970 }%
1971 %

```



```

1997     \l@d@add{#1}%
1998     \fi
1999     \gdef\@tempb{#4}%
2000     \ifx\@tempb\empty\else
2001         \l@d@add{|\line@set#2\|#4\}%
2002     \fi}
2003 %

```

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

```

2004 \newcommand{\l@d@add}[1]{\xdef\l@d@nums{\l@d@nums#1}}
2005
2006 %

```

`\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.

```

2007 \newcommand*\lineannot}[1]{%
2008     \lineannot@set#1|}%
2009 }%
2010 \def\lineannot@set#1|#2|{%
2011     \expandafter\parse@annot#1|#2|%
2012     \IfStrEq{#1}{}%
2013         {\let\@tempa\annot@start}%
2014         {\def\@tempa{#1}}%
2015     \IfStrEq{#2}{}%
2016         {\let\@tempb\annot@start}%
2017         {\def\@tempb{#2}}%
2018     \xdef\l@current@annot{\@tempa|\@tempb}%
2019 }%
2020 %

```

## VI.4 Lemma disambiguation

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

The general mechanism is the following:

- **At the first run**, each `\sameword` command increments an `etoolbox` counter the name of which contains the argument of the `\sameword` commands.
- Then this counter associated with the argument of `\sameword` is stored with the `\@sw` command in the auxiliary file of the current `reledmac` section (the `.1`, `.2...` file).
- **When this auxiliary file is read at the second run**, different operations are achieved:
  1. Get the rank of each `\sameword` in a line (relative rank) from the rank of each `\sameword` in all the numbered section (absolute rank):

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

```

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

```

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

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

```

bar1 foo1 foo2 bar2 foo3 (A)(B) foo4 bar3 (C) foo5 (D) bar4 (E)

```

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

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

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

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

- At the second run, when a critical notes is called, we associate it to the next item of the list associated to its `\edtext` level. So, in the previous example:
  - Critical notes (A) and (B) are associated with  $\{1, 2, 2, 3\}$ .
  - Critical note (C) is associated with  $\{1, 2, 2, 3, 4, 3\}$ .

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

After theory, implementation. First, getting a sanitized form of the argument of `\sameword`

```

\get@sw@txt21 \newcommand{\get@sw@txt}[1]{%
2022   \begingroup%
2023   \swnoexpands%
2024   %.
```

Using case sensibility option.

```

2025   \ifsw@caseinsensitive%
2026     \def\@tmpa##1{\lowercase{##1}}%
2027   \else%
2028     \def\@tmpa##1{##1}%
2029   \fi%
2030   %
```

And now, define `\sw@txt`.

```

2031     \@tmpa{\protected@xdef\sw@txt{##1}}%
2032   \endgroup%
2033 }%
2034 %
```

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}{%
2036   \let\sameword\l@secondmandarg%Allow to have nested \sameword
2037   \let\emph\@firstofone%
2038   \let\textit\@firstofone%
2039   \let\textbf\@firstofone%
2040   \let\textsc\@firstofone%
2041   \let\framebox\@firstofone%
2042   \let\edtext\dummy@edtext%
2043   \RenewExpandableDocumentCommand{\edindex}{om}{}%
2044   \ifdefined\index%
2045     \RenewExpandableDocumentCommand{\index}{om}{}%
2046   \fi%
2047   \let\selectlanguage@gobble%
2048   \let\foreignlanguage@secondoftwo%
2049   \ifdefined\xpg@loaded%
2050     \renewcommand\do[1]{%
2051       \expandafter\RenewExpandableDocumentCommand\expandafter{\csname
text##1\endcsname}{om}{###2}%
2052     }%
2053     \expandafter\docsvlist\expandafter{\xpg@loaded}%
2054   \fi%
2055 }%
2056 %

```

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

```

2057 \newcommandx{\sameword}[2][1,usedefault]{%
2058   \ifxetex%
2059     \if@RTL%
2060       \led@err@samewordRTL%
2061     \fi%
2062   \fi%
2063   \leavevmode%
2064   \get@sw@txt{#2}%
2065 %

```

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

```

2066   \unless\ifledRcol%
2067     \csnumgdef{sw@\sw@txt}{\csuse{sw@\sw@txt}+\@ne}%
2068 %

```

Then, write its value to the numbered file.

```

2069     \protected@write\linenum@out{\string\@sw{\sw@txt}{\csuse{sw@\sw@txt}
}}{#1}}%
2070 %

```

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

```

2071   \else%
2072     \csnumgdef{sw@\sw@txt}{\csuse{sw@\sw@txt}+\@ne}%

```



```

2073 \protected@write\linenum@outR{}{\string\@sw{\sw@txt}{\csuse{sw@\sw@txt
}}{#1}}%
2074 \fi%
2075 %

```

And print the word.

```

2076 #2%
2077 }%
2078 %

```

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

```

\if@addsw79 \newif\if@addsw%
2080 %

```

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

```

2081 \newcommand{\@sw}[3]{%
2082 \get@sw@txt{#1}%
2083 \unless\ifledRcol%
2084 %

```

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

```

2085 \csxdef{sw@\sw@txt @\the\absline@num @\the\section@num}{#2}%
2086 %

```

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

```

2087 \numdef{\prev@line}{\the\absline@num-1}%
2088 \ifcsundef{sw@\sw@txt @\prev@line @\the\section@num}{%
2089 \csnumgdef{sw@\sw@txt @\prev@line @\the\section@num}{#2-1}%
2090 }{}%
2091 %

```

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

```

2092 \numdef{\the@sw}{#2-\csuse{sw@\sw@txt @\prev@line @\the\section@num}}%
2093 %

```

And do the same thing for the right side.

```

2094 \else%
2095 \csxdef{sw@\sw@txt @\the\absline@numR @\the\section@numR @R}{#2}%
2096 \numdef{\prev@line}{\the\absline@numR-1}%
2097 \ifcsundef{sw@\sw@txt @\prev@line @\the\section@numR @R}{%
2098 \csnumgdef{sw@\sw@txt @\prev@line @\the\section@numR @R}{#2-1}%
2099 }{}%
2100 \numdef{\the@sw}{#2-\csuse{sw@\sw@txt @\prev@line @\the\section@numR @R
}}%
2101 \fi%
2102 %

```

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

```

2103 \@tempcnta=\@edtext@level
2104 \@whilenum{\@tempcnta>0}\do{%
2105   \ifcsdef{sw@list@edtext@tmp@the\@tempcnta}%
2106     {%
2107       \@addswfalse%
2108       \notbool{lemmacommand@the\@tempcnta}%
2109       {\@addswtrue}%
2110       {\IfStrEq{#3}{inlemma}%
2111        {\@addswtrue}%
2112        {%
2113          \def\do##1{%
2114            \ifnumequal{##1}{the\@tempcnta}%
2115              {\@addswtrue\listbreak}%
2116              {%
2117                }%
2118                \docsvlist{#3}%
2119              }%
2120            }%
2121            \if@addsw%
2122              \letcs{\@tmp}{sw@list@edtext@tmp@the\@tempcnta}%
2123              \ifledRcol%
2124                \xright@appenditem{\the@sw}{\the\absline@numR}}\to\@tmp%
2125              \else%
2126                \xright@appenditem{\the@sw}{\the\absline@num}}\to\@tmp%
2127              \fi%
2128              \cslet{sw@list@edtext@tmp@the\@tempcnta}{\@tmp}%
2129            \fi%
2130          }%
2131        }%
2132        \advance\@tempcnta by -1%
2133      }%
2134    }%
2135  %

```

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

```

2136 \newcommandx{\sameword@inedtext}[2][1,usedefault]{%
2137   \get@sw@txt{#2}%
2138   \unless\ifledRcol%
2139 %

```

Just a precaution.

```

2140   \ifx\sw@list@inedtext\empty%
2141     \def\the@sw{999}%
2142     \def\this@absline{-99}%
2143   \else%
2144 %

```

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

```

2145   \gl@p\sw@list@inedtext\to\@tmp%
2146   \edef\the@sw{\expandafter\@firstoftwo\@tmp}%
2147   \edef\this@absline{\expandafter\@secondoftwo\@tmp}%
2148   \fi%

```

```

2149 %
    First, calculate the number of occurrences of the word in the current line
2150 \ifcsdef{sw@sw@txt @\this@absline @\the\section@num}{%
2151   \numdef{\prev@line}{\this@absline-1}%
2152   \numdef{\sw@atthisline}{\csuse{sw@sw@txt @\this@absline @\the\
section@num}-\csuse{sw@sw@txt @\prev@line @\the\section@num}}%
2153   }%
2154   {\numdef{\sw@atthisline}{0}}%
2155 %
    Finally, print the rank, but only if there is more than one occurrence of the word in the
    current line.
2156 \ifnumgreater{\sw@atthisline}{1}%
2157   {\showwordrank{#2}{\the@sw}}%
2158   {#2}%
2159 %
    And the same for right side.
2160 \else%
2161   \ifx\sw@list@inedtext\empty%
2162     \def\the@sw{999}%
2163     \def\this@absline{-99}%
2164   \else%
2165     \glp\sw@list@inedtext\to\@tmp%
2166     \edef\the@sw{\expandafter\@firstoftwo\@tmp}%
2167     \edef\this@absline{\expandafter\@secondoftwo\@tmp}%
2168   \fi%
2169   \ifcsdef{sw@sw@txt @\this@absline @\the\section@numR @R}{%
2170     \numdef{\prev@line}{\this@absline-1}%
2171     \numdef{\sw@atthisline}{\csuse{sw@sw@txt @\this@absline @\the\
section@numR @R}-\csuse{sw@sw@txt @\prev@line @\the\section@numR @R}}%
2172     }%
2173     {\numdef{\sw@atthisline}{0}}%
2174     \ifnumgreater{\sw@atthisline}{1}%
2175       {\showwordrank{#2}{\the@sw}}%
2176       {#2}%
2177   \fi%
2178 }%
2179 %

```

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

```

2180 \newcommand{\showwordrank}[2]{%
2181   #1\textsuperscript{#2}%
2182 }%
2183 %

```

## 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@`  
`\numberedpar@true` When we first form the paragraph, it goes into a box register, `\raw@text`, instead  
`\numberedpar@false` of onto the current vertical list. The `\ifnumberedpar@` flag will be true while a para-  
`\num@lines` graph is being processed in that way. `\num@lines` will store the number of lines in the  
`\one@line` paragraph when it is complete. When we chop it up into lines, each line in turn goes  
`\par@line` into the `\one@line` register, and `\par@line` will be the number of that line within the  
paragraph.

```
2184 \newbox\raw@text
2185 \newif\ifnumberedpar@
2186 \newcount\num@lines
2187 \newbox\one@line
2188 \newcount\par@line
2189 %
```

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

```
2190 \newcount\pstarts@typeset@L%
2191 \newcount\pstarts@read@L%
2192 %
```

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

```
\thepstart
2193 \if@every@pstart@star@
2194 \newcommand{\AtStartEveryPstart}[1]{%
2195   \ifstrempy{#1}%
2196   {\gdef\@at@start@every@pstart{}}%
2197   {\gdef\@at@start@every@pstart{#1}}%
2198 }%
2199 \def\@at@start@every@pstart{}
```

```

2200
2201 \newif\ifat@every@pstart@star@%
2202 \newcommand{\AtEveryPstart}[1]{%
2203   \ifstrempy{#1}%
2204   {\gdef\at@every@pstart{}}%
2205   {\gdef\at@every@pstart{\noindent#1}}%
2206   \global\at@every@pstart@star@false%
2207 }%
2208 \WithSuffix\newcommand\AtEveryPstart*[1]{%
2209   \ifstrempy{#1}%
2210   {\gdef\at@every@pstart{}}%
2211   {\gdef\at@every@pstart{#1}}%
2212   \global\at@every@pstart@star@true%
2213 }%
2214 \def\at@every@pstart{}%
2215
2216 \newcounter{pstart}
2217 \renewcommand{\thepstart}{\bfseries\@arabic\c@pstart}. }
2218 \newif\ifnumberpstart
2219 \numberpstartfalse
2220 \newif\iflabelpstart
2221 \labelpstartfalse
2222 \newcommandx*\{pstart}[2][1,2,usedefault]{%
2223   \normal@pars%
2224   \ifboolexpr{%
2225     test {\ifstrempy{#1}}%
2226     and test {\ifstrempy{#2}}%
2227   }%
2228   {\at@every@pstart}%
2229   {%
2230     \ifstrempy{#1}{\noindent#1}%
2231     \ifstrempy{#2}{#2}%
2232   }%
2233   \ifautopar%
2234     \autopar%
2235   \fi%
2236   \ifluatex%
2237     \edef\l@luatextextdir@L{\the\textdir}%
2238   \fi%
2239   \@nobreaktrue%
2240   \ifnumbering \else%
2241     \led@err@PstartNotNumbered%
2242     \beginnumbering%
2243   \fi%
2244   \ifnumberedpar@%
2245     \led@err@PstartInPstart%
2246     \pend%
2247   \fi%
2248   \list@clear{\inserts@list}%
2249   \global\let\next@insert=\empty%
2250   \begingroup\normal@pars%
2251   \global\advance \l@dnumpstartsL\@ne
2252   \global\advance \pstarts@typeset@L\@ne%
2253   \global\advance \pstarts@read@L\@ne%

```

```

2254 \global\setbox\raw@text=\vbox\bgroup%
2255 \if@nobreak%
2256 \if@afterindent\else%
2257 \unless\ifinstanza%
2258 \noindent%
2259 \fi%
2260 \global\@afterindenttrue%
2261 \fi%
2262 \fi%
2263 \ifboolexpr{%
2264 bool{autopar}%
2265 and bool{by@autopar}}%
2266 }%
2267 {}%
2268 {%
2269 \ifnumberpstart%
2270 \ifinstanza\else%
2271 \ifsidepstartnum\else%
2272 \thepstart%
2273 \fi%
2274 \fi%
2275 \fi%
2276 }%
2277 \numberedpar@true%
2278 \iflabelpstart%
2279 \protected@edef\@currentlabel{\p@pstart\thepstart}%
2280 \fi%
2281 \l@dzeropenalties%
2282 \@at@start@every@pstart%
2283 \global\by@autoparfalse%
2284 \ignorespaces%because not automatically ignored if an optional argument
is used (classical TeX behavior for space after commands)
2285 }
2286 %

```

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

```

2287 \newcommand*{\pend}[2][1,2,usedefault]{\ifnumbering \else%
2288 \led@err@PendNotNumbered%
2289 \fi%
2290 \global\l@dskipversenumberfalse%
2291 \ifnumberedpar@ \else%
2292 \led@err@PendNoPstart%
2293 \fi%
2294 %

```

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.

```

2295 \l@dzeropenalties%
2296 \@at@end@every@pend%

```

```

2297 \endgraf\global\num@lines=\prevgraf\egroup%
2298 \global\par@line=0%
2299 \get@firsthalf@antilabe@wd%
2300 %

```

Output the lines.

```

2301 \loop\ifvbox\raw@text%
2302 \do@line%
2303 \repeat%
2304 %

```

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

```

2305 \flush@notes%
2306 \endgroup%
2307 \ignorespaces%
2308 %

```

Increase `pstart` counter.

```

2309 \ifnumberpstart%
2310 \global\pstartnumtrue%
2311 \fi%
2312 \addtocounter{pstart}{1}%
2313 \ifcontinuousnumberingwithcolumns%
2314 \addtocounter{pstartL}{1}%
2315 \addtocounter{pstartR}{1}%
2316 \fi%
2317 %

```

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

```

2318 \normal@pars%
2319 \ifboolexpr{%
2320 test {\ifstrempy{#1}}%
2321 and test {\ifstrempy{#2}}%
2322 }%
2323 {\at@every@pend}%
2324 {%
2325 \ifstrempy{#1}{\noindent#1}%
2326 \ifstrempy{#2}{#2}%
2327 }%
2328 %

```

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.

```

2329 \@nobreakfalse%
2330 \ifautopar%
2331 \autopar%
2332 \fi%
2333 %

```

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

```

2334 \ifbypstart@%
2335 \begingroup%
2336 \unless\ifinstanza%
2337 \let\leavevmode\relax%
2338 \setlinenum{0}%
2339 \fi%
2340 \endgroup%
2341 \resetprevline@%
2342 \fi%
2343 }
2344 %

```

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.

```

\AtEveryPend45
\at@every@pend46
\ifat@every@pend@star@47 \newif\ifat@every@pend@star@%
2348 \newcommand{\AtEveryPend}[1]{%
2349 \ifstrempy{#1}%
2350 {\gdef\at@every@pend{}}%
2351 {\gdef\at@every@pend{\noindent#1}}%
2352 \global\at@every@pend@star@false%
2353 }%
2354 \WithSuffix\newcommand\AtEveryPend*[1]{%
2355 \ifstrempy{#1}%
2356 {\gdef\at@every@pend{}}%
2357 {\gdef\at@every@pend{#1}}%
2358 \global\at@every@pend@star@true%
2359 }%
2360 \xdef\at@every@pend{}%
2361
2362 %

```

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

```

2363 \newcommand{\AtEndEveryPend}[1]{%
2364 \ifstrempy{#1}%
2365 {\xdef\@at@end@every@pend{}}%
2366 {\gdef\@at@end@every@pend{#1}}%
2367 }%
2368 \def\@at@end@every@pend{}%
2369 %

```

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

```

2370 \newcommand*\l@dzzeropenalties{%
2371 \brokenpenalty \z@ \clubpenalty \z@
2372 \displaywidowpenalty \z@ \interlinepenalty \z@ \predisplaypenalty \z@
2373 \postdisplaypenalty \z@ \widowpenalty \z@}
2374
2375 %

```



`\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.

```

2376 \newif\ifautopar
2377 \newif\ifby@autopar%
2378 \newcommand*{\autopar}{
2379   \ifledRcol
2380     \ifnumberingR \else
2381     \led@err@AutoparNotNumbered
2382     \beginnumberingR
2383     \fi
2384   \else
2385     \ifnumbering \else
2386     \led@err@AutoparNotNumbered
2387     \beginnumbering
2388     \fi
2389   \fi
2390   \autopartrue
2391   \everypar{\setbox0=\lastbox
2392             \endgraf \vskip-\parskip
2393             \global\by@autopartrue%
2394             \pstart \noindent \kern\wd0%
2395             \ifnumberpstart%
2396             \ifinstanza\else%
2397             \thepstart%
2398             \fi%
2399             \fi%
2400             \let\par=\pend}%

```

```
2401 \ignorespaces}
2402 %
```

`\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.

```
2403 \newcommand*{\normal@pars}{\everypar{} \let\par\endgraf}
2404
2405 %
```

`\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.

```
2406 \newif\ifautopar@pause
2407 %
```

## 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.

```
2408 \newcommand*{\l@dunhbox@line}[1]{\unhbox #1}
2409 \newcommand*{\do@line}{%
2410   {\vbadness=10000
2411     \splittopskip=\z@
2412     \do@linehook
2413   \l@emptyd@ta
2414     \global\setbox\one@line=\vsplit\raw@text to\baselineskip}%
2415   \unvbox\one@line \global\setbox\one@line=\lastbox
2416   \getline@num
2417   \IfStrEq{\led@pb@setting}{before}{\led@check@pb\led@check@nopb}{ }
2418   \ifnum\@lock>\@ne
2419     \inserthangingsymboltrue
2420   \else
2421     \inserthangingsymbolfalse
2422   \fi
2423   \check@pb@in@verse
2424   \ifl@dhidenumber%
2425     \global\l@dhidenumberfalse%
2426     \f@x@l@cks%
2427   \else%
2428     \affixline@num%
2429   \fi%
2430 %
```

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

```

2431 \xifinlist{\the\pstarts@typeset@L}{\eled@sections@}%
2432   {\print@eledsection}%
2433   {\print@line}%
2434 \IfStrEq{\led@pb@setting}{after}{\led@check@pb\led@check@nopb}{-}
2435 }%
2436 %

```

### VII.2.2 Process for “normal” line

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

```

2437 \def\print@line{
2438 %

```

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

```

2439   \affixpstart@num%
2440 %

```

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

```

2441   \hb@xt@ \linewidth{
2442 %

```

User hooks.

```

2443     \do@insidelinehook%
2444     \csuse{insidethis@the\absline@num @the\section@num}%
2445     \global\csundef{insidethis@the\absline@num @the\section@num}%
2446 %

```

Left line number

```

2447     \l@dld@ta%
2448 %

```

Prepare text to be inserted before notes.

```

2449     \if@firstlineofpage%
2450     \set@Xtxtbeforenotes%
2451     \set@txtbeforenotesX%
2452     \global\@firstlineofpagefalse%
2453     \fi%
2454 %

```

Insert footnotes made of manuscripts data and critical footnotes.

```

2455     \ifdefstring{\ms@data@position}{msdata-regular}{%
2456     \insert@msdata%
2457     \add@inserts%
2458     \add@Xgroupbyline%
2459     }{%
2460     \add@inserts%
2461     \add@Xgroupbyline%
2462     \insert@msdata%
2463     }%
2464 %

```

Insert marginal notes.

```

2465     \affixside@note%
2466 %

Print left notes.

2467     \l@dlsn@te
2468 %

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

2469     {\ledllfill\hb@xt@ \wd\one@line{%
2470       \new@line%
2471 %

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

2472     \continuousnumberingwithcolumns@sync@linenumber@singletext%
2473 %

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

2474     \ifluatex%
2475       \textdir\l@luatextextdir@L%
2476     \fi%
2477 %

Insert, if needed, the hanging symbol.

2478     \inserthangingsymbol%
2479 %

And so, print the line.

2480     \l@dunhbox@line{\one@line}}%
2481 %

Right line number

2482     \ledrlfill\l@drd@ta%
2483 %

Print right notes.

2484     \l@drsn@te%
2485     }}%
2486 %

And reinsert penalties (for page breaking)...

2487     \add@penalties%
2488     }
2489 %

```

### 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.

```

2490 \def\print@eledsection{%
2491   \disable@edindex%
2492   \disable@edlabel%
2493   \if@firstlineofpage%
2494     \set@Xtxtbeforenotes%
2495     \set@txtbeforenotesX%
2496     \global\@firstlineofpagefalse%
2497   \fi%
2498   \ifdefstring{\ms@data@position}{msdata-regular}{%
2499     \insert@msdata%
2500     \add@inserts%
2501     \add@Xgroupbyline%
2502   }{%
2503     \add@inserts%
2504     \add@Xgroupbyline%
2505     \insert@msdata%
2506   }%
2507   \affixside@note%
2508   \numdef{\temp@}{\pstarts@typeset@L-1}%
2509   \xifinlist{\temp@}{\eled@sections@@}{\@nobreaktrue}{\@nobreakfalse}%
2510   \@eled@sectioningtrue%
2511   \csuse{eled@sectioning@the\pstarts@typeset@L}%
2512   \@eled@sectioningfalse%
2513   \global\csundef{eled@sectioning@the\pstarts@typeset@L}%
2514   \if@RTL%
2515     \hspace{-3\paperwidth}%
2516     {\hbox{\l@dunhbox@line{\one@line}} \new@line}%
2517   \else%
2518     \hspace{3\paperwidth}%
2519     {\new@line \hbox{\l@dunhbox@line{\one@line}}}%
2520   \fi%
2521   \vskip-\baselineskip%
2522   \continuousnumberingwithcolumns@sync@linenumber@singletext%
2523   \restore@edindex%
2524   \restore@edlabel%
2525 }
2526 %

```

### VII.2.4 Hooks

`\do@linehook` `\do@insidelinehook` 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.

```

2527 \newcommand*{\do@linehook}{}
2528 \newcommand*{\do@insidelinehook}{}
2529 %

```

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

```

2530 \newcommand*{\dolinehook}[1]{\gdef\do@linehook{#1}}%
2531 \newcommand*{\doinsidelinehook}[1]{\gdef\do@insidelinehook{#1}}%
2532

```

2533 %

`\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.

```
2534 \newcommand{\doinsidethislinehook}[1]{%
2535   \leavevmode%In case it begins with a \pstart, ensure the \@insidethisline
is written after \@nl
2536   \ifledRcol%
2537     \write\linenum@outR{\string\@insidethisline[\unexpanded{#1}]}%
2538   \else%
2539     \write\linenum@out{\string\@insidethisline[\unexpanded{#1}]}%
2540   \fi%
2541 }%
2542 %
```

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

```
2543 \newcommand{\@insidethisline}[1]{%
2544   \ifledRcol%
2545     \csgappto{insidethisR@the\absline@numR @\the\section@numR}{#1}%
2546   \else%
2547     \csgappto{insidethis@the\absline@num @\the\section@num}{#1}%
2548   \fi%
2549 }%
2550 %
```

### 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
2551 \newcommand*{\l@emptyd@ta}{%
2552   \gdef\l@dld@ta{}%
2553   \gdef\l@drd@ta{}%
2554   \gdef\l@dcsnotetext@l{}%
2555   \gdef\l@dcsnotetext@r{}%
2556   \gdef\l@dcsnotetext{}%
2557
2558 %
```

`\l@dlsn@te` `\l@drsn@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.

```
2559 \newcommand{\l@dlsn@te}{%
2560   \ifboolexpr{%
2561     bool {l@dprintingcolumns}%
2562     and bool {ledRcol@}%
2563   }{% If we are on a right column
2564     \@tempdima=\@morespace@leftnote@rightcolumn%
2565   }{%
2566     \@tempdima=\z@%
```

```

2567 }%
2568 \hb@xt@ \z@{\hss\box\l@dlp@rbox\kern\ledlsnotesep\hskip\@tempdima}%
2569 }%
2570 \newcommand{\l@drsn@te}{%
2571 \ifboolexpr{%
2572     bool {l@dprintingcolumns}%
2573     and not bool {ledRcol@}%
2574 }{% If we are on a left column
2575     \@tempdima=\@morespace@rightnote@leftcolumn%
2576 }{%
2577     \@tempdima=\z@%
2578 }%
2579 \hb@xt@ \z@{\hskip\@tempdima\kern\ledrsnotesep\box\l@drp@rbox\hss}%
2580 }%
2581
2582 %

```

`\ledllfill` These macros are called at the left (`\ledllfill`) and the right (`\ledllfill`) of each numbered line.

```

2583 \newcommand*\ledllfill{\unless\if@RTL\hfil\fi}%
2584 \newcommand*\ledrlfill{\if@RTL\hfil\fi}%
2585
2586 %

```

## 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.

```

2587 \newcommand*\getline@num{%
2588     \global\advance\absline@num \@ne%
2589     \do@actions
2590     \do@ballast
2591     \ifnumberline
2592         \ifsublines@
2593             \ifnum\sub@lock<\tw@
2594                 \global\advance\subline@num \@ne
2595             \fi
2596         \else
2597             \ifnum\@lock<\tw@
2598                 \global\advance\line@num \@ne
2599                 \global\subline@num \z@
2600             \fi
2601         \fi
2602     \fi
2603 }
2604 %

```

`\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. 173](#)).

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

`\c@ballast`

```
2605 \newcount\ballast@count
2606 \newcounter{ballast}
2607 \setcounter{ballast}{0}
2608 %
```

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.

```
2609 \newcommand*{\do@ballast}{\global\ballast@count \z@
2610 \begingroup
2611 \advance\absline@num \@ne
2612 \ifnum\next@actionline=\absline@num
2613 \ifnum\next@action>-1001\relax
2614 \global\advance\ballast@count by -\c@ballast
2615 \fi
2616 \fi
2617 \endgroup}
2618 %
```

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

`\do@actions@next`

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

```
2619 \newcommand*{\do@actions}{%
2620 \global\let\do@actions@next=\relax
2621 \ifnum\absline@num<\next@actionline\else
2622 %
```

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.

```
2623 \ifnum\next@action>-1001
2624 \global\page@num=\next@action
2625 \ifresumenummering@start%
2626 \setbox0=\hbox{}%Required to get the correct page number, when the
\resumenummering is just after a \newpage
2627 \ifnum\pausenummering@page@num<\page@num%
2628 \global\resumenummering@startfalse%
2629 \fi%
2630 \fi%
2631 \ifboolexpr{%
2632 bool{resumenummering@start}%
2633 and test {\ifdimgreater{\pagedepth}{\z@}}%
2634 }%
2635 {}%
```



```

2636     {\global\@firstlineofpagetrue}%
2637     \ifcsdef{reset@line\the\absline@num @\the\section@num}%
2638     {%
2639         \global\line@num=\z@%
2640         \global\subline@num=\z@%
2641         \resetprevline@%
2642     }%
2643     {}%
2644     \global\resumenumering@startfalse%
2645     \add@msdata@firstlineofpage%
2646 %

```

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`.)

```

2647     \else
2648         \ifnum\next@action<-4999
2649             \@l@dttempcnta=-\next@action
2650             \advance\@l@dttempcnta by -5001
2651             \ifsublines@
2652                 \global\subline@num=\@l@dttempcnta
2653             \else
2654                 \global\line@num=\@l@dttempcnta
2655             \fi
2656 %

```

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

```

2657     \else
2658         \@l@dttempcnta=-\next@action
2659         \advance\@l@dttempcnta by -1000
2660         \do@actions@fixedcode
2661     \fi
2662 \fi
2663 %

```

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

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

```

2664     \ifx\actionlines@list\empty
2665         \gdef\next@actionline{1000000}%
2666     \else
2667         \gl@p\actionlines@list\to\next@actionline
2668         \gl@p\actions@list\to\next@action
2669         \global\let\do@actions@next=\do@actions
2670     \fi
2671 \fi
2672 %

```

Make the recursive call, if necessary.

```

2673 \do@actions@next}
2674
2675 %

```

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

```

2676 \newcommand*{\do@actions@fixedcode}{%
2677   \ifcase\@l@dttempcnta
2678   \or%           % 1001 = starting sublineation
2679     \global\sublines@true
2680   \or%           % 1002 = ending sublineation
2681     \global\sublines@false
2682   \or%           % 1003 = starting locking number
2683     \global\@lock=\@ne
2684   \or%           % 1004 = ending locking number
2685     \ifnum\@lock=\tw@
2686       \global\@lock=\thr@@
2687     \else
2688       \global\@lock=\z@
2689     \fi
2690   \or%           % 1005 = starting locking subnumber
2691     \global\sub@lock=\@ne
2692   \or%           % 1006 = ending locking subnumber
2693     \ifnum\sub@lock=\tw@
2694       \global\sub@lock=\thr@@
2695     \else
2696       \global\sub@lock=\z@
2697     \fi
2698   \or%           % 1007 = skipping numbering
2699     \l@dskipnumbertrue
2700   \or%           % 1008 = skipping numbering in stanza
2701     \l@dskipversenumbertrue%
2702   \or%           % 1009 = hiding number
2703     \l@dhidnumbertrue
2704   \or%           % 1010 = inserting msdata
2705     \add@msdata%
2706   \else
2707     \led@warn@BadAction
2708   \fi}
2709
2710
2711 %

```

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

`columns@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 the not the same.

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

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

```

2712 \newcommand{\continuousnumberingwithcolumns@sync@linenumber@singletext}{%
2713   \ifcontinuousnumberingwithcolumns%
2714     \unless\ifafterendnumberingR%
2715       \newlineR%
2716       \xappto\next@line@list@stuffR{%
2717         \unexpanded{\global\line@numR=}\the\line@num%
2718       }%
2719     \fi%
2720   \fi%
2721 }%
2722 %

```

## 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 commands, which only write commands to the numbered files, storing the annotation.

`\resetlinenumannotation`

```

2723 \newcommand{\linenumannotation}[1]{%
2724   \leavevmode%In case it begins with a \pstart, ensure the \@annot is
written after \@nl
2725   \ifledRcol%
2726     \write\linenum@outR{\string\@annot[#1]}%
2727   \else%
2728     \write\linenum@out{\string\@annot[#1]}%
2729   \fi%
2730 }%
2731
2732 \newcommand{\resetlinenumannotation}[0]{%
2733   \leavevmode%In case it begins with a \pstart, ensure the \@resetannot is
written after \@nl
2734   \ifledRcol%
2735     \write\linenum@outR{\string\@resetannot}%
2736   \else%
2737     \write\linenum@out{\string\@resetannot}%
2738   \fi%
2739 }%
2740
2741 %

```

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

```
2742 \def\Xlinenumannotationposition@side{after}%
2743 %
```

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

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

```
2744 \def\Xwraplinenumannotation@ref{\textsuperscript}%
2745 \def\Xwraplinenumannotation@side{\textsuperscript}%
2746 \newtoggle{Xnoidenticallinenumannotation@ref}%
2747 \newtoggle{Xnoidenticallinenumannotation@side}%
2748 %
```

`\iflinenumannotationotherside` The `\iflinenumannotationotherside` boolean allows to print the annotation on the side opposite the line number.

```
2749 \newif\iflinenumannotationotherside%
2750 %
```

`\@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.

```
2751 \newcommand{\@annot}[1]{%
2752   \store@annot@to@absline{#1}%
2753   \def\current@annot{#1}%
2754 }%
2755 \newcommand{\store@annot@to@absline}[1]{%
2756   \ifledRcol%
2757     \ifcsdef{annotR\the\absline@numR @\the\section@numR}{%
2758       \csgappto{annotR@\the\absline@numR @\the\section@numR}{\
@linenumannotationsep#1}%
2759     }{%
2760       \csgdef{annotR@\the\absline@numR @\the\section@numR}{#1}%
2761     }%
2762   \else%
2763     \ifcsdef{annot@\the\absline@num @\the\section@num}{%
2764       \csgappto{annot@\the\absline@num @\the\section@num}{\
@linenumannotationsep#1}%
2765     }{%
```

```

2766 \csgdef{annot@the\absline@num @the\section@num}{#1}%
2767 }%
2768 \fi%
2769 }%
2770 %

```

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

```

2771 \let\current@annot=\empty%
2772 %

```

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

The `\@resetannot` macro is written in the auxiliary files with `\resetlinenumannotation` macro. It resets the annotation.

```

2773 \newcommand{\reset@current@annot}{%
2774 \unless\ifnoresetlinenumannotation@
2775 \let\current@annot\empty%
2776 \fi
2777 }%
2778 \newcommand{\@resetannot}{%
2779 \let\current@annot\empty%
2780 }%
2781 %

```

`\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.

```

2782 \def\parse@annot#1|#2|{%
2783 \gdef\annot@start{#1}%
2784 \gdef\annot@end{#2}%
2785 }%
2786 %

```

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

```

\@linenumannotationsep
2787 \newcommand{\setlinenumannotationsep}[1]{\gdef\@linenumannotationsep{#1}}%
2788 \def\@linenumannotationsep{, }%
2789 %

```

## X Line number printing

`\affixline@num` `\affixline@num` just puts a left line number into `\l@dld@ta` or a right line number into `\l@drd@ta` as required.

To determine whether we need to affix a line number to this line, we compute the following:

$$\begin{aligned}
 n &= \text{int}((\text{linenum} - \text{firstlinenum}) / \text{linenumincrement}) \\
 m &= \text{firstlinenum} + (n \times \text{linenumincrement})
 \end{aligned}$$

(where *int* truncates a real number to an integer). *m* will be equal to *linenum* only if we are to paste a number on here. However, the formula breaks down for the first line to number (and any before that), so we check that case separately: if  $\backslash\text{line@num} \leq \backslash\text{firstlinenum}$ , we compare the two directly instead of making these calculations.

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

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

```
2790 \newcommand*{\affixline@num}{%
2791 %
```

No number is attached if  $\backslash\text{ifl@dskipnumber}$  is TRUE (and then it is set to its normal FALSE value). No number is attached if  $\backslash\text{ifnumberline}$  is FALSE (the normal value is TRUE).

```
2792 \ifledgroupnotesL\else
2793 \ifnumberline
2794 \ifl@dskipnumber
2795 \global\l@dskipnumberfalse
2796 \else
2797 \ifsublines@
2798 \l@tempcntb=\subline@num
2799 \ifnum\subline@num>\c@firstsublinenum
2800 \l@tempcnta=\subline@num
2801 \advance\l@tempcnta by-\c@firstsublinenum
2802 \divide\l@tempcnta by\c@sublinenumincrement
2803 \multiply\l@tempcnta by\c@sublinenumincrement
2804 \advance\l@tempcnta by\c@firstsublinenum
2805 \else
2806 \l@tempcnta=\c@firstsublinenum
2807 \fi
2808 %
```

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.

```
2809 \ch@cksub@l@ck
2810 %
```

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

```
2811 \else
2812 \l@tempcntb=\line@num
2813 %
```

Check on the  $\backslash\text{linenumberlist}$  If it is  $\backslash\text{empty}$  use the standard algorithm.

```
2814 \ifx\linenumberlist\empty
2815 \ifnum\line@num>\c@firstlinenum
2816 \l@tempcnta=\line@num
2817 \advance\l@tempcnta by-\c@firstlinenum
2818 \divide\l@tempcnta by\c@linenumincrement
2819 \multiply\l@tempcnta by\c@linenumincrement
2820 \advance\l@tempcnta by\c@firstlinenum
2821 \else
```

```

2822         \@l@dtmpcnta=\c@firstlinenum
2823         \fi
2824     \else
2825 %

```

The `\linenumberlist` was not `\empty`, so here is Wayne's numbering mechanism. This takes place in  $\TeX$ 's mouth.

```

2826         \@l@dtmpcnta=\line@num
2827         \edef\rem@inder{\,\linenumberlist,\number\line@num,}%
2828         \edef\sc@n@list{\def\noexpand\sc@n@list
2829             ###1,\number\@l@dtmpcnta,###2|{\def\noexpand\rem@inder
2830 {#####2}}}%
2831         \sc@n@list\expandafter\sc@n@list\rem@inder|%
2832         \ifx\rem@inder\empty%
2833             \advance\@l@dtmpcnta\@ne
2834         \fi
2835 %

```

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

```

2836         \ch@ck@l@ck
2837     \fi
2838 %

```

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

```

2839         \ifnum\@l@dtmpcnta=\@l@dtmpcntb
2840         \ifl@dskipversenumber\else
2841 %

```

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

For  $\mathbb{E}\TeX$  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
2842         \if@twocolumn
2843             \if@firstcolumn
2844                 \gdef\l@dld@ta{\llap{\leftlinenum}}%
2845                 \iflinenumannotationotherside%
2846                     \gdef\l@drd@ta{\rlap{\rightlinenumannotation}}%
2847                 \fi%
2848             \else
2849                 \gdef\l@drd@ta{\rlap{\rightlinenum}}%
2850                 \iflinenumannotationotherside%
2851                     \gdef\l@dld@ta{\llap{\leftlinenumannotation}}%
2852                 \fi%
2853         \fi

```

```

2854         \else
2855             \ifboolexpr{bool {l@dprintingcolumns} and test {\
ifnumgreater{\line@margin@columns}{m@ne}}}%
2856                 {\@l@tempcntb=\line@margin@columns}%
2857                 {\@l@tempcntb=\line@margin}%
2858             \ifnum\@l@tempcntb>\@ne
2859                 \advance\@l@tempcntb \page@num
2860             \fi
2861             \ifboolexpr{%
2862                 bool {l@dprintingcolumns}%
2863                 and (%
2864                     (test {\ifdefstring{\linenum@OnlyPages@ForColumns}{left
}}}%
2865                     and test {\ifnumodd{\page@num}}%
2866                     )%
2867                 or%
2868                     (test {\ifdefstring{\linenum@OnlyPages@ForColumns}{right
}}}%
2869                     and not test {\ifnumodd{\page@num}}%
2870                     )%
2871                 )%
2872             }%
2873             {}%
2874             {%
2875                 \ifodd\@l@tempcntb%
2876                     \gdef\l@drd@ta{\rlap{{\rightlinenum}}}%
2877                     \iflinenumannotationotherside%
2878                         \gdef\l@dld@ta{\llap{{\leftlinenumannotation}}}%
2879                     \fi%
2880                 \else%
2881                     \gdef\l@dld@ta{\llap{{\leftlinenum}}}%
2882                     \iflinenumannotationotherside%
2883                         \gdef\l@drd@ta{\rlap{{\rightlinenumannotation}}}%
2884                     \fi%
2885                 \fi%
2886             }%
2887         \fi
2888     \fi
2889 \fi
2890 %

```

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

```

2891     \f@x@l@cks
2892     \fi
2893     \fi
2894 \fi
2895 }
2896 %
2897 %

```

`\ch@cksub@l@ck` These macros handle line number locking for `\affixline@num`. `\ch@cksub@l@ck`  
`\ch@ck@l@ck` checks subline locking. If it fails, then we disable the line-number display by setting the  
`\f@x@l@cks`



counters to arbitrary but unequal values.

```

2898 \newcommand*{\ch@cksub@l@ck}{%
2899   \ifcase\sub@lock
2900     \or
2901       \ifnum\sublock@disp=\@ne
2902         \@l@dttempcntb=\z@ \@l@dttempcnta=\@ne
2903       \fi
2904     \or
2905       \ifnum\sublock@disp=\tw@ \else
2906         \@l@dttempcntb=\z@ \@l@dttempcnta=\@ne
2907       \fi
2908     \or
2909       \ifnum\sublock@disp=\z@
2910         \@l@dttempcntb=\z@ \@l@dttempcnta=\@ne
2911       \fi
2912   \fi}
2913 %

```

Similarly for line numbers.

```

2914 \newcommand*{\ch@ck@l@ck}{%
2915   \ifcase\@lock
2916     \or
2917       \ifnum\lock@disp=\@ne
2918         \@l@dttempcntb=\z@ \@l@dttempcnta=\@ne
2919       \fi
2920     \or
2921       \ifnum\lock@disp=\tw@ \else
2922         \@l@dttempcntb=\z@ \@l@dttempcnta=\@ne
2923       \fi
2924     \or
2925       \ifnum\lock@disp=\z@
2926         \@l@dttempcntb=\z@ \@l@dttempcnta=\@ne
2927       \fi
2928   \fi}
2929 %

```

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

```

2930 \newcommand*{\f@x@l@cks}{%
2931   \ifcase\@lock
2932     \or
2933       \global\@lock=\tw@
2934     \or \or
2935       \global\@lock=\z@
2936     \fi
2937   \ifcase\sub@lock
2938     \or
2939       \global\sub@lock=\tw@
2940     \or \or
2941       \global\sub@lock=\z@
2942   \fi}
2943 %
2944 %

```

## 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.

```

\leftpstartnum45
\rightpstartnum46 \newif\ifsidepstartnum
\ifsidepstartnum47 \newcommand*{\affixpstart@num}{%
2948   \ifsidepstartnum
2949     \if@twocolumn
2950       \if@firstcolumn
2951         \gdef\l@dld@ta{\llap{\leftpstartnum}}%
2952       \else
2953         \gdef\l@drd@ta{\rlap{\rightpstartnum}}%
2954       \fi
2955     \else
2956       \l@dttempcntb=\line@margin%
2957       \ifnum\l@dttempcntb>\@ne
2958         \advance\l@dttempcntb \page@num
2959       \fi
2960       \ifodd\l@dttempcntb
2961         \gdef\l@drd@ta{\rlap{\rightpstartnum}}%
2962       \else
2963         \gdef\l@dld@ta{\llap{\leftpstartnum}}%
2964       \fi
2965     \fi
2966   \fi
2967 }
2968 %
2969
2970
2971 \newif\ifpstartnum
2972 \pstartnumtrue
2973 \newcommand*{\leftpstartnum}{
2974   \ifpstartnum\thepstart
2975   \kern\linenumsep\fi
2976   \global\pstartnumfalse
2977 }
2978 \newcommand*{\rightpstartnum}{
2979   \ifpstartnum
2980   \kern\linenumsep
2981   \thepstart
2982   \fi
2983   \global\pstartnumfalse
2984 }

```

```
2985 %
```

## 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.

```
2986 \list@create{\inserts@list}
2987 %
```

`\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`.

```
2988 \newcommand*{\add@inserts}{%
2989   \global\let\add@inserts@next=\relax
2990 %
```

If `\inserts@list` is empty, there are not any more notes or insertions for this paragraph, and we need not waste our time.

```
2991   \ifx\inserts@list\empty \else
2992 %
```

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.

```
2993   \ifx\next@insert\empty
2994     \ifx\insertlines@list\empty
2995       \global\noteschanged@true
2996       \gdef\next@insert{100000}%
2997     \else
2998       \gl@p\insertlines@list\to\next@insert
2999     \fi
3000   \fi
3001 %
```

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.

```

3002 \ifnum\next@insert=\absline@num
3003 \glp\inserts@list\to\@insert
3004 \@insert
3005 \global\let\@insert=\undefined
3006 \global\let\next@insert=\empty
3007 \global\let\add@inserts@next=\add@inserts
3008 \fi
3009 \fi
3010 %

```

Make the recursive call, if necessary.

```

3011 \add@inserts@next}
3012
3013 %

```

`\add@Xgroupbyline` If you use `\Xgroupbyline`, the insertion of the critical footnotes are not made immediately in `\add@inserts`, but the content to be inserted is stored, in order to be inserted later in one block. This insertion in one block is made by `\add@Xgroupbyline`.

```

3014 \newcommand{\add@Xgroupbyline}{%
3015 \unless\ifnocritical@%
3016 \def\do##1{%Looping on the series
3017 \let\olddo\do%Save the old \do macro, that is this macro itself!
3018 \def\do####1{%Looping on the ##1@forinserting command
3019 \ifcsdef{##1@forinserting@####1}{%
3020 \X@beforeinsertion{##1}%
3021 \if@ledgroup%
3022 \global\setbox\@nameuse{mp##1footins}=\vbox%
3023 \else%
3024 \insert\csname ##1footins\endcsname%
3025 \fi%
3026 {%
3027 \Xsetparindent{##1}%
3028 \ifcsdef{Xhsize\csuse{series@display##1}@##1}%
3029 {\hsize \csuse{Xhsize\csuse{series@display##1}@##1}}%
3030 {}%
3031 \if@ledgroup%
3032 \unvbox\@nameuse{mp##1footins}%
3033 \fi%
3034 \X@atbegininsertion{##1}%
3035 \ifcsstring{series@display##1}{normal}%
3036 {%
3037 \Xledsetnormalparstuff{##1}%
3038 \rule\z@\splittopskip%
3039 }%
3040 {}%
3041 \csuse{##1@forinserting@####1}%
3042 \strut\par\allowbreak%
3043 }%
3044 \global\csundef{##1@forinserting@####1}%

```

```

3045     }%
3046     {}%
3047     }%
3048     \ifcsdef{##1@forinserting}{%
3049     \dolistcslloop{##1@forinserting}%
3050     }{}%
3051     \global\csundef{##1@forinserting}%
3052     \let\do\olddo%Restore old do
3053     }%
3054     \dolistloop{\@series}%
3055     \fi%
3056 }%
3057
3058
3059 %

```

## 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. 159). Finally, the penalty is checked to see that it does not go below  $-10000$ .

```

3060 \newcommand*{\add@penalties}{\@l@dttempcnta=\ballast@count
3061   \ifnum\num@lines>\@ne
3062     \global\advance\par@line \@ne
3063     \ifnum\par@line=\@ne
3064       \advance\@l@dttempcnta \clubpenalty
3065       \fi
3066       \@l@dttempcntb=\par@line \advance\@l@dttempcntb \@ne
3067       \ifnum\@l@dttempcntb=\num@lines
3068         \advance\@l@dttempcnta \widowpenalty
3069       \fi
3070       \ifnum\par@line<\num@lines
3071         \advance\@l@dttempcnta \interlinepenalty
3072       \fi
3073     \fi
3074     \ifnum\@l@dttempcnta=\z@
3075       \relax
3076     \else
3077       \ifnum\@l@dttempcnta>-10000
3078         \penalty\@l@dttempcnta
3079       \else
3080         \penalty -10000
3081       \fi
3082     \fi}
3083

```

```
3084 %
```

### XII.3 Printing leftover notes

`\flush@notes` The `\flush@notes` macro is called after the entire paragraph has been sliced up and sent on to the vertical list. If the number of notes to this paragraph has increased since the previous run of  $\TeX$ , then there can be leftover notes that have not yet been printed. An appropriate error message will be printed elsewhere; but it is best to go ahead and print these notes somewhere, even if it is not in quite the right place. What we do is dump them all out here, so that they should be printed on the same page as the last line of the paragraph. We can hope that is not too far from the proper location, to which they will move on the next run. For the first run, we do not flush the notes, as that means all the notes will be added at the end of numbered section, and so, very far of the expected position

```
3085 \newcommand*{\flush@notes}{%
3086   \iftoggle{notfirstrun@jobname.\extensionchars\the\section@num}{%
3087     \@xloop%
3088     \ifx\inserts@list\empty \else%
3089       \gl@p\inserts@list\to\@insert%
3090       \@insert%
3091       \global\let\@insert=\undefined%
3092     \repeat%
3093   }%
3094 }%
3095
3096
3097 %
```

`\@xloop` `\@xloop` is a variant of the PLAIN  $\TeX$  `\loop` macro, useful when it's hard to construct a positive test using the  $\TeX$  `\if` commands—as in `\flush@notes` above. One types `\@xloop ... \if ... \else ... \repeat`, and the action following `\else` is repeated as long as the `\if` test fails. (This macro will work wherever the PLAIN  $\TeX$  `\loop` is used, too, so we could just call it `\loop`; but it seems preferable not to change the definitions of any of the standard macros.)

This variant of `\loop` was introduced by Alois Kabelaš in *TUGboat* 8 (1987), pp. 184–5.

```
3098 \def\@xloop#1\repeat{%
3099   \def\body{#1\expandafter\body\fi}%
3100   \body}
3101
3102 %
```

### 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 `\Xtxtbeforenotes` will be inserted.

```
3103 \newcommand{\set@Xtxtbeforenotes}{%
3104   \unless\ifnocritical@%
```

```

3105 \def\do##1{%
3106   \nottoggle{Xtxtbeforenotesonlyonce@##1}{%
3107     \global\togglefalse{Xtxtbeforenotes@##1@typeset}%
3108   }{}%
3109 }%
3110 \dolistloop{\@series}%
3111 \fi%
3112 }%
3113 %

```

`\set@txtbeforenotesX` The `\set@txtbeforenotesX` does the same for the `\txtbeforenotesX`.

```

3114 \newcommand{\set@txtbeforenotesX}{%
3115   \unless\ifnofamiliar@%
3116   \def\do##1{%
3117     \nottoggle{txtbeforenotesonlyonceX@##1}{%
3118       \global\togglefalse{txtbeforenotesX@##1@typeset}%
3119     }{}%
3120   }%
3121   \dolistloop{\@series}%
3122   \fi%
3123 }%
3124 %

```

`\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.

```

3125 \newcommand{\insert@Xtxtbeforenotes}[1]{%
3126   \nottoggle{Xtxtbeforenotes@#1@typeset}{%
3127     \global\toggletrue{Xtxtbeforenotes@#1@typeset}%
3128     \ifcvoid{Xtxtbeforenotes@#1}{}{%
3129       \ifcstring{series@display#1}{paragraph}%
3130       {\noindent\csuse{Xtxtbeforenotes@#1}}%
3131       {\expandafter\insert\curname#1footins\endcurname%
3132         \bgroup%
3133           \noindent%
3134           \ifcdef{\csuse{series@display#1}@begin@insert}{%
3135             \csuse{\csuse{series@display#1}@begin@insert}{#1}}%
3136           }{}%
3137           \strut\csuse{Xnotefontsize@#1}%
3138           \csuse{Xtxtbeforenotes@#1}%
3139         \egroup%
3140       }%
3141     }%
3142   }%
3143 }{}%
3144 }%
3145
3146 \newcommand{\insert@txtbeforenotesX}[1]{%
3147   \nottoggle{txtbeforenotesX@#1@typeset}{%

```

```

3149 \global\toggletrue{txtbeforenotesX@#1@typeset}%
3150 \ifcsvoid{txtbeforenotesX@#1}{-}{%
3151   \ifcsstring{series@displayX#1}{paragraph}%
3152     {\noindent\csuse{txtbeforenotesX@#1}}%
3153     {\expandafter\insert\csname footins#1\endcsname%
3154       \bgroup%
3155         \noindent%
3156         \ifcsdef{\csuse{series@displayX#1}@begin@insert}{%
3157           \csuse{\csuse{series@displayX#1}@begin@insert}{#1}%
3158         }-}%
3159         \strut\csuse{notefontsizeX@#1}\csuse{txtbeforenotesX@#1}%
3160       \egroup%
3161     }%
3162   }%
3163 }%
3164 {}%
3165 }%
3166
3167
3168 %

```

## 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.

```

3169 \def\select@lemmafont#1|#2|#3|#4|#5|#6|#7|{\select@lemmafont#7|}
3170 \def\select@lemmafont#1/#2/#3/#4|%
3171   {\fontencoding{#1}\fontfamily{#2}\fontseries{#3}\fontshape{#4}%
3172   \selectfont}
3173
3174 %

```

### 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.



```

3175 \newcommand*{\footnoteoptions@}[3]{%
3176   \def\do##1{%
3177     \ifstrequal{##1}{L}{% On the left side
3178       \xright@appenditem{\noexpand\setkeys[mac]{#3footnoteoption}{\
unexpanded{##1}}}\to\inserts@list%
3179       \global\advance\insert@count \@ne% Increment the left insert
counter.
3180     }%
3181     {%
3182       \xright@appenditem{\noexpand\setkeys[mac]{#3footnoteoption}{\
unexpanded{##1}}}\to\inserts@listR%
3183       \global\advance\insert@countR \@ne% Increment the right insert
counter insert.
3184     }%
3185   }%
3186   \notblank{#2}{\docsvlist{#2}}}% Parsing all options
3187 }
3188 %

```

### XIII.3 Notes language

`\footnotelang@lua` `\footnotelang@lua` is called to remember the information about the direction of a lemma when Lua $\TeX$  is used.

```

3189 \newcommandx*{\footnotelang@lua}[1][1=L,usedefault]{%
3190   \ifstrequal{##1}{L}{%
3191     \xright@appenditem{\csxdef{footnote@luatextextdir}{\the\textdir}}\to\
inserts@list%Know the dir of lemma
3192     \global\advance\insert@count \@ne%
3193     \xright@appenditem{\csxdef{footnote@luatexpardir}{\the\pardir}}\to\
inserts@list%Know the dir of lemma
3194     \global\advance\insert@count \@ne%
3195   }%
3196   {%
3197     \xright@appenditem{\csxdef{footnote@luatextextdir}{\the\textdir}}\to\
inserts@listR%Know the dir of lemma
3198     \global\advance\insert@countR \@ne%
3199     \xright@appenditem{\csxdef{footnote@luatexpardir}{\the\pardir}}\to\
inserts@listR%Know the dir of lemma
3200     \global\advance\insert@countR \@ne%
3201   }%
3202 }
3203 %

```

`\footnotelang@poly` `\footnotelang@poly` is called to remember the information about the language of a lemma when polyglossia is used.

```

3204 \newcommandx*{\footnotelang@poly}[1][1=L,usedefault]{%
3205   \ifstrequal{##1}{L}{%
3206     \ifRTL%
3207       \xright@appenditem{\csxdef{footnote@dir}{@RTLtrue}}\to\
inserts@list%Know the language used in the lemma
3208       \global\advance\insert@count \@ne%
3209     \else

```

```

3210 \xright@appenditem{\csxdef{footnote@dir}{@RTLfalse}}\to\
inserts@list%Know the language of lemma
3211 \global\advance\insert@count \@ne%
3212 \fi%
3213 \xright@appenditem{\csxdef{footnote@lang}{\expandonce\language}}\
to\inserts@list%Know the language of lemma
3214 \global\advance\insert@count \@ne%
3215 }%
3216 {%
3217 \if@RTL
3218 \xright@appenditem{\csxdef{footnote@dir}{@RTLtrue}}\to\
inserts@listR%Know the language of lemma
3219 \global\advance\insert@countR \@ne%
3220 \else
3221 \xright@appenditem{\csxdef{footnote@dir}{@RTLfalse}}\to\
inserts@listR%Know the language of lemma
3222 \global\advance\insert@countR \@ne%
3223 \fi
3224 \xright@appenditem{\csxdef{footnote@lang}{\expandonce\language}}\
to\inserts@listR%Know the language of lemma
3225 \global\advance\insert@countR \@ne%
3226 }%
3227 }
3228 %

```

### XIII.4 General survey of the way we manage notes

The processing of each note is done by four principal macros: the `\vfootnote` macro takes the text of the footnote and does the `\insert`; it calls on the `\footfmt` macro to select the right fonts, print the line number and lemma, and do any other formatting needed for that individual note. Within the output routine, the two other macros, `\footstart` and `\footgroup`, are called; the first prints extra vertical space and a footnote rule, if desired; the second does any reformatting of the whole set of the footnotes in this series for this page—such as paragraphing or division into columns—and then sends them to the page.

These four macros, and the other macros and parameters shown here, are distinguished by the “series letter” that indicates which set of the footnotes we are dealing with—A, B, C, D, or E. The series letter always precedes the string `foot` in macro and parameter names. Hence, for the A series, the four macros are called `\vAfootnote`, `\Afootfmt`, `\Afootstart`, and `\Afootgroup`.

These macros are changed depending on the footnote 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 process 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 L<sup>A</sup>T<sub>E</sub>X `\floatingpenalty`.

```

3229 \newcommand*\footplitskips}{%
3230   \interlinepenalty=\interfootnotelinepenalty
3231   \unless\ifl@dprintingpages%
3232     \floatingpenalty=\@MM%
3233   \fi%
3234   \splittopskip=\ht\strutbox \splitmaxdepth=\dp\strutbox
3235   \leftskip=\z@skip \rightskip=\z@skip}
3236
3237 %

```

`\normalfootnoterule` `\normalfootnoterule` is a standard footnote-rule macro, for use by a `footstart` macro: just the same as the PLAIN T<sub>E</sub>X footnote rule.

```

3238 \let\normalfootnoterule=\footnoterule
3239 %

```

## 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.

```

3240 \newcommandx*\Xarrangement}[2][1,usedefault]{%
3241   \def\do##1{%
3242     \csname Xarrangement@#2\endcsname{##1}%
3243   }%
3244   \ifstrempy{#1}%
3245     {%
3246     \dolistloop{\@series}%
3247     }%
3248     {
3249     \docsvlist{#1}%
3250     }%
3251 }%
3252 %

```

### 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\vfootnote=\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.

```

3253 \newcommand*{\Xarrangement@normal}[1]{%
3254   \csgdef{series@display#1}{normal}
3255   \expandafter\let\csname #1footstart\endcsname=\normalfootstart
3256   \expandafter\let\csname v#1footnote\endcsname=\normalvfootnote
3257   \expandafter\let\csname #1footfmt\endcsname=\normalfootfmt
3258   \expandafter\let\csname #1footgroup\endcsname=\normalfootgroup
3259   \expandafter\let\csname #1footnoterule\endcsname=%
3260                                     \normalfootnoterule
3261   \count\csname #1footins\endcsname=1000
3262   \dimen\csname #1footins\endcsname=\csuse{Xmaxhnotes@#1}
3263   \skip\csname #1footins\endcsname=\csuse{Xbeforenotes@#1}%
3264   \advance\skip\csname #1footins\endcsname by\csuse{Xafterrule@#1}%
3265   %

```

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.

```

3266   \csxdef{default@#1footins}{1000}%Use this to confine the notes to one
side only
3267   %

```

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).

```

3268   \ifnoledgroup@else%
3269     \expandafter\let\csname mpv#1footnote\endcsname=\mpnormalvfootnote
3270     \expandafter\let\csname mp#1footgroup\endcsname=\mpnormalfootgroup
3271     \count\csname mp#1footins\endcsname=1000
3272     \dimen\csname mp#1footins\endcsname=\csuse{Xmaxhnotes@#1}
3273     \skip\csname mp#1footins\endcsname=\csuse{Xbeforenotes@#1}%
3274     \advance\skip\csname mp#1footins\endcsname by\csuse{Xafterrule@#1}%
3275     \fi
3276   }
3277
3278   %

```

`\normalvfootnote` We now begin a series of commands that do “normal” footnote formatting: a format much like that implemented in PLAIN  $\TeX$ , in which each footnote is a separate paragraph.

`\normalvfootnote` takes the series letter as #1 and the entire text of the footnote is #2. It does the `\insert` for this note, calling on the `\footfmt` macro for this note series to format the text of the note.

```

3279 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\normalvfootnote}[2]{%
3280 \iftoggle{Xgroupbyline@#1}{%In the case we use \Xgroupbyline, the
insertion is done later, in \add@Xgroupbyline.
3281 \prepare@Xgroupbyline{#1}{#2}{\normalvfootnote@inserted}%
3282 }{%In the case we don't use \Xgroupbyline, the insertion is made directly
3283 \X@beforeinsertion{#1}%
3284 \insert\csname #1footins\endcsname{%
3285 \X@atbegininsertion{#1}%
3286 \normalvfootnote@inserted{#1}{#2}%
3287 }%
3288 }%
3289 }%
3290 %

```

`\normalvfootnote@inserted` The `\normalvfootnote@inserted` macro is expanded to the content to be added to a `\insert` for normal critical footnote.

```

3291 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\
normalvfootnote@inserted}[2]{%
3292 \nottoggle{Xgroupbyline@#1}{\noindent}{\csuse{Xbhooknote@#1}%
3293 \csuse{Xnotefontsize@#1}%
3294 \iftoggle{Xgroupbyline@#1}{\strut}{}%
3295 \footplitskips
3296 \ifl@dpairing\ifl@dpaging\else%
3297 \setXnoteswidthliketwocolumns@{#1}%
3298 \fi\fi%
3299 \setXnotespositionliketwocolumns@{#1}%
3300 \spaceskip=\z@skip \xspaceskip=\z@skip%
3301 \csname #1footfmt\endcsname #2{#1}%
3302 }%
3303 %

```

```

\X@beforeinsertion04 \newcommand{\X@beforeinsertion}[1]{%
3305 \if@ledgroup\else%
3306 \insert@Xtxtbeforenotes{#1}%
3307 \fi%
3308 \csuse{Xbeforeinserting@#1}%
3309 }%
3310 %

```

```

\beforeinsertion@X11 \newcommand{\beforeinsertion@X}[1]{%
3312 \if@ledgroup\else%
3313 \insert@txtbeforenotesX{#1}%
3314 \fi%
3315 \csuse{beforeinsertingX@#1}%
3316 }%
3317 %

```

```

\X@atbegininsertion18 \newcommand{\X@atbegininsertion}[1]{%
3319 \hspace=\expandafter\dimexpr\csuse{Xwidth@#1}\relax%
3320 }%
3321 %

```

And somewhat different versions of `\normalvfootnote` and `\normalvfootnote@inserted` for minipages.

```
\mpnormalvfootnote 22 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\mpnormalvfootnote}[2]{%
3323 \iftoggle{Xgroupbyline@#1}{%
3324 \prepare@Xgroupbyline{#1}{#2}{\mpnormalvfootnote@inserted}%
3325 }%
3326 {%
3327 \global\setbox\@nameuse{mp#1footins}%
3328 \vbox{%
3329 \unvbox\@nameuse{mp#1footins}%
3330 \mpnormalvfootnote@inserted{#1}{#2}%
3331 }%
3332 }%
3333 }%
3334 %
3335 %
```

```
\mpnormalvfootnote@inserted 36 \newcommand{\mpnormalvfootnote@inserted}[2]{%
3337 \noindent\csuse{Xhooknote@#1}%
3338 \csuse{Xnotefontsize@#1}%
3339 \hsize\columnwidth%
3340 \@parboxrestore%
3341 \color@begingroup%
3342 \csname #1footfmt\endcsname #2{#1}\color@endgroup%
3343 }%
3344 %
```

`\normalfootfmt` `\normalfootfmt` is a “normal” macro to take the footnote line and page number information (see V.9 p. 105), 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.

```
3345 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\normalfootfmt}[4]{%
3346 \Xstorelineinfo{#1}{#4}%
3347 \nottoggle{Xgroupbyline@#4}{\Xledsetnormalparstuff{#4}}{%
3348 \Xsethangindent{#4}%
3349 \nottoggle{Xgroupbyline@#4}{\rule\z@\splittopskip}{}%
3350 {\printlinefootnote{#1}{#4}}%
3351 \print@lemma{#1}{#2}{#4}%
3352 \csuse{Xwrapcontent@#4}{#3}%
3353 \nottoggle{Xgroupbyline@#4}{\strut\par}{}%
3354 }%
3355 %
```

`\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`.

```
3356 \newcommand*{\normalfootstart}[1]{%
3357 %
```

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, Maieul Rouquette have defined an algorithm explained in [XIX p. 235](#). Here is part of this algorithm, when the block of notes are ready to be printed.

```
3358 \ifdimequal{0pt}{\Xprenotes@}{%
3359   {%
3360     \iftoggle{Xprenotes@}{%
3361       \togglefalse{Xprenotes@}%
3362       \skip\csname #1footins\endcsname=%
3363       \glueexpr\csuse{Xprenotes@}+\csuse{Xafterrule@#1}\relax%
3364     }%
3365   }%
3366 }%
3367 \vskip\skip\csname #1footins\endcsname%
3368 %
```

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. 233](#) for the general description of the problem.

```
3369 \leftskip0pt \rightskip0pt
3370 \ifl@dpairing\else%
3371   \hsize=\old@hsize%
3372 \fi%
3373 \setXnoteswidthliketwocolumns@{#1}%
3374 \setXnotespositionliketwocolumns@{#1}%
3375 %
```

And now, print the footnote's rule to finish the footnote's introduction.

```
3376 \print@Xfootnoterule{#1}%
3377 }%
3378 %
```

`\normalfootgroup` `\normalfootgroup` is a standard footnote-grouping macro: it sends the contents of the footnote-insert box to the output page without alteration.

```
3379 \newcommand*{\normalfootgroup}[1]{%
3380   \csuse{Xbhookgroup@#1}%
3381   \unvbox\csname #1footins\endcsname%
```

```

3382 \hsize=\old@hsize%
3383 }%
3384
3385 %

```

`\mpnormalfootgroup` A somewhat different version for minipages. Note that, in this case, we do not make distinctions between the `\Xfootgroup` and `\Xfootstarts` macros.

```

3386 \unless\ifnoledgroup@
3387 \newcommand*{\mpnormalfootgroup}[1]{%
3388 \vskip\skip\@nameuse{mp#1footins}
3389 \ifl@dpairing\ifparledgroup%
3390 \leavevmode\marks\parledgroup@{begin}%
3391 \marks\parledgroup@series{#1}%
3392 \marks\parledgroup@type{Xfootnote}%
3393 \fi\fi\normalcolor%
3394 \ifparledgroup%
3395 \ifl@dpairing%
3396 \else%
3397 \setXnoteswidthliketwocolumns@{#1}%
3398 \setXnotespositionliketwocolumns@{#1}%
3399 \print@Xfootnoterule{#1}%%
3400 \fi%
3401 \else%
3402 \setXnoteswidthliketwocolumns@{#1}%
3403 \setXnotespositionliketwocolumns@{#1}%
3404 \print@Xfootnoterule{#1}%%
3405 \fi%
3406 \setlength{\parindent}{0pt}
3407 \csuse{Xbhookgroup@#1}%
3408 \unvbox\csname mp#1footins\endcsname}}
3409 \fi
3410 %

```

### XIII.6.3 Paraphed footnotes

The paraphed-footnote option reformats all the footnotes of one series for a page into a single paragraph; this is especially appropriate when the notes are numerous and brief. The code is based on *The TeXbook*, pp. 398–400, with alterations for our environment. This algorithm uses a considerable amount of save-stack space: a TeX of ordinary size may not be able to handle more than about 100 notes of this kind on a page.

`\Xarrangement@paragraph` The `\Xarrangement@paragraph` macro sets up everything for one series of the footnotes so that they will be paraphed; 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 paraphed 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 paraphed.

```

3411 \newcommand*{\Xarrangement@paragraph}[1]{%
3412 \csgdef{series@display#1}{paragraph}
3413 \expandafter\let\csname #1footstart\endcsname=\parafootstart

```



```

3414 \expandafter\let\csname v#1footnote\endcsname=\paravfootnote
3415 \expandafter\let\csname #1footfmt\endcsname=\parafootfmt
3416 \expandafter\let\csname #1footgroup\endcsname=\parafootgroup
3417 \count\csname #1footins\endcsname=1000
3418 \csxdef{default@#1footins}{1000}%Use this to confine the notes to one
side only
3419 \dimen\csname #1footins\endcsname=\csuse{Xmaxhnotes@#1}
3420 \skip\csname #1footins\endcsname=\csuse{Xbeforenotes@#1}%
3421 \advance\skip\csname #1footins\endcsname by\csuse{Xafterrule@#1}%
3422 \para@footsetup{#1}
3423 %

```

And the extra setup for minipages.

```

3424 \ifnoledgroup@else
3425 \expandafter\let\csname mpv#1footnote\endcsname=\mpparavfootnote
3426 \expandafter\let\csname mp#1footgroup\endcsname=\mpparafootgroup
3427 \count\csname mp#1footins\endcsname=1000
3428 \dimen\csname mp#1footins\endcsname=\csuse{Xmaxhnotes@#1}
3429 \skip\csname mp#1footins\endcsname=\csuse{Xbeforenotes@#1}%
3430 \advance\skip\csname mp#1footins\endcsname by\csuse{Xafterrule@#1}%
3431 \fi
3432 }
3433 %

```

`\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.

```

3434 \providecommand{\footfudgefiddle}{64}
3435 %

```

`\para@footsetup` `\footparagraph` calls the `\para@footsetup` macro to calculate a special fudge factor, which is the ratio of the `\baselineskip` to the `\hsize`. We assume that the proper value of `\baselineskip` for the footnotes (normally 9pt) has been set already. The argument of the macro is again the note series letter.

Peter Wilson thinks that `\columnwidth` should be used here for  $\LaTeX$  not `\hsize`. Peter Wilson have also included `\footfudgefiddle`.

```

3436 \newcommand*{\para@footsetup}[1]{\csuse{Xbhookgroup@#1}\csuse{
Xnotefontsize@#1}
3437 \setXnoteswidthliketwocolumns@{#1}%
3438 \ifcsemt{Xwidth@#1}%
{}%
3439 {\columnwidth=\expandafter\dimexpr\csuse{Xwidth@#1}\relax}%
3440 \dimen0=\baselineskip
3441 \multiply\dimen0 by 1024
3442 \divide \dimen0 by \columnwidth \multiply\dimen0 by \footfudgefiddle\
relax
3444 \csxdef{#1footfudgefactor}{%
3445 \expandafter\strip@pt\dimen0 }}
3446 %
3447 %

```

`\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.

```

3448 \newcommand*{\parafootstart}[1]{%
3449   \rightskip=0pt \leftskip=0pt%
3450   \Xsetparindent{#1}%
3451   \ifdimequal{0pt}{\Xprenotes@}{-}{%
3452     {%
3453       \iftoggle{Xprenotes@}{%
3454         \togglefalse{Xprenotes@}%
3455         \skip\csname #1footins\endcsname=%
3456         \glueexpr\csuse{Xprenotes@}+\csuse{Xafterrule@#1}\relax%
3457       }%
3458     }%
3459   }%
3460   \vskip\skip\csname #1footins\endcsname%
3461   \setXnoteswidthliketwocolumns@{#1}%
3462   \setXnotespositionliketwocolumns@{#1}%
3463   \print@Xfootnoterule{#1}%
3464   \let\old@bidi@RTL@everypar\@bidi@RTL@everypar%
3465   \let\@bidi@RTL@everypar\@empty%
3466   \noindent\leavevmode%
3467   \let\@bidi@RTL@everypar\old@bidi@RTL@everypar%
3468 }%
3469 %

```

`\paravfootnote` `\paravfootnote` is a version of the `\vfootnote` command that is used for paragraphed notes. It gets appended to the `\inserts@list` list by an outer-level footnote command like `\Afootnote`. The first argument is the note series letter; the second is the full text of the printed note itself, including line numbers, lemmata, and footnote text.

The initial model for this insertion is, of course, the `\insert\footins` definition in *The TeXbook*, p. 398. There, the footnotes are first collected up in `hboxes`, and these `hboxes` are later unpacked and stuck together into a paragraph.

However, Michael Downes has pointed out that because text in `hboxes` gets typeset in restricted horizontal mode, there are some undesirable side-effects if you later want to break such text across lines. In restricted horizontal mode, where  $\TeX$  does not expect to have to break lines, it does not insert certain items like `\discretionary`s. If you later unbox these `hboxes` and stick them together, as the *TeXbook* macros do to make these footnotes, you lose the ability to hyphenate after an explicit hyphen. This can lead to overfull `\hboxes` when you would not expect to find them, and to the uninitiated it might be very hard to see why the problem had arisen.<sup>32</sup>

Wayne Sullivan pointed out to us another subtle problem that arises from the same cause:  $\TeX$  also leaves the `\language` `whatsit` nodes out of the horizontal list.<sup>33</sup> So changes from one language to another will not invoke the proper hyphenation rules in

<sup>32</sup>Michael Downes, “Line Breaking in `\unboxed` Text”, *TUGboat* 11 (1990), pp. 605–612.

<sup>33</sup>See *The TeXbook*, p. 455 (editions after January 1990).

such footnotes. Since critical editions often do deal with several languages, especially in a footnotes, we really ought to get this bit of code right.

To get around these problems, Wayne suggested emendations to the *TeXbook* versions of these macros which are broadly the same as those described by Michael: the central idea (also suggested by Donald Knuth in a letter to Michael) is to avoid collecting the text in an `\hbox` in the first place, but instead to collect it in a `\vbox` whose width is (virtually) infinite. The text is therefore typeset in unrestricted horizontal mode, as a paragraph consisting of a single long line. Later, there is an extra level of unboxing to be done: we have to unpack the `\vbox`, as well as the `\hboxes` inside it, but that is not too hard. For details, we refer you to Michael's article, where the issues are clearly explained.<sup>34</sup> Michael's unboxing macro is called `\Xunvxh`: unvbox, extract the last line, and unhbox it.

Doing things this way has an important consequence: as Michael pointed out, you really can't put an explicit line-break into a note built in a `\vbox` the way we are doing.<sup>35</sup> In other words, be very careful not to use `\break`, or `\penalty-10000`, or any equivalent inside your para-footnote. If you do, most of the note will probably disappear. You *are* allowed to make strong suggestions; in fact `\penalty-9999` will be quite okay. Just do not make the break mandatory. We have not applied any of Michael's solutions here, since we feel that the problem is exiguous, and `reledmac` is quite baroque enough already. If you think you are having this problem, look up Michael's solutions.

One more thing; we set `\leftskip` and `\rightskip` to zero. This has the effect of neutralizing any such skips which may apply to the main text (cf. XIII.6.2 p. 183 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.

```

3470 \newcommand*{\paravfootnote}[2]{%
3471   \csuse{Xbeforeinserting@#1}%
3472   \insert\csname #1footins\endcsname
3473   \bgroup
3474   \csuse{Xnotefontsize@#1}
3475   \footsplitskips
3476   \setbox0=\vbox{\hsize=\maxdimen%
3477     \let\bidir@RTL@everypar\@empty%
3478     \insert@Xtxtbeforenotes{#1}%
3479     \noindent\csuse{Xhooknote@#1}%
3480     \csname #1footfmt\endcsname #2{#1}}%
3481   \setbox0=\hbox{\Xunvxh{0}{#1}}%
3482   \dp0=0pt
3483   \ht0=\csname #1footfudgefactor\endcsname\wd0
3484 %

```

Here we produce the contents of the footnote from box 0, and add a penalty of 0 between boxes in this insert.

```

3485   \if@RTL\noindent \leavevmode\fi\box0%
3486   \penalty0
3487 \egroup}
3488

```

<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.

3489 %

The final penalty of 0 was added here at Wayne’s suggestion to avoid a weird page-breaking problem, which occurs on those occasions when  $\TeX$  attempts to split foot paragraphs. After trying out such a split (see *The TeXbook*, p. 124),  $\TeX$  inserts a penalty of  $-10000$  here, which nearly always forces the break at the end of the whole footnote paragraph (since individual notes can’t be split) even when this leads to an overfull vbox. The change above results in a penalty of 0 instead which allows, but does not force, such breaks. This penalty of 0 is later removed, after page breaks have been decided, by the `\unpenalty` macro in `\makehboxofhboxes`. So it does not affect how the footnote paragraphs are typeset (the notes still have a penalty of  $-10$  between them, which is added by `\parafootfmt`).

`\mpparavfootnote` This version is for minipages.

```

3490 \newcommand*\mpparavfootnote}[2]{%
3491   \global\setbox\@nameuse{mp#1footins}\vbox{%
3492     \unvbox\@nameuse{mp#1footins}%
3493     \csuse{Xnotefontsize@#1}
3494     \footplitskips
3495     \setbox0=\vbox{\hsize=\maxdimen%
3496       \let\bidirectional@everypar\@empty%
3497       \insert@Xtxtbeforenotes{#1}%
3498       \noindent\color@begingroup%
3499       \csuse{Xbhooknote@#1}%
3500       \csname #1footfmt\endcsname #2{#1}\color@endgroup}%
3501     \setbox0=\hbox{\Xunvxh{0}{#1}}%
3502     \dp0=\z@
3503     \ht0=\csname #1footfudgefactor\endcsname\wd0
3504     \box0
3505     \penalty0
3506   }}
3507
3508 %

```

`\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`.

```

3509 \newcommand*\Xunvxh}[2]{%
3510   \setbox0=\vbox{\unvbox#1%
3511     \global\setbox1=\lastbox}%
3512   \unhbox1
3513   \unskip           % remove \rightskip,
3514   \unskip           % remove \parfillskip,
3515   \unpenalty       % remove \penalty of 10000,
3516   \hskip\csuse{Xafternote@#2}\relax}% add the glue to go between the notes
3517
3518 %

```

`\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).

```

3519 \newcommand*\parafootfmt}[4]{%
3520   \Xstorelineinfo{#1}{#4}%
3521   \Xinsertparafootsep{#4}%
3522   \ledsetnormalparstuff@common%
3523   \printlinefootnote{#1}{#4}%
3524   \print@lemma{#1}{#2}{#4}%
3525   \csuse{Xwrapcontent@#4}{#3}%
3526   \penalty-10 }
3527 %

```

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.

```

3528 \newcommand*\parafootgroup}[1]{%
3529   \hsize=\expandafter\dimexpr\csuse{Xwidth@#1}\relax%
3530   \unvbox\csname #1footins\endcsname
3531   \ifcsstring{Xragged@#1}{L}{\RaggedLeft}{}%
3532   \ifcsstring{Xragged@#1}{R}{\RaggedRight}{}%
3533   \makehboxofhboxes
3534   \setbox0=\hbox{\unhbox0 \removehboxes}%
3535   \csuse{Xbhookgroup@#1}%
3536   \csuse{Xnotefontsize@#1}%
3537   \unhbox0\par%
3538   \global\hsize=\old@hsize%
3539   }%
3540
3541 %

```

`\mpparafootgroup` The minipage version.

```

3542 \newcommand*\mpparafootgroup}[1]{%
3543   \setXnoteswidthliketwocolumns@{#1}%
3544   \vskip\skip\@nameuse{mp#1footins}
3545   \ifl@dpairing\ifparledgroup%
3546     \leavevmode\marks\parledgroup@{begin}%
3547     \marks\parledgroup@series{#1}%
3548     \marks\parledgroup@type{Xfootnote}%
3549   \fi\fi\normalcolor
3550   \ifparledgroup%
3551     \ifl@dpairing%
3552     \else%

```

```

3553 \setXnoteswidthliketwocolumns@{#1}%
3554 \setXnotespositionliketwocolumns@{#1}%
3555 \print@Xfootnoterule{#1}%%
3556 \fi%
3557 \else%
3558 \setXnoteswidthliketwocolumns@{#1}%
3559 \setXnotespositionliketwocolumns@{#1}%
3560 \print@Xfootnoterule{#1}%
3561 \fi%
3562 \unvbox\csname mp#1footins\endcsname
3563 \ifcsstring{Xragged@#1}{L}{\RaggedLeft}{}%
3564 \ifcsstring{Xragged@#1}{R}{\RaggedRight}{}%
3565 \makehboxofhboxes
3566 \setbox0=\hbox{\unhbox0 \removehboxes}%
3567 \cuse{Xbhookgroup@#1}%
3568 \cuse{Xnotefontsize@#1}%
3569 \Xsetparindent{#1}%
3570 \unhbox0\par}}
3571
3572 %

```

And finally, the two macros which are required to transform the long horizontal box stored in the insert' box to a printable text.

```

\makehboxofhboxes 73 \newcommand*\makehboxofhboxes{\setbox0=\hbox{}}%
\removehboxes 74 \loop
3575 \unpenalty
3576 \setbox2=\lastbox
3577 \ifhbox2
3578 \setbox0=\hbox{\box2\unhbox0}%
3579 \repeat}
3580
3581 \newcommand*\removehboxes{\setbox0=\lastbox
3582 \ifhbox0{\removehboxes}\unhbox0 \fi}
3583
3584 %

```

**Insertion of the footnotes separator** The command `\Xinsertparafootsep{<series>}` must be called at the beginning of `\parafootftm`.

```

\prevpage@num 85 \newcommand{\Xinsertparafootsep}[1]{%
\Xinsertparafootsep 86 \ifledRcol@%
3587 \ifnumequal{\cuse{#1prevpage@numR}}{\page@numR}%
3588 {\ifcsdef{prevline#1}% Be sur \prevline#1 exists.
3589 {\ifcsequal{prevline#1}{lineinfo@}%
3590 {\ifcsequal{Xsymlinenum@#1}{\cuse{Xparafootsep@#1}}{}}%
3591 {\cuse{Xparafootsep@#1}}%
3592 }%
3593 {\cuse{Xparafootsep@#1}}%
3594 }%
3595 {}%
3596 \global\csname #1prevpage@numR\endcsname=\page@numR%

```

```

3597 \else%
3598 \ifunequal{\csuse{#1prevpage@num}}{\page@num}%
3599 {\ifcsdef{prevline#1}% Be sur \prevline#1 exists.
3600 {\ifcsequal{prevline#1}{lineinfo@}%
3601 {\ifcsequal{Xsymlinenum@#1}{\csuse{Xparafootsep@#1}}{}}%
3602 {\csuse{Xparafootsep@#1}}%
3603 }%
3604 {\csuse{Xparafootsep@#1}}%
3605 }%
3606 }%
3607 \global\csname #1prevpage@num\endcsname=\page@num%
3608 \fi%
3609 }
3610 %

```

### XIII.6.4 Columnar footnotes

#### Common tools

`\rigidbalance` We will now define macros for three-column notes and two-column notes. Both sets  
`\rigidbalanceX` of macros will use `\rigidbalance`, which splits a box (#1) into into a number (#2) of  
`\Xrigidbalance` columns, each with a space (#3) between the top baseline and the top of the `\vbox`. The  
`\dosplits` `\rigidbalance` macro is taken from *The TeXbook*, p. 397, with a slight change to the  
`\splitoff` syntax of the arguments so that they do not depend on white space. Note also the extra  
`\@h` unboxing in `\splitoff`, which allows the new `\vbox` to have its natural height as it  
`\@k` goes into the alignment.

The  $\TeX$  `\line` macro has no relationship to the TeX `\line`. The  $\TeX$  equivalent is `\@@line`.

We do not call directly `\rigidbalance`, but we call `\Xrigidbalance` for critical notes and `\rigidbalanceX` for familiar notes. Both of them call `\rigidbalance`.

```

3611 \newcount\@k \newdimen\@h
3612 \newcommand*\Xrigidbalance}[3]{%
3613 \hsize=\expandafter\dimexpr\csuse{Xwidth@\@currentseries}\relax%
3614 \rigidbalance{#1}{#2}{#3}%
3615 }%
3616
3617 \newcommand*\rigidbalanceX}[3]{%
3618 \hsize=\expandafter\dimexpr\csuse{widthX@\@currentseries}\relax%
3619 \rigidbalance{#1}{#2}{#3}%
3620 }%
3621
3622 \newcommand*\rigidbalance}[3]{%
3623 \setbox0=\box#1 \@k=#2 \@h=#3%
3624 \@@line{\splittopskip=\@h \vbadness=\@M \hfilneg
3625 \valign{##\vfil\cr\dosplits}}
3626
3627 \newcommand*\dosplits{\ifnum\@k>0 \noalign{\hfil}\splitoff
3628 \global\advance\@k-1\cr\dosplits\fi}
3629
3630 \newcommand*\splitoff{\dimen0=\ht0
3631 \divide\dimen0 by\@k \advance\dimen0 by\@h
3632 \setbox2 \vsplit0 to \dimen0

```

```

3633 \unvbox2 }
3634
3635 %

```

### Three columns

```

\Xarrangement@threecol 36 \newcommand*{\Xarrangement@threecol}[1]{%
3637 \csgdef{series@display#1}{threecol}
3638 \expandafter\let\csname v#1footnote\endcsname=\threecolvfootnote
3639 \expandafter\let\csname #1footfmt\endcsname=\threecolfootfmt
3640 \expandafter\let\csname #1footgroup\endcsname=\threecolfootgroup
3641 \dimen\csname #1footins\endcsname=\csuse{Xmaxhnotes@#1}%
3642 \skip\csname #1footins\endcsname=\csuse{Xbeforenotes@#1}%
3643 \advance\skip\csname #1footins\endcsname by\csuse{Xafterrule@#1}%
3644 \threecolfootsetup{#1}
3645 %

```

The additional setup for minipages.

```

3646 \ifnoledgroup@else
3647 \expandafter\let\csname mpv#1footnote\endcsname=\mpnormalvfootnote
3648 \expandafter\let\csname mp#1footgroup\endcsname=\mpthreecolfootgroup
3649 \skip\csname mp#1footins\endcsname=\csuse{Xbeforenotes@#1}%
3650 \advance\skip\csname mp#1footins\endcsname by\csuse{Xafterrule@#1}%
3651 \mpthreecolfootsetup{#1}
3652 \fi
3653 }
3654
3655 %

```

The `\footstart` and `\footnoterule` macros for these notes assume the normal values (XIII.6.2 p. 182 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.

```

3656 \newcommand*{\threecolfootsetup}[1]{%
3657 \count\csname #1footins\endcsname 333
3658 \csxdef{default@#1footins}{333}%Use this to confine the notes to one
side only
3659 \multiply\dimen\csname #1footins\endcsname \thr@@}
3660 %

```



`\mpthreecolfootsetup` The setup for minipages.

```

3661 \newcommand*\mpthreecolfootsetup}[1]{%
3662   \count\csname mp#1footins\endcsname 333
3663   \multiply\dimen\csname mp#1footins\endcsname \thr@@}
3664
3665 %

```

`\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 `\hspace` to 0.3 of its normal value. This determines the widths of the individual columns. So if the normal `\hspace` 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).

```

3666 \notbool{parapparatus@}\newcommand*\newcommand*\threecolvfootnote}[2]{%
3667   \iftoggle{Xgroupbyline@#1}{%
3668     \prepare@Xgroupbyline{#1}{#2}\threecolvfootnote@inserted}%
3669   }%
3670   {%
3671     \let\bidir@RTL@everypar\relax%
3672     \X@beforeinsertion{#1}%
3673     \insert\csname #1footins\endcsname{%
3674       \threecolvfootnote@inserted{#1}{#2}%
3675     }%
3676   }%
3677 }%
3678 %

```

```

\threecolvfootnote@inserted 3679 \notbool{parapparatus@}\newcommand*\newcommand*\threecolvfootnote@inserted}[2]{%
3680   \hspace=\expandafter\dimexpr\csuse{Xwidth@#1}\relax%
3681   \noindent\csuse{Xhooknote@#1}%
3682   \csuse{Xnotefontsize@#1}%
3683   \footssplitskips%
3684   \csname #1footfmt\endcsname #2{#1}%
3685 }%
3686 %

```

`\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.

```

3687 \notbool{parapparatus@}\newcommand*\newcommand*\threecolfootfmt}[4]{%
3688   \Xstorelineinfo{#1}{#4}%
3689   \threecol@begin@insert{#4}%
3690   \hspace{\parindent}%
3691   \printlinefootnote{#1}{#4}%
3692   \print@lemma{#1}{#2}{#4}%

```

```

3693 \csuse{Xwrapcontent@#4}{#3}%
3694 \nottoggle{Xgroupbyline@#4}%
3695   {\strut\par\allowbreak}%
3696   }%
3697 }%
3698 %

```

`\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`.

```

3699 \newcommand{\threecol@begin@insert}[1]{%
3700   \normal@pars%
3701   \nottoggle{Xgroupbyline@#1}%
3702   {\hspace \csuse{Xhsizethreecol@#1}}%
3703   }%
3704   \Xsetparindent{#1}%
3705   \tolerance=5000%
3706   \Xsethangindent{#1}%
3707   \@tempdima=\parindent%
3708   \csuse{Xcolalign@#1}%
3709   \parindent=\@tempdima%
3710   \strut%
3711 }%
3712 %

```

`\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.

```

3713 \newcommand*{\threecolfootgroup}[1]{%
3714   \begingroup%
3715     \csuse{Xhookgroup@#1}%
3716     \csuse{Xnotefontsize@#1}%
3717     \par%
3718     \splittopskip=\ht\strutbox%
3719     \expandafter%
3720     \Xrigidbalance\csname #1footins\endcsname \thr@@ \splittopskip%
3721   \endgroup%
3722 }%
3723 %

```

`\mpthreecolfootgroup` The setup for minipages.

```

3724 \newcommand*{\mpthreecolfootgroup}[1]{%
3725   \vskip\skip\@nameuse{mp#1footins}
3726   \ifl@dpairing\ifparledgroup%

```

```

3727 \leavevmode\marks\parledgroup@{begin}%
3728 \marks\parledgroup@series{#1}%
3729 \marks\parledgroup@type{Xfootnote}%
3730 \fi\fi\normalcolor
3731 \ifparledgroup%
3732 \ifl@dpairing%
3733 \else%
3734 \setXnoteswidthliketwocolumns@{#1}%
3735 \setXnotespositionliketwocolumns@{#1}%
3736 \print@Xfootnoterule{#1}%
3737 \fi%
3738 \else%
3739 \setXnoteswidthliketwocolumns@{#1}%
3740 \setXnotespositionliketwocolumns@{#1}%
3741 \print@Xfootnoterule{#1}%
3742 \fi%
3743 \csuse{Xbhookgroup@#1}\par%
3744 \splittopskip=\ht\strutbox
3745 \expandafter
3746 \Xrigidbalance\csname mp#1footins\endcsname \thr@@ \splittopskip}}
3747
3748 %

```

### Two columns

```

\Xarrangement@twocol149 \newcommand*\Xarrangement@twocol}[1]{%
3750 \csgdef{series@display#1}{twocol}
3751 \expandafter\let\csname v#1footnote\endcsname=\twocolvfootnote
3752 \expandafter\let\csname #1footfmt\endcsname=\twocolfootfmt
3753 \expandafter\let\csname #1footgroup\endcsname=\twocolfootgroup
3754 \dimen\csname #1footins\endcsname=\csuse{Xmaxhnotes@#1}%
3755 \skip\csname #1footins\endcsname=\csuse{Xbeforenotes@#1}%
3756 \advance\skip\csname #1footins\endcsname by\csuse{Xafterrule@#1}%
3757 \twocolfootsetup{#1}
3758 %

```

The additional setup for minipages.

```

3759 \ifnoledgroup@else
3760 \expandafter\let\csname mpv#1footnote\endcsname=\mpnormalvfootnote
3761 \expandafter\let\csname mp#1footgroup\endcsname=\mptwocolfootgroup
3762 \skip\csname mp#1footins\endcsname=\csuse{Xbeforenotes@#1}%
3763 \advance\skip\csname mp#1footins\endcsname by\csuse{Xafterrule@#1}%
3764 \mptwocolfootsetup{#1}
3765 \fi
3766 }
3767
3768 %

```

`\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`

```

3769 \newcommand*{\twocolfootsetup}[1]{%
3770   \count\csname #1footins\endcsname 500
3771   \csxdef{default@#1footins}{500}%
3772   \multiply\dimen\csname #1footins\endcsname \tw@}
3773 %

3774 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\twocolvfootnote}[2]{%
3775   \iftoggle{Xgroupbyline@#1}{%
3776     \prepare@Xgroupbyline{#1}{#2}{\twocolvfootnote@inserted}%
3777   }{%
3778     \let\bidir@RTL@everypar\relax%
3779     \X@beforeinsertion{#1}%
3780     \insert\csname #1footins\endcsname{%
3781       \twocolvfootnote@inserted{#1}{#2}%
3782     }%
3783   }%
3784 }%
3785 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\twocolvfootnote@inserted}[2]{%
3786   \hsize=\expandafter\dimexpr\csuse{Xwidth@#1}\relax%
3787   \noindent\csuse{Xbhooknote@#1}%
3788   \csuse{Xnotefontsize@#1}%
3789   \footsplitskips%
3790   \csname #1footfmt\endcsname #2{#1}%
3791 }%
3792 %

3793 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\twocolfootfmt}[4]{% 4th
arg is optional, for backward compatibility
3794   \Xstorelineinfo{#1}{#4}%
3795   \twocol@begin@insert{#4}%
3796   \hspace{\parindent}%
3797   \printlinefootnote{#1}{#4}%
3798   \print@lemma{#1}{#2}{#4}%
3799   \csuse{Xwrapcontent@#4}{#3}%
3800   \nottoggle{Xgroupbyline@#4}%
3801     {\strut\par\allowbreak}%
3802   }%
3803 }%
3804 %

3805 \newcommand{\twocol@begin@insert}[1]{%
3806   \normal@pars%
3807   \nottoggle{Xgroupbyline@#1}%
3808     {\hsize \csuse{Xhsizetwocol@#1}}%
3809   }%
3810   \Xsetparindent{#1}%
3811   \tolerance=5000%
3812   \Xsethangindent{#1}%
3813   \@tempdima=\parindent%
3814   \csuse{Xcolalign@#1}%
3815   \parindent=\@tempdima%
3816   \strut%
3817 }%

```

```

3818
3819 \newcommand*{\twocolfootgroup}[1]{%
3820   \begingroup%
3821   \csuse{Xbhookgroup@#1}%
3822   \csuse{Xnotefontsize@#1}%
3823   \par%
3824   \splittopskip=\ht\strutbox%
3825   \expandafter%
3826   \Xrigidbalance\csname #1footins\endcsname \tw@ \splittopskip%
3827   \endgroup%
3828 }%
3829
3830 %

```

`\mptwocolfootsetup` The versions for minipages.

`\mptwocolfootgroup`

```

3831 \newcommand*{\mptwocolfootsetup}[1]{%
3832   \count\csname mp#1footins\endcsname 500
3833   \multiply\dimen\csname mp#1footins\endcsname \tw@}
3834 %

3835 \newcommand*{\mptwocolfootgroup}[1]{%
3836   \vskip\skip\@nameuse{mp#1footins}
3837   \ifl@dpairing\ifparledgroup%
3838     \leavevmode\marks\parledgroup@{begin}%
3839     \marks\parledgroup@series{#1}%
3840     \marks\parledgroup@type{Xfootnote}%
3841     \fi\fi\normalcolor
3842   \ifparledgroup%
3843     \ifl@dpairing%
3844     \else%
3845       \setXnoteswidthliketwocolumns@{#1}%
3846       \setXnotespositionliketwocolumns@{#1}%
3847       \print@Xfootnoterule{#1}%
3848     \fi%
3849   \else%
3850     \setXnoteswidthliketwocolumns@{#1}%
3851     \setXnotespositionliketwocolumns@{#1}%
3852     \print@Xfootnoterule{#1}%
3853   \fi%
3854   \csuse{Xbhookgroup@#1}\par%
3855   \splittopskip=\ht\strutbox
3856   \expandafter
3857   \Xrigidbalance\csname mp#1footins\endcsname \tw@ \splittopskip}}
3858
3859 %

```

### XIII.7 Footnote paragraph indent

`\Xsetparindent` `\setparindentX` These two commands set the paragraph indentation of the footnotes, depending on the settings of the user.

```

3860 \newcommand{\Xsetparindent}[1]{%
3861   \nottoggle{Xparindent@#1}{%

```

```

3862 \parindent=\z@%
3863 }%
3864 {%
3865 \ifdef{\parindent@before stanza}{\parindent=\parindent@before stanza}{}%
3866 }%
3867 }%
3868 \newcommand{\setparindentX}[1]{%
3869 \nottoggle{parindentX@#1}{%
3870 \parindent=\z@%
3871 }%
3872 }%
3873 \ifdef{\parindent@before stanza}{\parindent=\parindent@before stanza}{}%
3874 }%
3875 }%
3876 %

```

### XIII.8 Footnote hanging indent

`\Xsethangindent` `\Xsethangindent` sets the hangindent for a critical footnote, while `\sethangindentX` does it for a familiar footnote.

```

3877 \newcommand{\Xsethangindent}[1]{%
3878 \if@RTL%
3879 \hangindent=-\csuse{Xhangindent@#1}%
3880 \everypar{\hangindent=-\csuse{Xhangindent@#1}}%
3881 \else%
3882 \hangindent=\csuse{Xhangindent@#1}%
3883 \everypar{\hangindent=\csuse{Xhangindent@#1}}%
3884 \fi%
3885 }%
3886 \newcommand{\sethangindentX}[1]{%
3887 \if@RTL%
3888 \hangindent=-\csuse{hangindentX@#1}%
3889 \everypar{\hangindent=-\csuse{hangindentX@#1}}%
3890 \else%
3891 \hangindent=\csuse{hangindentX@#1}%
3892 \everypar{\hangindent=\csuse{hangindentX@#1}}%
3893 \fi%
3894 }%
3895 %

```

### XIII.9 Critical notes presentation

Here, we define some common macro which are used in order to print a critical notes, that is a note with 1) line number 2) lemma 3) lemma separator 4) text associated to the lemma.

#### XIII.9.1 Font tools

`\endashchar` `\fullstop` `\rbracket` The fonts that are used for printing notes might not have the character mapping we expect: for example, the Computer Modern font that contains old-style numerals does not contain an en-dash or square brackets, and its period and comma are in odd locations.

To allow use of the standard footnote macros with such fonts, we use the following macros for certain characters.

The `\endashchar` macro is simply an en-dash from the normal font and is immune to changes in the surrounding font. The same goes for the full stop. These two are used in `\printlines`. The right bracket macro is the same again; it crops up in `\normalfootfmt` and the other footnote macros for controlling the format of the footnotes.

Note that these commands are not directly called by `reledmac`, but are enclosed as default value of specific hooks. Consequently, people should not redefine them, but use instead the `\Xlinerangeseparator`, `\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.

```

3896 \def\endashchar{\textnormal{--}}
3897
3898 \newcommand*\fullstop{\textnormal{.}}
3899 \def\Xsublinesep@side{\fullstop}
3900
3901 \newcommand*\rbracket{\textnormal{%
3902   \csuse{text\csuse{footnote@lang}}{%
3903     \ifluatex%
3904       \ifdefstring{\footnote@luatextextdir}{TRT}{\thinspace []}{\thinspace
3905     }%
3906     \else%
3907     \thinspace]}%
3908   \fi}%
3909 }
3910
3911 %

```

### XIII.9.2 Pstart number in footnote

`\printpstart` The `\printpstart` macro prints the pstart number for a note.

```

3912 \newcommand{\printpstart}[0]{%
3913   \ifboolexpr{bool{l@dpairing} or bool{l@dprintingpages} or bool{
3914     l@dprintingcolumns}}{%
3915     \ifledRcol%
3916     \thepstartR%
3917   \else%
3918   \thepstartL%
3919   \fi}%
3920   \thepstart%
3921 }%
3922 }
3923 %

```

### XIII.9.3 Lemma printing

`\print@lemma` `\print@lemma` is called inside critical footnotes to print the lemma and the lemma separator (#1: line number and font information, #2: lemma, #3: series).

```

3924 %
3925 \newcommand{\print@lemma}[3]{%
3926   \bgroup%
3927   \nottoggle{Xlemmadisablefontselection@#3}%
3928   {\select@lemmafnt#1|}%
3929   }%
3930   \bgroup%
3931   \csuse{Xlemmafnt@#3}%Deprecated
3932   \csuse{Xwraplemma@#3}-#2}%
3933   \egroup%
3934   \egroup%
3935   \iftoggle{nosep@}{%
3936     \hskip\csuse{Xinplaceoflemmaseparator@#3}%
3937     \relax%
3938   }%
3939   {\ifcsemt{Xlemmaseparator@#3}%
3940     {%
3941       \hskip\csuse{Xinplaceoflemmaseparator@#3}%
3942       \relax%
3943     }%
3944     {%
3945       \nobreak%
3946       \hskip\csuse{Xbeforelemmaseparator@#3}%
3947       \csuse{Xlemmaseparator@#3}%
3948       \hskip\csuse{Xafterlemmaseparator@#3}%
3949       \relax%
3950     }%
3951   }%
3952 }%
3953 %

```

### XIII.9.4 Line number printing

`\Xstorelineinfo` The `\Xstorelineinfo` macro is used to store some data about line number of the current critical footnote, data which will be reused later for the `\Xnumberonlyfirstinline` and related setting.

#1 footnote specification for the current footnote; #2 footnote series.

```

3954 \newcommand{\Xstorelineinfo}[2]{%
3955   \l@dp@rsefootspec#1|%
3956   \iftoggle{Xnumberonlyfirstintwolines@#2}{%
3957     \xdef\lineinfo@{\l@dparsedstartline - \l@dparsedstartsub - \
3958     \l@dparsedendline - \l@dparsedendsub}%
3959     }%
3960     \xdef\lineinfo@{\l@dparsedstartline - \l@dparsedstartsub}%
3961     }%
3962 }%
3963 %

```



`\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`.

```

3964 \newcommand{\printlinefootnote}[2]{%
3965   \xdef\@currentseries{#2}%
3966   \iftoggle{nonum@}{%Try if the line number must printed for this specific
not (by default, yes)
3967     \hspace{\csuse{Xinplaceofnumber@#2}}%
3968   }%
3969   {%
3970     {%
3971       \iftoggle{Xnonumber@#2}%Try if the line number must printed (by
default, yes)
3972       {%
3973         \hspace{\csuse{Xinplaceofnumber@#2}}%
3974       }%
3975       {%
3976         {\iftoggle{Xnumberonlyfirstinline@#2}% If for this series the
line number must be printed only in the first time.
3977         {%
3978           \ifcsdef{prevline#2}%
3979             {%Be sure the \prevline exists.
3980             \ifcsequal{prevline#2}{\lineinfo@}%Try it
3981             {%
3982             \ifcsequal{Xsymlinenum@#2}%
3983             {%
3984               \hspace{\csuse{Xinplaceofnumber@#2}}%
3985             }%
3986             {\printsymlinefootnotearea{#2}}%
3987           }%
3988           {%
3989             \printlinefootnotearea{#1}{#2}%
3990           }%
3991         }%
3992         {%
3993           \printlinefootnotearea{#1}{#2}%
3994         }%
3995       }%
3996       {%
3997         \printlinefootnotearea{#1}{#2}%
3998       }%
3999       \csxdef{prevline#2}{\lineinfo@}%
4000     }%
4001   }%
4002 }%
4003 }%
4004 }
4005 %

```

`\printsymlinefootnotearea` This macro prints the space before the line symbol, changes the font, when prints the line symbol and the space after it.

```

4006 \newcommand{\printsymlinefootnotearea}[1]{%
4007   \hspace{\csuse{Xbeforexymlineum@#1}}%
4008   \csuse{Xnotenumfont@#1}%
4009   \ifdimequal{\csuse{Xboxxymlineum@#1}}{\z@}%
4010     {\csuse{Xsymlineum@#1}}%
4011     {\hbox to \csuse{Xboxxymlineum@#1}%
4012       {\csuse{Xsymlineum@#1}\hfill}}%
4013   }%
4014   \hspace{\csuse{Xaftersymlineum@#1}}%
4015 }%
4016 %

```

**\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 on the options about repeating line numbers. The first argument is line information, the second is the notes series (A, B, C, etc.)

```

4017 \newcommand{\printlinefootnotearea}[2]{%
4018   \printXbeforenumber{#2}%
4019   \csuse{Xnotenumfont@#2}%
4020   \boxfootnotenumbers{#1}{#2}%
4021   \printXafternumber{#2}%
4022 }%
4023 %

```

**\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.

```

4024 \newcommand{\boxfootnotenumbers}[2]{%
4025   \ifdimequal{\csuse{Xboxxlineum@#2}}{Opt}{%
4026     \printlinefootnotenumbers{#1}{#2}%
4027   }%
4028   {%
4029     \hbox to \csuse{Xboxxlineum@#2}%
4030     {%
4031       \IfSubStr{RC}{\csuse{Xboxxlineumalign@#2}}{\hfill}{}%
4032       \printlinefootnotenumbers{#1}{#2}%
4033       \IfSubStr{LC}{\csuse{Xboxxlineumalign@#2}}{\hfill}{}%
4034     }%
4035   }%
4036 }%
4037 %

```

**\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.

```

4038 \newcommand{\printlinefootnotenumbers}[2]{%
4039   \get@pstartcounter%
4040   \ifboolexpr{%
4041     (togl{Xpstart@#2} and bool{numberpstart})%
4042     or togl{Xpstarteverytime@#2}%
4043     and not (%

```

```

4044     togl{Xpstartonlyfirst@#2}%
4045     and test {\ifnumequal{\csuse{Xprevpstart@#2}}{\current@pstart}}%
4046   )%
4047 }%
4048 {%
4049   \printpstart%
4050   \csuse{Xpstartseparator@#2}%
4051 }{}%
4052 \global\cslet{Xprevpstart@#2}{\current@pstart}%
4053 \get@stanzacounter%
4054 \ifboolexpr{%
4055   togl{Xstanzaonlyfirst@#2}%
4056   and bool {numberstanza}%
4057   and not (%
4058     togl{Xstanzaonlyfirst@#2}%
4059     and test {\ifnumequal{\csuse{Xprevstanza@#2}}{\current@stanza}}%
4060   )%
4061 }{%
4062   \printstanza%
4063   \csuse{Xstanzaseparator@#2}%
4064 }{}%
4065 \global\cslet{Xprevstanza@#2}{\current@stanza}%
4066 \iftoggle{Xonlypstart@#2}{-}{%
4067   \csuse{Xtxtbeforenumber@#2}%
4068   \printlines#1|\ifledRcol@{\@Rlineflag\fi}}%
4069 }%
4070 %

```

`\get@pstartcounter` A single macro to get the current pstart counter.

```

4071 \newcommand{\get@pstartcounter}[0]{%
4072   \ifboolexpr{bool{!@dprintingpages} or bool{!@dprintingcolumns}}%
4073   {%
4074     \ifledRcol%
4075     \global\numdef\current@pstart{\the\c@pstartR}%
4076     \else%
4077     \global\numdef\current@pstart{\the\c@pstartL}%
4078     \fi%
4079   }%
4080   {%
4081     \global\numdef\current@pstart{\the\c@pstart}%
4082   }%
4083 }%
4084 %

```

`\get@stanzacounter` The same to get the current stanza counter.

```

4085 \newcommand{\get@stanzacounter}[0]{%
4086   \ifboolexpr{bool{!@dprintingpages} or bool{!@dprintingcolumns}}%
4087   {%
4088     \ifledRcol%
4089     \global\numdef\current@stanza{\the\c@stanzaR}%
4090     \else%
4091     \global\numdef\current@stanza{\the\c@stanzaL}%

```

```

4092   \fi%
4093   }%
4094   {%
4095   \global\newdef\current@stanza{\the\c@stanza}%
4096   }%
4097 }%
4098 %

```

`\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.)

```

4099 \newcommand{\printXbeforenumber}[1]{%
4100   \hspace{\csuse{Xbeforenumber@#1}}%
4101 }%
4102 %

```

`\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

```

4103 \newcommand{\printXafternumber}[1]{%
4104   \iftoggle{Xnonbreakableafternumber@#1}{\nobreak}{}%
4105   \hspace{\csuse{Xafternumber@#1}}%
4106 }%
4107 %

```

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. 105](#): 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 was no boolean for the line number because it was always printed.

Maïeul Rouquette has added:

- `\ifl@d@Xtwolines` and `\ifl@d@Xmorethantwolines` to print a symbol which stands for “and subsequent” when there are two, three or more lines;
- `\@annot@start@print` and `\@annot@end@print` which define annotations associated with the starting and ending line numbers in critical footnotes;
- `\ifl@d@Xlinenumber`, which value depends on the configuration of `\Xnolinenumber` and `\Xnolinenumberifannotation`.

```

\ifl@d@pnum08 \newif\ifl@d@pnum
\ifl@d@ssub09 \newif\ifl@d@ssub
\ifl@d@elin10 \newif\ifl@d@elin
\ifl@d@esl11 \newif\ifl@d@esl
\ifl@d@dash12 \newif\ifl@d@dash
\ifl@d@Xtwolines13 \newif\ifl@d@Xtwolines%
\ifl@d@Xmorethantwolines14 \newif\ifl@d@Xmorethantwolines%
\ifl@d@Xlinenumber15 \let\@annot@start@print\relax%
\@annot@start@print16 \let\@annot@end@print\relax%
\@annot@end@print17 \newif\ifl@d@Xlinenumber%
\@annot@end@print18 %

```

```

\l@dp@rsefootsspec \l@dp@rsefootsspec{<spec>}{<lemma>}{<text>} parses a footnote specification. <lemma>
\l@dp@rsefootsspec and <text> are the lemma and text respectively. <spec> is the line and page num-
\l@dp@rsefootsspec ber and lemma font specifier in \l@d@nums style format. The real work is done by
\l@dp@rsefootsspec \l@dp@rsefootsspec which defines macros holding the numeric values. In many cases,
\l@dp@rsefootsspec this last command is called directly. Just a reminder of the arguments:
\l@dp@rsefootsspec \printlines #1 | #2 | #3 | #4 | #5 | #6 | #7
\l@dp@rsefootsspec \printlines start-page | line | subline | end-page | line | subline | fontflag
\l@dp@rsefootsspec \newcommand*{\l@dp@rsefootsspec}[3]{\l@dp@rsefootsspec#1|}
\l@dp@rsefootsspec \def\l@dp@rsefootsspec#1|#2|#3|#4|#5|#6|#7|{%
\l@dp@rsefootsspec \gdef\l@dp@rsefootsspecstartpage{#1}%
\l@dp@rsefootsspec \gdef\l@dp@rsefootsspecstartline{#2}%
\l@dp@rsefootsspec \gdef\l@dp@rsefootsspecstartsub{#3}%
\l@dp@rsefootsspec \gdef\l@dp@rsefootsspecendpage{#4}%
\l@dp@rsefootsspec \gdef\l@dp@rsefootsspecendline{#5}%
\l@dp@rsefootsspec \gdef\l@dp@rsefootsspecendsub{#6}%
\l@dp@rsefootsspec }
\l@dp@rsefootsspec %

```

Initialise the several number value macros.

```

\l@dp@rsefootsspec \def\l@dp@rsefootsspecstartpage{0}%
\l@dp@rsefootsspec \def\l@dp@rsefootsspecstartline{0}%
\l@dp@rsefootsspec \def\l@dp@rsefootsspecstartsub{0}%
\l@dp@rsefootsspec \def\l@dp@rsefootsspecendpage{0}%
\l@dp@rsefootsspec \def\l@dp@rsefootsspecendline{0}%
\l@dp@rsefootsspec \def\l@dp@rsefootsspecendsub{0}%
\l@dp@rsefootsspec %

```

`\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`.

```

\l@dp@rsefootsspec \newcommand*\setprintlines[6]{%
\l@dp@rsefootsspec \let\@annot@start@print\relax%
\l@dp@rsefootsspec \let\@annot@end@print\relax%
\l@dp@rsefootsspec \l@d@pnumfalse%
\l@dp@rsefootsspec \l@d@dashfalse%
\l@dp@rsefootsspec \l@d@elinfalse%
\l@dp@rsefootsspec %

```

Should we print the line number? It depends on hooks and annotations.

```

4144 \l@d@Xlinenumbertrue%
4145 \ifboolexpr{togl{Xnolinenumber@\@currentseries}%
4146   or (%
4147     togl{Xnolinenumberifannotation@\@currentseries}%
4148     and not test{\ifdefvoid{\annot@start}}}%
4149     and not test{\ifdefvoid{\annot@end}}}%
4150   )%
4151 }{%
4152   \l@d@Xlinenumberfalse%
4153 }{%
4154 %

```

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.

```

4155 \ifbypage@
4156   \ifnum#4=#1 \else
4157     \l@d@pnumtrue
4158     \l@d@dashtrue
4159   \fi
4160 \fi
4161 %

```

We print the ending line number if:

1. we print the ending page number;
2. or the ending line number is different from the starting line number and we didn't disable the line number printing

If either of these conditions is true, we also print the annotation linked to the ending line number annotations.

```

4162 \ifboolexpr{%
4163   bool{\l@d@pnum}%
4164   or (%
4165     not test{\ifnumequal{#2}{#5}}%
4166     and bool{\l@d@Xlinenumber}%
4167   )%
4168 }{%
4169   \l@d@elintrue%
4170   \l@d@dashtrue%
4171   \unless\ifx\relax\annot@end%
4172   \def\@annot@end@print{%
4173     \l@wrappcs@ifnotemptybox{Xwraplinenumannotation@\@currentseries}{\
annot@end}%
4174   }%
4175   \fi%
4176 }{%
4177 %

```

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.

```

4178 \ifl@d@elin%
4179   \def\@annot@start@print{%

```

```

4180 \l@wrapcs@ifnotemptybox{Xwraplinenumannotation@\@currentseries}{\
annot@start}%
4181 }%
4182 \else%
4183 \ifx\annot@start\annot@end%
4184 \unless\ifx\@annot@start\relax%
4185 \def\@annot@start@print{%
4186 \l@wrapcs@ifnotemptybox{Xwraplinenumannotation@\@currentseries}{\
annot@start}%
4187 }%
4188 \fi%
4189 \else%
4190 \ifx\@annot@end@print\relax%
4191 \def\@annot@start@print{%
4192 \l@wrapcs@ifnotemptybox{Xwraplinenumannotation@\@currentseries}{%
4193 \ifx\annot@start\empty\else%
4194 \annot@start%
4195 \ifdefined\linrangesep@%
4196 \linrangesep@%
4197 \else%
4198 \csuse{Xlinrangeseparator@\@currentseries}%
4199 \fi%
4200 \fi%
4201 \annot@end%
4202 }%
4203 }%
4204 \else%
4205 \let\@annot@start@print\@annot@end@print%
4206 \let\@annot@end@print\relax%
4207 \fi%
4208 \fi%
4209 \fi%
4210 %

```

We print the starting sub-line if it is nonzero.

```

4211 \l@d@ssubfalse
4212 \ifnum#3=0 \else
4213 \l@d@ssubtrue
4214 \fi
4215 %

```

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.

```

4216 \l@d@eslfalse
4217 \ifnum#6=0 \else
4218 \ifnum#6=#3
4219 \ifl@d@elin \l@d@esltrue \else \l@d@eslfalse \fi
4220 \else
4221 \l@d@esltrue
4222 \l@d@dashtrue
4223 \fi
4224 \fi%
4225 %

```

However, if the `\Xtwolines` is set for the current series, we do not print the last line number.

```

4226 \ifl@dash%
4227 \ifboolexpr{togl{fulllines@} or test{\ifcsempy{Xtwolines@
@currentseries}}}%
4228 {}%
4229 {%
4230 \setistwofollowinglines{#1}{#2}{#4}{#5}%
4231 \ifboolexpr{%
4232 (%
4233 togl {Xtwolinesbutnotmore@\@currentseries}%
4234 and not%
4235 (%
4236 bool {istwofollowinglines@}%
4237 )%
4238 )%
4239 or%
4240 (%
4241 (not test{\ifnumequal{#1}{#4}})%
4242 and togl{Xtwolinesonlyinsamepage@\@currentseries}%
4243 )%
4244 }%
4245 {}%
4246 {%
4247 \l@dashfalse%
4248 \l@Xtwolinestrue%
4249 \l@elinfalse%
4250 \l@eslfalse%
4251 \ifcsempy{Xmorethantwolines@\@currentseries}%
4252 {}%
4253 {\ifistwofollowinglines@else%
4254 \l@Xmorethantwolinestrue%
4255 \fi%
4256 }%
4257 }%
4258 }%
4259 \fi%
4260 %

```

If the `\Xnoidenticallinenumannotation` is set for this series, we check if the ending annotation is identical to the starting. If true, we don't print the ending annotation

```

4261 \iftoggle{Xnoidenticallinenumannotation@\@currentseries}{%
4262 \ifx\annot@start\annot@end%
4263 \let\@annot@end@print\relax%
4264 \ifboolexpr{%Do not print the dash if we do not print the line number
4265 test {\ifdefempty{\linenumrep}}%
4266 or not bool {l@d@Xlinenum}}%
4267 }{%
4268 \l@dashfalse%
4269 }{}%
4270 \fi%
4271 }{}%
4272 %

```



Finally, we check for `\Xlinenumannotationonlyfirst` and `\Xlinenumannotationonlyfirstintwo`, and we redefine, if required, `\@annot@start@print` and `\annot@end@print`. We also store the current line number annotations.

```

4273 \iftoggle{Xlinenumannotationonlyfirst@\@currentseries}{%
4274   \ifboolexpr{%
4275     (%
4276       togl{Xlinenumannotationonlyfirstintwo@\@currentseries}%
4277       and test{\ifcsequal{annot@start}{prevannot@start@\@currentseries}}%
4278       and test{\ifcsequal{annot@end}{prevannot@end@\@currentseries}}%
4279     )%
4280     or%
4281     (%
4282       not togl{Xlinenumannotationonlyfirstintwo@\@currentseries}%
4283       test{\ifcsequal{annot@start}{prevannot@start@\@currentseries}}%
4284     )%
4285   }{%
4286     \def\@annot@start@print{%
4287       \l@wrapcs@ifnotemptybox{Xwraplinenumannotation@\@currentseries}{\
csuse{Xsymlinenumannotation@\@currentseries}}%
4288     }%
4289     \let\@annot@end@print\relax%
4290     \ifboolexpr{%Do not print the dash if we do not print the line number
4291       test {\ifdefempty{\linenumrep}}%
4292       or not bool {l@d@Xlinenum}}%
4293     }{%
4294       \l@d@d@dashfalse%
4295     }{%
4296   }{%
4297     \global\cslet{prevannot@start@\@currentseries}{\annot@start}%
4298     \global\cslet{prevannot@end@\@currentseries}{\annot@end}%
4299   }%
4300   }%
4301 %

End of \setprintlines.

4302 }%
4303 %

```

`\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.

```

4304 \newif\ifistwofollowinglines@%
4305 \newcommand{\setistwofollowinglines}[4]{%
4306   \ifcsdef{lastlinenumberon@#1}%
4307     {\numdef{\tmp}{\csuse{lastlinenumberon@#1}}}%
4308     {\numdef{\tmp}{0}}%
4309   \istwofollowinglines@false%
4310   \ifnumequal{#4-#2}{1}%
4311     {\istwofollowinglines@true}%%
4312   {\ifbypage@%
4313     \ifnumequal{#3-#1}{1}%
4314     {%
4315       \ifnumequal{#2}{\tmp}%
4316       {\ifnumequal{#4}{1}{\istwofollowinglines@true}{}}%
4317       {}}%
4318     }%
4319     {}}%
4320   \fi%
4321 }%
4322 }%
4323 %

```

`\printlines` So, we have decided which part of line number sets will be printed depending on these values. 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

```

4324 \newcount\printlines@count%
4325 \def\printlines#1|#2|#3|#4|#5|#6|#7|#8|{%
4326   \global\advance\printlines@count by 1%
4327   \begingroup%
4328 %

```

Decide which part of line number components we will print.

```

4329   \setprintlines{#1}{#2}{#3}{#4}{#5}{#6}%
4330 %

```

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.

```

4331   \ifdimequal{\csuse{Xboxstartlinenum@\@currentseries}}{0pt}%
4332     {\bgroup}%
4333     {\leavevmode\hbox to \csuse{Xboxstartlinenum@\@currentseries}\bgroup\
4334 hfill}%
4335   \ifcsstring{Xlinenumannotationposition@\@currentseries}{before}%
4336     {\@annot@start@print}%
4337 %

```

Sometime, we need to print the page number. However, if both the current page is equal to the page number and we use `\Xnopagenumberifcurrent`, we do not want to print it. To determine the current page, we need to use `crossref` mechanism. See <https://tex.stackexchange.com/a/239457/7712> for explanation.

```

4338 \ifl@d@pnum%
4339 \iftoggle{Xnopagenumberifcurrent@\@currentseries}{%
4340 \edlabel{printlines@\the\printlines@count}%
4341 }{ }%
4342 \ifboolexpr{%
4343 not test{\ifnumequal{\xpageref{printlines@\the\printlines@count
}}{#1}}%
4344 or not togl{Xnopagenumberifcurrent@\@currentseries}%
4345 }%
4346 {%
4347 \csuse{Xbeforepagenumber@\@currentseries}%
4348 \wrap@edcrossref{\@this@crossref@start}{%
4349 #1%
4350 }%
4351 \ifl@d@Xlinenumber%
4352 \csuse{Xpagelinesep@\@currentseries}%
4353 \fi%
4354 }{ }%
4355 \fi%
4356 \ifl@d@Xlinenumber%
4357 \wrap@edcrossref{\@this@crossref@start}{%
4358 \linenumrep{#2}%
4359 \iftoggle{Xlineflag@\@currentseries}{#8}{ }%
4360 }%
4361 \iftoggle{Xnotboxingsubline@\@currentseries}{\egroup}{ }%Close startline
box
4362 \ifl@d@ssub%
4363 \csuse{Xsublinesep@\@currentseries}%
4364 \wrap@edcrossref{\@this@crossref@start}{\sublinenumrep{#3}}%
4365 \fi%
4366 \fi%
4367 \ifcsstring{Xlinenumannotationposition@\@currentseries}{after}%
4368 {\@cannot@start@print}%
4369 }{ }%
4370 \iftoggle{Xnotboxingsubline@\@currentseries}{\egroup}{ }%Close startline
box
4371 %

```

Then print the dash + end line number, or the range symbol.

```

4372 \ifdimequal{\csuse{Xboxendlinenumber@\@currentseries}}{0pt}%
4373 {\bgroup}%
4374 {\hbox to \csuse{Xboxendlinenumber@\@currentseries}\bgroup}%
4375 \ifl@d@Xtwolines%
4376 \ifl@d@Xlinenumber%
4377 \ifl@d@Xmorethantwolines%
4378 \wrap@edcrossref{\@this@crossref@end}{\csuse{Xmorethantwolines@\@currentseries}}%
4379 \else%
4380 \wrap@edcrossref{\@this@crossref@end}{\csuse{Xtwolines@\@currentseries}}%

```

```

4381 \fi%
4382 \fi%
4383 \else%
4384 \ifl@d@dash%
4385 \ifdefined\linangesep%
4386 \linangesep%
4387 \else%
4388 \csuse{Xlinangeseparator@\@currentseries}%
4389 \fi%
4390 \fi%
4391 \ifcsstring{Xlinenumannotationposition@\@currentseries}{before}%
4392 {\@annot@end@print}%
4393 {}%
4394 \ifl@d@pnum%
4395 \ifboolexpr{%
4396 not test{\ifnumequal{\xpageref{printlines@\the\printlines@count
}}{#4}}%
4397 or not togl{Xnopagenumberifcurrent@\@currentseries}%
4398 }{%
4399 \csuse{Xbeforepagenumber@\@currentseries}%
4400 \wrap@edcrossref{\@this@crossref@end}{%
4401 #4%
4402 }%
4403 \csuse{Xpagelinesep@\@currentseries}%
4404 }}%
4405 \fi%
4406 \ifl@d@Xlinenum%
4407 \ifl@d@elin%
4408 \wrap@edcrossref{\@this@crossref@end}{%
4409 \linenumrep{#5}%
4410 \iftoggle{Xlineflag@\@currentseries}{#8}{}%
4411 }%
4412 \fi%
4413 \ifl@d@esl%
4414 \ifl@d@elin%
4415 \csuse{Xsublinesep@\@currentseries}%
4416 \fi%
4417 \wrap@edcrossref{\@this@crossref@end}{\sublinenumrep{#6}}%
4418 \fi%
4419 \fi%
4420 \ifcsstring{Xlinenumannotationposition@\@currentseries}{after}%
4421 {\@annot@end@print}%
4422 {}%
4423 \fi%
4424 \ifdimequal{\csuse{Xboxendlinum@\@currentseries}}{0pt}%
4425 {}%
4426 {\hfill}%Prevent underfull hbox
4427 \egroup%
4428 \endgroup%
4429 }%
4430 %

```

## XIII.9.5 Footnote grouped by line

`\prepare@Xgroupbyline` `\prepare@Xgroupbyline` is a macro called on the `\metaXXXvfootnote` if `\Xgroupbyline` is set to true, instead of calling `\insert` directly. #1 The series  
 #2 The content of the footnote, which is also the line number indication  
 #3 This macro, which prepares the contents of the insertion

```

4431 \newcommand{\prepare@Xgroupbyline}[3]{%
4432   \iftoggle{Xgroupbylineseparetwolines@#1}{%
4433     \l@dparsespec#2%
4434     \ifcsdef{#1@forinserting@l@dparseendpage-l@dparseendline-l@
l@dparseendsub}%
4435       {%
4436         \csgappto{#1@forinserting@l@dparseendpage-l@dparseendline-l@
l@dparseendsub}{%
4437           \hskip\csuse{Xafternote@#1}\relax%
4438         }%
4439       }%
4440     }%
4441     \add@hooktoggle@specific@to@cs{#1@forinserting@l@dparseendpage-l@
l@dparseendline-l@dparseendsub}%
4442     \add@hookarg@specific@to@cs{#1@forinserting@l@dparseendpage-l@
l@dparseendline-l@dparseendsub}%
4443     \csxappto%
4444       {#1@forinserting@l@dparseendpage-l@dparseendline-l@dparseendsub
}%
4445     {%
4446       \keep@this@crossref@forinserting%
4447       \unexpanded{%
4448         \ifcempty{Xsymlinenum@#1}%
4449           {\csuse{Xparafootsep@#1}}%
4450         }%
4451         #3{#1}{#2}%
4452       }%
4453     }%
4454     \listcsxadd{#1@forinserting}{l@dparseendpage-l@dparseendline-l@
l@dparseendsub}%
4455   }{%
4456     \ifcsdef{#1@forinserting@all}{%
4457       \csgappto%
4458         {#1@forinserting@all}%
4459         {\hskip\csuse{Xafternote@#1}\relax}%
4460       }{%
4461         \add@hooktoggle@specific@to@cs{#1@forinserting@all}%
4462         \add@hookarg@specific@to@cs{#1@forinserting@all}%
4463         \csxappto%
4464           {#1@forinserting@all}%
4465         {%
4466           \keep@this@crossref@forinserting%
4467           \unexpanded{%
4468             \ifcempty{Xsymlinenum@#1}%
4469               {\csuse{Xparafootsep@#1}}%
4470             }%
4471             #3{#1}{#2}%

```

```

4472 }%
4473 }%
4474 }%
4475 \listcsgadd{#1@forinserting}{all}%
4476 }%
4477 %

```

`\keep@this@crossref@forinserting` `\keep@this@crossref@forinserting` is called by `\prepare@Xgroupbyline`. It just ensures that `\@this@crossref@start` and `\this@crossref@end` value are kept in memory.

```

4478 \newcommand{\keep@this@crossref@forinserting}{%
4479   \unexpanded{\gdef\@this@crossref@start}{\@this@crossref@start}%
4480   \unexpanded{\gdef\@this@crossref@end}{\@this@crossref@end}%
4481 }%
4482 %

```

## XIV Familiar footnotes

### XIV.1 Adjacent footnotes

The original EDMAC provided users with five series of critical footnotes (`\Afootnote` `\Bfootnote` `\Cfootnote` `\Dfootnote` `\Efootnote`), and  $\text{\LaTeX}$  provides a single numbered footnote. The `reledmac` package uses the EDMAC mechanism to provide six series of numbered footnotes.

First, though, the `footmisc` package has an option whereby two or more consecutive `\footnotes` have their marks separated by commas. This seemed to Peter Wilson such a useful ability that it was provided automatically by `eledmac`.

Maïeul Rouquette has maintained this feature in `reledmac`, despite he thought that is not directly in relationship with the aim of `reledmac`.

`\multiplefootnotemarker` `\multiplefootnotemarker` These macros may have been defined by the memoir class, are provided by the `footmisc` package and perhaps by other footnote packages. That is why we use `\providecommand` and not `\newcommand`.

```

4483 \providecommand*\multiplefootnotemarker}{3sp}
4484 \providecommand*\multfootsep}{\textsuperscript{\normalfont,}}
4485
4486 %

```

`\m@mmf@prepare` A pair of self-cancelling kerns. This may have been defined in the memoir class.

```

4487 \providecommand*\m@mmf@prepare}{%
4488   \kern-\multiplefootnotemarker
4489   \kern\multiplefootnotemarker\relax}
4490 %

```

`\m@mmf@check` This may have been defined in the memoir class. If it recognises the last kern as `\multiplefootnotemarker` it typesets `\multfootsep`.

```

4491 \providecommand*\m@mmf@check}{%
4492   \ifdim\lastkern=\multiplefootnotemark\relax
4493     \edef\@x@sf{\the\spacefactor}%
4494     \unkern
4495     \multfootsep
4496     \spacefactor\@x@sf\relax
4497   \fi}
4498
4499 %

```

We have to modify `\@footnotetext` and `\@footnotemark`. However, if `memoir` is used the modifications have already been made.

```

4500 \@ifclassloaded{memoir}{}{%
4501 %

```

`\@footnotetext` Add `\m@mmf@prepare` at the end of `\@footnotetext`.

```

4502 \apptocmd{\@footnotetext}{\m@mmf@prepare}{}{}
4503 %

```

`\@footnotemark` Modify `\@footnotemark` to cater for adjacent footnotes.

```

4504
4505 \patchcmd{\@footnotemark}
4506   {\nobreak}
4507   {\m@mmf@check
4508    \nobreak
4509   }
4510   {}{}
4511 \patchcmd{\@footnotemark}
4512   {\@makefnmark}
4513   {\@makefnmark
4514    \m@mmf@prepare
4515   }
4516   {}{}
4517 %

```

Finished the modifications for the non-memoir case.

```

4518 }
4519
4520 %

```

## 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.

```

4521 \pretocmd{\@footnotetext}{%
4522   \ifnumberedpar@
4523   \edtext{}{}%

```

```

4524     \advance \edtext@notes by 1%
4525     \l@dbfnote{#1}%
4526   }%
4527   \else
4528   {}{}
4529 \apptocmd{\@footnotetext}{\fi}{}{}%
4530 %

```

`\l@dbfnote` `\l@dbfnote` adds the footnote to the insert list, and `\vl@dbfnote` calls the original  
`\vl@dbfnote` `\@footnotetext`. We also patch `\footnote` in order to get the correct footnote  
`\vl@dbfnote` numbers when typesetting parallel texts. This is moved into a `\get@fnmark` command.  
`\footnote`

```

\get@fnmark
\get@thisfootnote
4531 \patchcmd%
4532   {\footnote}%
4533   {\stepcounter\@mpfn}%
4534   {%
4535   \ifboolexpr{bool{l@dpairing} or bool{l@dprintingpages} or bool{
4536   l@dprintingcolumns}}{%
4537     \global\advance\footnote@reading by \@ne%
4538     \get@thisfootnote%
4539     \get@fnmark{\thisc@footnote}%
4540     \ifcsdef{footnotereading\the\footnote@reading=typeset}%
4541     {\setcounter{\@mpfn}{\csuse{footnotereading\the\footnote@reading=
typeset}}}%
4542     {\setcounter{\@mpfn}{\footnote@reading}}}%
4543   }-%
4544   \stepcounter\@mpfn%
4545   }%
4546   }%
4547   {}
4548   {}
4549
4550 \newcommand{\get@thisfootnote}{%
4551   \ifboolexpr{bool{l@dpairing} or bool{l@dprintingpages} or bool{
l@dprintingcolumns}}{%
4552     \protected@xdef\thisc@footnote{\the\footnote@reading}%
4553   }-%
4554   \protected@xdef\thisc@footnote{\the\c@footnote}%
4555   }%
4556 }%
4557
4558 \newcommand{\l@dbfnote}[1]{%
4559   \get@thisfootnote%
4600   \gdef\@tag{#1\relax}%
4601   \ifledRcol%
4602     \xright@appenditem{%
4603       \ifdefined\Hy@footnote@currentHref%
4604         \noexpand\def\noexpand\Hy@footnote@currentHref{\
Hy@footnote@currentHref}%
4605       \fi%
4606       \noexpand\vl@dbfnote{{\expandonce\@tag}}{\thisc@footnote}%
4607     }%

```



```

4568         \to\inserts@listR
4569         \global\advance\insert@countR \@ne%
4570     \else%
4571         \xright@appenditem{%
4572             \ifdefined\Hy@footnote@currentHref%
4573             \noexpand\def\noexpand\Hy@footnote@currentHref{\
Hy@footnote@currentHref}%
4574             \fi%
4575             \noexpand\vl@dbfnote{\expandonce\@tag}{\thisc@footnote}%
4576         }%
4577         \to\inserts@list
4578         \global\advance\insert@count \@ne%
4579     \fi
4580     \ignorespaces%
4581 }%
4582
4583 \newcommand{\get@fnmark}[1]{%
4584     \ifboolexpr{bool{!@dpairing} or bool{!@dprintingpages} or bool{
!@dprintingcolumns}}%
4585     {%
4586         \stepcounter{footnote@typeset}%
4587         \setcounter{footnote}{\c@footnote@typeset}%
4588         \immediate\write\@mainaux{%
4589             \csgdef{footnotereading#1=typeset}{\the\c@footnote@typeset}%
4590         }%
4591         \def\@thefnmark{\thefootnote}%
4592     }%
4593     {%
4594         \setcounter{footnote}{#1}%
4595         \def\@thefnmark{\thefootnote}%
4596     }%
4597 }%
4598
4599 \newcommand{\vl@dbfnote}[2]{%
4600     \get@fnmark{#2}%
4601     \@footnotetext{#1}%
4602 }%
4603 %

```

### 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 `\dots@footnotemark\dots` macros.  
`\postbodyfootmark`

```

4604 \newcommand*\prebodyfootmark{%
4605     \leavevmode
4606     \ifhmode
4607         \edef\@x@sff{\the\spacefactor}%
4608         \m@mff@check
4609         \nobreak
4610     \fi}
4611 \newcommand*\postbodyfootmark{%

```

```

4612 \m@mmf@prepare
4613 \ifhmode\spacefactor\@x@sf\fi\relax}
4614
4615 %

```

## XIV.4 Footnote arrangement

### XIV.4.1 User level macro

`\arrangementX` `\arrangementX[ $\langle s \rangle$ ]{ $\langle arrangement \rangle$ }` command calls, for each series, a specific command which set many counters and commands in order to define specific arrangement.

```

4616 \newcommandx{\arrangementX}[2][1,usedefault]{%
4617   \def\do##1{%
4618     \csname arrangementX@#2\endcsname{##1}%
4619   }%
4620   \ifstrempy{##1}%
4621     {%
4622     \dolistloop{\@series}%
4623     }%
4624     {
4625     \docsvlist{##1}%
4626     }%
4627 }%
4628 %

```

### XIV.4.2 Normal footnotes

`\normal@footnotemarkX` `\normal@footnotemarkX{ $\langle series \rangle$ }` sets up the typesetting of the marker at the point where the footnote is called for.

```

4629 \newcommand*{\normal@footnotemarkX}[1]{%
4630   \prebodyfootmark
4631   \wrapped@bodyfootmarkX{##1}%
4632   \postbodyfootmark}
4633
4634 %

```

`\normalbodyfootmarkX` The `\normalbodyfootmarkX{ $\langle series \rangle$ }` *really* typesets the in-text marker. The style is the normal superscript.

```

4635 \newcommand*{\normalbodyfootmarkX}[1]{%
4636   \hbox{\textsuperscript{\normalfont\@nameuse{@thefnmark#1}}}}
4637 %

```

`\normalvfootnoteX` `\normalvfootnoteX{ $\langle series \rangle$ }{ $\langle text \rangle$ }` does the `\insert` for the  $\langle series \rangle$  and calls the series' `\footfmt...` to format the  $\langle text \rangle$ .

```

4638 \notbool{parapparatus@}{\newcommand*{\newcommand}{\normalvfootnoteX}[2]{%
4639   \beforeinsertion@X{##1}%
4640   \insert\@nameuse{footins#1}\bgroup
4641     \fontseries{\seriesdefault}%
4642     \fontshape{\shapedefault}%
4643     \hsize=\expandafter\dimexpr\csuse{widthX@#1}\relax%

```

```

4644 \noindent\csuse{bhooknoteX@#1}%
4645 \csuse{notefontsizeX@#1}%
4646 \footsplitskips
4647 \ifl@dpairing\ifl@dpaging\else%
4648   \setnoteswidthliketwocolumnsX@{#1}%
4649 \fi\fi%
4650 \setnotesXpositionliketwocolumns@{#1}%
4651 \spaceskip=\z@skip \xspaceskip=\z@skip
4652 \csuse{\csuse{footnote@dir}}\@nameuse{footfmt#1}{#1}{#2}\egroup}
4653
4654 %

```

`\mpnormalvfootnoteX` The minipage version.

```

4655 \newcommand*{\mpnormalvfootnoteX}[3]{%
4656   \get@thisfootnoteX{#1}%
4657   \get@fnmarkX{#1}{\thisc@footnote}{#3}%
4658   \ifstrempy{#3}{%
4659     \edef\this@footnoteX@reading{\the\csname footnote#1@reading\endcsname}%
4660   }{%
4661     \edef\this@footnoteX@reading{###3}%
4662   }%
4663   \global\setbox\@nameuse{mpfootins#1}\vbox{%
4664     \unvbox\@nameuse{mpfootins#1}
4665     \noindent\csuse{bhooknoteX@#1}%
4666     \csuse{notefontsizeX@#1}%
4667     \hsize\columnwidth
4668     \@parboxrestore
4669     \color@begingroup
4670     \@nameuse{footfmt#1}{#1}{#2}\color@endgroup}}
4671
4672 %

```

`\normalfootfmtX` `\normalfootfmtX{<series>}{<text>}` typesets the footnote text, prepended by the marker.

```

4673 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\normalfootfmtX}[2]{%
4674   \ifluatex%
4675     \textdir\footnote@luatextextdir%
4676     \pardir\footnote@luatexpardir%
4677   \fi%
4678   \protected@edef\@currentlabel{%
4679     \@nameuse{@thefnmark#1}%
4680   }%
4681   \ledsetnormalparstuffX{#1}%
4682   \sethangindentX{#1}%
4683   \rule\z@\splittopskip%
4684   {{\csuse{notenumberfontX@#1}\wrapped@footfootmarkX{#1}}%
4685     \csuse{wrapcontentX@#1}{#2}%
4686   \strut\par}}
4687
4688 %

```

`\normalfootfootmarkX` `\normalfootfootmarkX{<series>}` is called by `\normalfootfmtX` to typeset the footnote marker in the footer before the footnote text.

```

4689 \newcommand*{\normalfootfootmarkX}[1]{%
4690   \textsuperscript{\@nameuse{@thefnmark#1}}
4691 }
4692 %

```

**\normalfootstartX** `\normalfootstartX{<series>}` is the `<series>` footnote starting macro used in the output routine.

```

4693 \newcommand*{\normalfootstartX}[1]{%
4694   \ifdimequal{Opt}{\prenotesX@}{%
4695     {%
4696       \iftoggle{prenotesX@}{%
4697         \togglefalse{prenotesX@}%
4698         \skip\csname footins#1\endcsname=%
4699         \glueexpr\csuse{prenotesX@}+\csuse{afterruleX@#1}\relax%
4700       }%
4701     }%
4702   }%
4703   \vskip\skip\csname footins#1\endcsname%
4704   \leftskip=\z@
4705   \rightskip=\z@
4706   \ifl@dpairing\else%
4707     \hsize=\old@hsize%
4708   \fi%
4709   \setnoteswidthliketwocolumnsX@{#1}%
4710   \setnotesXpositionliketwocolumns@{#1}%
4711   \print@footnoteXrule{#1}%
4712 }%
4713 %
4714 %

```

**\normalfootnoteruleX** The rule drawn before the footnote series group.

```

4715 \let\normalfootnoteruleX=\footnoterule
4716 %
4717 %

```

**\normalfootgroupX** `\normalfootgroupX{<series>}` sends the contents of the `<series>` insert box to the output page without alteration.

```

4718 \newcommand*{\normalfootgroupX}[1]{%
4719   \csuse{bhookgroupX@#1}%
4720   \unvbox\@nameuse{footins#1}%
4721   \hsize=\old@hsize%
4722 }%
4723 %
4724 %

```

**\mpnormalfootgroupX** The minipage version.

```

4725 \newcommand*{\mpnormalfootgroupX}[1]{%
4726   \vskip\skip\@nameuse{mpfootins#1}
4727   \ifl@dpairing\ifparledgroup%
4728     \leavevmode\marks\parledgroup@{begin}%

```

```

4729 \marks\parledgroup@series{#1}%
4730 \marks\parledgroup@type{footnoteX}%
4731 \fi\fi\normalcolor
4732 \ifparledgroup%
4733 \ifl@dpairing%
4734 \else%
4735 \setnoteswidthliketwocolumnsX@{#1}%
4736 \setnotesXpositionliketwocolumns@{#1}%
4737 \print@footnoteXrule{#1}%
4738 \fi%
4739 \else%
4740 \setnoteswidthliketwocolumnsX@{#1}%
4741 \setnotesXpositionliketwocolumns@{#1}%
4742 \print@footnoteXrule{#1}%
4743 \fi%
4744 \csuse{bhookgroupX@#1}%
4745 \unvbox\@nameuse{mpfootins#1}}
4746
4747 %

```

`\normalbfnoteX` #1 = footnote series, #2 = footnote content, #3 manual footnote number

```

4748
4749 \newcommand{\normalbfnoteX}[3]{%
4750 \get@thisfootnoteX{#1}%
4751 \ifledRcol%
4752 \ifluatex
4753 \footnotelang@lua[R]%
4754 \fi
4755 \@ifundefined{xpg@main@language}%if polyglossia
4756 {}%
4757 {\footnotelang@poly[R]}%
4758 \xright@appenditem{%
4759 \noexpand\led@set@index@fornote{#1}%
4760 \noexpand\prepare@edindex@fornote{\led@nums}%
4761 \unexpanded{\gdef\@currentnototype{familiar}}%
4762 \unexpanded{\def\this@footnoteX@reading}{\the\csname footnote#1
@reading\endcsname}%
4763 \noexpand\vbfnoteX{#1}{#2}{\thisc@footnote}{#3}%
4764 \noexpand\led@reinit@index@fornote%
4765 \unexpanded{\advance\@edindex@fornote@\m@ne}%
4766 }%
4767 \to\inserts@listR
4768 \global\advance\insert@countR \@ne%
4769 \else%
4770 \ifluatex
4771 \footnotelang@lua%
4772 \fi
4773 \@ifundefined{xpg@main@language}%if polyglossia
4774 {}%
4775 {\footnotelang@poly}%
4776 \xright@appenditem{%
4777 \noexpand\led@set@index@fornote{#1}%
4778 \noexpand\prepare@edindex@fornote{\led@nums}%

```

```

4779     \unexpanded{\gdef\@currentnotetype{familiar}}%
4780     \unexpanded{\def\this@footnoteX@reading{\the\csname footnote#1
@reading\endcsname}}%
4781     \noexpand\vbfnoteX{#1}{#2}{\this@footnote}{#3}%
4782     \noexpand\led@reinit@index@fornote%
4783     \unexpanded{\advance\@edindex@fornote@\m@ne}%
4784     }%
4785         \to\inserts@list
4786     \global\advance\insert@count \@ne%
4787     \fi
4788     \ignorespaces}
4789
4790 %

```

`\get@thisfootnoteX` The macro `\get@thisfootnote` command just saves the footnote number in the `\thisfootnote` macro, depending on the use of pairing environments.

```

4791 \newcommand{\get@thisfootnoteX}[1]{%
4792   \ifboolexpr{bool{l@dpairing} or bool{l@dprintingpages} or bool{
l@dprintingcolumns}}{%
4793     \protected@xdef\this@footnote{\the\csname footnote#1@reading\
endcsname}}%
4794   }{%
4795     \protected@xdef\this@footnote{\the\csname c@footnote#1\endcsname}}%
4796   }%
4797 }%
4798 %

```

`\vbfnoteX` This command calls the correct footnote-inserting commands. #1 footnote series, #2 footnote content, #3 footnote counter, #4 manual footnote mark

```

4799 \newcommand{\vbfnoteX}[4]{%
4800   \get@fnmarkX{#1}{#3}{#4}\relax%
4801   \@nameuse{regvfootnote#1}{#1}{#2}%
4802 }%
4803 %

```

`\get@fnmarkX` This command gets the correct footnote number when typesetting parallel texts. #1 footnote series, #2 footnote counter, #3 manual footnote number. It also gets, if required, a hyperref (almost) unique identifier for the footnote.

```

4804 \newcommand{\get@fnmarkX}[3]{%
4805   \ifstrempy{#3}{%
4806     \ifboolexpr{bool{l@dpairing} or bool{l@dprintingpages} or bool{
l@dprintingcolumns}}{%
4807       {%
4808         \stepcounter{footnote#1@typeset}%
4809         \setcounter{footnote#1}{\value{footnote#1@typeset}}%
4810         \immediate\write\@mainaux{%
4811           \csgdef{footnote#1reading#2=typeset}{\the\csname c@footnote#1
@typeset\endcsname}}%
4812         }%
4813       }%
4814     }%

```

```

4815     \setcounter{footnote#1}{#2}%
4816   }%
4817   \csdef{@thefnmark#1}{\csuse{thefootnote#1}}%
4818   \csxdef{hr@thefnmark#1}{\the\value{footnote#1}@the\csname footnote#1
@series\endcsname}%
4819 }%
4820 {%
4821   \csdef{@thefnmark#1}{#3}%
4822   \csxdef{hr@thefnmark#1}{#3@the\csname footnote#1@series\endcsname}%
4823 }%
4824 }%
4825 %
4826 %

```

`\vnumfootnoteX` #1 footnote series, #2 footnote content, #3 manual footnote mark

```

4827
4828 \newcommand{\vnumfootnoteX}[3]{%
4829   \ifnumberedpar@
4830     \edtext{}{%
4831       \advance\edtext@notes by 1%
4832       \normalbfnoteX{#1}{#2}{#3}%
4833     }%
4834   \else
4835     \def\this@footnoteX@reading{\the\csname footnote#1@reading\endcsname}%
4836     \get@thisfootnoteX{#1}%
4837     \get@fnmarkX{#1}{\expandonce\thisc@footnote}{#3}%
4838     \@nameuse{regvfootnote#1}{#1}{#2}%
4839   \fi}
4840 %

```

`arrangementX@normal` `\arrangementX@normal{<series>}` initialises the settings for the `<series>` footnotes. This should always be called for each series.

```

4841 \newcommand*{\arrangementX@normal}[1]{%
4842   \csgdef{series@displayX#1}{normal}
4843   \expandafter\let\csname footstart#1\endcsname=\normalfootstartX
4844   \@namedef{@footnotemark#1}{\normal@footnotemarkX{#1}}
4845   \@namedef{bodyfootmark#1}{\normalbodyfootmarkX{#1}}
4846   \expandafter\let\csname regvfootnote#1\endcsname=\normalvfootnoteX
4847   \expandafter\let\csname vfootnote#1\endcsname=\vnumfootnoteX
4848   \expandafter\let\csname footfmt#1\endcsname=\normalfootfmtX
4849   \@namedef{footfootmark#1}{\normalfootfootmarkX{#1}}
4850   \expandafter\let\csname footgroup#1\endcsname=\normalfootgroupX
4851   \expandafter\let\csname footnoterule#1\endcsname=\normalfootnoteruleX
4852   \count\csname footins#1\endcsname=1000
4853   \csxdef{default@footins#1}{1000}%Use to have note only for one side
4854   \dimen\csname footins#1\endcsname=\csuse{maxhnotesX@#1}
4855   \skip\csname footins#1\endcsname=\csuse{beforenotesX@#1}%
4856   \advance\skip\csname footins#1\endcsname by\csuse{afterruleX@#1}%
4857 %

```

Additions for minipages.

```

4858   \ifnoledgroup@ \else%

```

```

4859 \expandafter\let\csname mpvfootnote#1\endcsname=\mpnormalvfootnoteX
4860 \expandafter\let\csname mpfootgroup#1\endcsname=\mpnormalfootgroupX
4861 \count\csname mpfootins#1\endcsname=1000
4862 \dimen\csname mpfootins#1\endcsname=\csuse{maxhnotesX@#1}
4863 \skip\csname mpfootins#1\endcsname=\csuse{beforenotesX@#1}%
4864 \advance\skip\csname mpfootins#1\endcsname by\csuse{afterruleX@#1}%
4865 \fi
4866 }
4867
4868 %

```

#### 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 4869 \newcommand*\arrangementX@twocol}[1]{%
4870 \csgdef{series@displayX#1}{twocol}
4871 \expandafter\let\csname regvfootnote#1\endcsname=\twocolvfootnoteX
4872 \expandafter\let\csname footfmt#1\endcsname=\twocolfootfmtX
4873 \expandafter\let\csname footgroup#1\endcsname=\twocolfootgroupX
4874 \dimen\csname footins#1\endcsname=\csuse{maxhnotesX@#1}%
4875 \skip\csname footins#1\endcsname=\csuse{beforenotesX@#1}%
4876 \advance\skip\csname footins#1\endcsname by \csuse{afterruleX@#1}\relax%
4877 \twocolfootsetupX{#1}
4878 \ifnoledgroup@else%
4879 \expandafter\let\csname mpvfootnote#1\endcsname=\mpnormalvfootnoteX
4880 \expandafter\let\csname mpfootgroup#1\endcsname=\mptwocolfootgroupX
4881 \skip\csname mpfootins#1\endcsname=\csuse{beforenotesX@#1}%
4882 \advance\skip\csname mpfootins#1\endcsname by\csuse{afterruleX@#1}
4883 \mptwocolfootsetupX{#1}
4884 \fi%
4885 }
4886
4887 %

```

```

\twocolfootsetupX \twocolfootsetupX{<series>}
\mptwocolfootsetupX
4888 \newcommand*\twocolfootsetupX}[1]{%
4889 \count\csname footins#1\endcsname 500
4890 \csxdef{default@footins#1}{500}%Use this to confine the notes to one
side only
4891 \multiply\dimen\csname footins#1\endcsname by \tw@}
4892 \newcommand*\mptwocolfootsetupX}[1]{%
4893 \count\csname mpfootins#1\endcsname 500
4894 \multiply\dimen\csname mpfootins#1\endcsname by \tw@}
4895
4896 %

```

```

\twocolvfootnoteX \twocolvfootnoteX{<series>}
4897 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\twocolvfootnoteX}[2]{%
4898 \beforeinsertion@X{#1}%

```



```

4899 \let\ bidi@RTL@everypar\relax%
4900 \insert\csname footins#1\endcsname\bgroup%
4901 \hsize=\expandafter\dimexpr\csuse{widthX@#1}\relax%
4902 \noindent\csuse{bhooknoteX@#1}%
4903 \csuse{notefontsizeX@#1}%
4904 \footsplitskips%
4905 \spaceskip=\z@skip \xspaceskip=\z@skip%
4906 \@nameuse{footfmt#1}{#1}{#2}\egroup}
4907
4908 %

```

`\twocolfootfmtX` `\twocolfootfmtX{<series>}`

```

4909 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\twocolfootfmtX}[2]{%
4910 \protected@edef\@currentlabel{%
4911 \@nameuse{@thefnmark#1}%
4912 }%
4913 \normal@pars%
4914 \sethangindentX{#1}%
4915 \hsize \csuse{hsizetwocolX@#1}%
4916 \setparindentX{#1}%
4917 \tolerance=5000\relax%
4918 \par%
4919 \@tempdima=\parindent%
4920 \csuse{colalignX@#1}%
4921 \parindent=\@tempdima%
4922 {\hspace{\parindent}%
4923 \csuse{notenunfontX@#1}\wrapped@footfootmarkX{#1}\strut%
4924 \csuse{wrapcontentX@#1}{#2}%
4925 \strut\par}%
4926 \allowbreak%
4927 }%
4928
4929 %

```

`\twocolfootgroupX` `\twocolfootgroupX{<series>}`

`\mptwocolfootgroupX`

```

4930 \newcommand*{\twocolfootgroupX}[1]{\csuse{bhookgroupX@#1}\csuse{
notefontsizeX@#1}
4931 \splittopskip=\ht\strutbox
4932 \expandafter
4933 \rigidbalanceX\csname footins#1\endcsname \tw@ \splittopskip}}
4934
4935 \newcommand*{\mptwocolfootgroupX}[1]{%
4936 \vskip\skip\@nameuse{mpfootins#1}
4937 \ifl@dpairing\ifparledgroup%
4938 \leavevmode\marks\parledgroup@{begin}%
4939 \marks\parledgroup@series{#1}%
4940 \marks\parledgroup@type{footnoteX}%
4941 \fi\fi\normalcolor
4942 \ifparledgroup%
4943 \ifl@dpairing%
4944 \else%
4945 \setnoteswidthliketwocolumnsX@{#1}%

```

```

4946 \setnotesXpositionliketwocolumns@{#1}%
4947 \print@footnoteXrule{#1}%
4948 \fi%
4949 \else%
4950 \setnoteswidthliketwocolumnsX@{#1}%
4951 \setnotesXpositionliketwocolumns@{#1}%
4952 \print@footnoteXrule{#1}%
4953 \fi%
4954 \csuse{bhookgroupX@#1}%
4955 \splittopskip=\ht\strutbox
4956 \expandafter
4957 \rigidbalanceX\csname mpfootins#1\endcsname \tw@ \splittopskip}}
4958
4959 %

```

#### 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  \newcommand*{\arrangementX@threecol}[1]{%
4961 \csgdef{series@displayX#1}{threecol}
4962 \expandafter\let\csname regvfootnote#1\endcsname=\threecolvfootnoteX
4963 \expandafter\let\csname footfmt#1\endcsname=\threecolfootfmtX
4964 \expandafter\let\csname footgroup#1\endcsname=\threecolfootgroupX
4965 \dimen\csname footins#1\endcsname=\csuse{maxhnotesX@#1}%
4966 \skip\csname footins#1\endcsname=\csuse{beforenotesX@#1}%
4967 \advance\skip\csname footins#1\endcsname by \csuse{afterruleX@#1}\relax%
4968 \threecolfootsetupX{#1}
4969 \ifnoledgroup@\else%
4970 \expandafter\let\csname mpvfootnote#1\endcsname=\mpnormalvfootnoteX
4971 \expandafter\let\csname mpfootgroup#1\endcsname=\mpthreecolfootgroupX
4972 \skip\csname mpfootins#1\endcsname=\csuse{beforenotesX@#1}%
4973 \advance\skip\csname mpfootins#1\endcsname by \csuse{afterruleX@#1}
4974 \mpthreecolfootsetupX{#1}
4975 \fi%
4976 }
4977
4978 %

```

`\threecolfootsetupX` `\threecolfootsetupX{<series>}`

`\mpthreecolfootsetupX`

```

4979 \newcommand*{\threecolfootsetupX}[1]{%
4980 \count\csname footins#1\endcsname 333
4981 \csxdef{default@footins#1}{333}%Use this to confine the notes to one
side only
4982 \multiply\dimen\csname footins#1\endcsname by \thr@@}
4983 \newcommand*{\mpthreecolfootsetupX}[1]{%
4984 \count\csname mpfootins#1\endcsname 333
4985 \multiply\dimen\csname mpfootins#1\endcsname by \thr@@}
4986
4987 %

```

```

\threecolvfootnoteX \threecolvfootnoteX{<series>}{<text>}
4988 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\threecolvfootnoteX}[2]{
%
4989 \let\bidir@RTL@everypar\relax%
4990 \beforeinsertion@X{#1}%
4991 \insert\csname footins#1\endcsname\bgroup%
4992 \hsize=\expandafter\dimexpr\csuse{widthX@#1}\relax%
4993 \noindent\csuse{bhooknoteX@#1}%
4994 \csuse{notefontsizeX@#1}%
4995 \footsplitskips%
4996 \@nameuse{footfmt#1}{#1}{#2}\egroup}
4997
4998 %

```

```

\threecolfootfmtX \threecolfootfmtX{<series>}
4999 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\threecolfootfmtX}[2]{%
5000 \protected@edef\@currentlabel{%
5001 \nameuse{@thefnmark#1}%
5002 }%
5003 \sethangindentX{#1}%
5004 \normal@pars%
5005 \hsize \csuse{hsizethreecolX@#1}%
5006 \setparindentX{#1}%
5007 \tolerance=5000\relax%
5008 \@tempdima=\parindent%
5009 \csuse{colalignX@#1}%
5010 \parindent=\@tempdima%
5011 {\hspace{\parindent}%
5012 \csuse{notenumfontX@#1}\wrapped@footfootmarkX{#1}\strut%
5013 \csuse{wrapcontentX@#1}{#2}%
5014 \strut\par}\allowbreak}
5015
5016 %

```

```

\threecolfootgroupX \threecolfootgroupX{<series>}
\mpthreecolfootgroupX
5017 \newcommand*{\threecolfootgroupX}[1]{\csuse{bhookgroupX@#1}\csuse{
notefontsizeX@#1}
5018 \splittopskip=\ht\strutbox
5019 \expandafter
5020 \rigidbalanceX\csname footins#1\endcsname \thr@@ \splittopskip}}
5021
5022 \newcommand*{\mpthreecolfootgroupX}[1]{%
5023 \vskip\skip\@nameuse{mpfootins#1}
5024 \ifl@dpairing\ifparledgroup
5025 \leavevmode\marks\parledgroup@{begin}%
5026 \marks\parledgroup@series{#1}%
5027 \marks\parledgroup@type{footnoteX}%
5028 \fi\fi\normalcolor
5029 \ifparledgroup%
5030 \ifl@dpairing%
5031 \else%
5032 \setnoteswidthliketwocolumnsX@{#1}%

```

```

5033 \setnotesXpositionliketwocolumns@{#1}%
5034 \print@footnoteXrule{#1}%
5035 \fi%
5036 \else%
5037 \setnoteswidthliketwocolumnsX@{#1}%
5038 \setnotesXpositionliketwocolumns@{#1}%
5039 \print@footnoteXrule{#1}%
5040 \fi%
5041 \csuse{bhookgroupX@#1}%
5042 \splittopskip=\ht\strutbox
5043 \expandafter
5044 \rigidbalanceX\csname mpfootins#1\endcsname \thr@@ \splittopskip}}
5045
5046 %

```

#### XIV.4.5 Paraphed footnotes

The following macros set footnotes as one paragraph.

`\arrangementX@paragraph` `\footparagraphX{series}`

```

5047 \newcommand*{\arrangementX@paragraph}[1]{%
5048 \csgdef{series@displayX#1}{paragraph}%
5049 \expandafter\let\csname footstart#1\endcsname=\parafootstartX
5050 \expandafter\let\csname regvfootnote#1\endcsname=\para@vfootnoteX
5051 \expandafter\let\csname footfmt#1\endcsname=\parafootfmtX
5052 \expandafter\let\csname footgroup#1\endcsname=\para@footgroupX
5053 \expandafter\let\csname footnoterule#1\endcsname=\normalfootnoteruleX
5054 \count\csname footins#1\endcsname=1000
5055 \csxdef{default@footins#1}{1000}%Use this to confine the notes to one
side only
5056 \dimen\csname footins#1\endcsname=\csuse{maxhnotesX@#1}
5057 \skip\csname footins#1\endcsname=\csuse{beforenotesX@#1}%
5058 \advance\skip\csname footins#1\endcsname by\csuse{afterruleX@#1}%
5059 \para@footsetupX{#1}
5060 \ifnoledgroup@\else
5061 \expandafter\let\csname mpvfootnote#1\endcsname=\mppara@vfootnoteX
5062 \expandafter\let\csname mpfootgroup#1\endcsname=\mppara@footgroupX
5063 \count\csname mpfootins#1\endcsname=1000
5064 \dimen\csname mpfootins#1\endcsname=\csuse{maxhnotesX@#1}
5065 \skip\csname mpfootins#1\endcsname=\csuse{beforenotesX@#1}%
5066 \advance\skip\csname mpfootins#1\endcsname by\csuse{afterruleX@#1}%
5067 \fi
5068 }
5069
5070 %

```

`\para@footsetupX` `\para@footsetupX{series}`

```

5071 \newcommand*{\para@footsetupX}[1]{\csuse{bhookgroupX@#1}\csuse{
notefontsizeX@#1}
5072 \setnoteswidthliketwocolumnsX@{#1}%
5073 \ifcempty{widthX@#1}%
5074 {}%

```

```

5075     {\columnwidth=\expandafter\dimexpr\csuse{widthX@#1}\relax}%
5076     \dimen0=\baselineskip
5077     \multiply\dimen0 by 1024
5078     \divide\dimen0 by \columnwidth \multiply\dimen0 by \footfudgefiddle\relax
5079     %
5080     \expandafter
5081     \xdef\csname footfudgefactor#1\endcsname{%
5082     \expandafter\strip@pt\dimen0 }}}
5083     %

```

`\parafootstartX` `\parafootstartX{<series>}`

```

5084 \newcommand*\parafootstartX[1]{%
5085   \ifdimequal{Opt}{\prenotesX@}{}%
5086   {%
5087     \iftoggle{prenotesX@}{%
5088       \togglefalse{prenotesX@}%
5089       \skip\csname footins#1\endcsname=%
5090       \glueexpr\csuse{prenotesX@}+\csuse{afterruleX@#1}\relax%
5091     }%
5092   }%
5093 }%
5094 \leftskip=\z@
5095 \rightskip=\z@
5096 \setparindentX{#1}%
5097 \vskip\skip\@nameuse{footins#1}%
5098 \setnoteswidthliketwocolumnsX@{#1}%
5099 \setnotesXpositionliketwocolumns@{#1}%
5100 \print@footnoteXrule{#1}%
5101 }
5102 %
5103 %

```

`\para@vfootnoteX` `\para@vfootnoteX{<series>}{<text>}`

`\mppara@vfootnoteX`

```

5104 \newcommand*\para@vfootnoteX[2]{%
5105   \csuse{beforeinsertingX@#1}%
5106   \insert\csname footins#1\endcsname%
5107   \bgroup
5108     \csuse{notefontsizeX@#1}
5109     \footsplitskips
5110     \setbox0=\vbox{\hsize=\maxdimen%
5111       \let\bidir@RTL@everypar\@empty%
5112       \insert@txtbeforenotesX{#1}%
5113       \noindent\csuse{bhooknoteX@#1}%
5114       \@nameuse{footfmt#1}{#1}{#2}}%
5115     \setbox0=\hbox{\unvXH{0}{#1}}%
5116     \dp0=\z@
5117     \ht0=\csname footfudgefactor#1\endcsname\wd0
5118     \box0
5119     \penalty0
5120   \egroup}
5121 \newcommand*\mppara@vfootnoteX[3]{%

```

```

5122 \get@thisfootnoteX{#1}%
5123 \get@fnmarkX{#1}{\thisc@footnote}{#3}%
5124 \edef\this@footnoteX@reading{\the\csname footnote#1@reading\endcsname}%
5125 \global\setbox\@nameuse{mpfootins#1}\vbox{%
5126   \unvbox\@nameuse{mpfootins#1}
5127   \csuse{notefontsizeX@#1}
5128   \footsplitskips
5129   \setbox0=\vbox{\hsize=\maxdimen%
5130     \let\bidirectional\everypar\@empty%
5131     \noindent\color@begingroup%
5132     \csuse{bhooknoteX@#1}%
5133     \@nameuse{footfmt#1}{#1}{#2}\color@endgroup}%
5134   \setbox0=\hbox{\unvxhX{0}{#1}}%
5135   \dp0=\z@
5136   \ht0=\csname footfudgefactor#1\endcsname\wd0
5137   \box0
5138   \penalty0}}
5139 %
5140 %

```

```

\unvxhX41 \newcommand*\unvxhX[2]{% 2th is optional for retro-compatibility
5142   \setbox0=\vbox{\unvbox#1%
5143     \global\setbox1=\lastbox}%
5144   \unhbox1
5145   \unskip           % remove \rightskip,
5146   \unskip           % remove \parfillskip,
5147   \unpenalty       % remove \penalty of 10000,
5148   \hskip\csuse{afternoteX@#2}%
5149   \relax}% but add the glue to go between the notes
5150 %
5151 %

```

`\parafootfmtX` `\parafootfmtX{<series>}`

```

5152 \newcommand*\parafootfmtX[2]{%
5153   \protected@edef\@currentlabel{%
5154     \@nameuse{@thefnmark#1}%
5155   }%
5156   \insertparafootsepX{#1}%
5157   \ledsetnormalparstuff@common%
5158   {\csuse{notenumfontX@#1}%
5159     \csuse{notenumfontX@#1}%
5160     \wrapped@footfootmarkX{#1}%
5161     \strut%
5162     \csuse{wrapcontentX@#1}{#2}%
5163     \penalty-10}}
5164 %
5165 %

```

`\para@footgroupX` `\para@footgroupX{<series>}`  
`\mppara@footgroupX`

```

5166 \newcommand*\para@footgroupX[1]{%
5167   \hsize=\expandafter\dimexpr\csuse{widthX@#1}\relax%

```

```

5168 \unvbox\csname footins#1\endcsname
5169 \ifcsstring{raggedX@#1}{L}{\RaggedLeft}{}%
5170 \ifcsstring{raggedX@#1}{R}{\RaggedRight}{}%
5171 \makehboxofhboxes
5172 \setbox0=\hbox{\unhbox0 \removehboxes}%
5173 \csuse{bhookgroupX@#1}
5174 \csuse{notefontsizeX@#1}
5175 \unhbox0\par}
5176
5177 \newcommand*\mppara@footgroupX}[1]{%
5178 \setnoteswidthliketwocolumnsX@{#1}%
5179 \vskip\skip\@nameuse{mpfootins#1}
5180 \ifl@dpairing\ifparledgroup
5181 \leavevmode%
5182 \leavevmode\marks\parledgroup@{begin}%
5183 \marks\parledgroup@series{#1}%
5184 \marks\parledgroup@type{footnoteX}%
5185 \fi\fi\normalcolor
5186 \ifparledgroup%
5187 \ifl@dpairing%
5188 \else%
5189 \setnoteswidthliketwocolumnsX@{#1}%
5190 \setnotesXpositionliketwocolumns@{#1}%
5191 \print@footnoteXrule{#1}%
5192 \fi%
5193 \else%
5194 \setnoteswidthliketwocolumnsX@{#1}%
5195 \setnotesXpositionliketwocolumns@{#1}%
5196 \print@footnoteXrule{#1}%
5197 \fi%
5198 \unvbox\csname mpfootins#1\endcsname
5199 \ifcsstring{raggedX@#1}{L}{\RaggedLeft}{}%
5200 \ifcsstring{raggedX@#1}{R}{\RaggedRight}{}%
5201 \makehboxofhboxes
5202 \setbox0=\hbox{\unhbox0 \removehboxes}%
5203 \csuse{bhookgroupX@#1}%
5204 \csuse{notefontsizeX@#1}%
5205 \setparindentX{#1}%
5206 \unhbox0\par}}
5207
5208 %

```

**Insertion of the footnotes separator** The command `\insertparafootsepX{<series>}` must be called at the beginning of `\parafootftmX`.

```

\insertparafootsepX09 \newcommand{\insertparafootsepX}[1]{%
5210 \ifledRcol%
5211 \ifnumequal{\csuse{prevpage#1@numR}}{\page@numR}%
5212 {\csuse{parafootsepX@#1}}%
5213 {}%
5214 \global\csname prevpage#1@numR\endcsname=\page@numR%
5215 \else%

```

```

5216 \ifboolexpr{bool{!@dprintingpages} or bool{!@dprintingcolumns} or bool
{numbering}}{%
5217 \ifnumequal{\csuse{prevpage#1@num}}{\page@num}%
5218 {\csuse{parafootsepX@#1}}%
5219 {}%
5220 \global\csname prevpage#1@num\endcsname=\page@num%
5221 }{%We can't use directly the page counter, see https://tex.
stackexchange.com/a/239457/7712
5222 \global\expandafter\advance\csname footnote#1@parafootsepX\endcsname
by 1%
5223 \edef\@tmpa{\parafootsepX@#1\the\csname footnote#1@parafootsepX\
endcsname}%
5224 \expandafter\edlabel\expandafter{\@tmpa}%
5225 \edef\@tmpb{\expandafter\xpageref\expandafter{\@tmpa}}%
5226 \IfStrEq{\csuse{prevpage#1@num@outnumbering}}{\@tmpb}%
5227 {\csuse{parafootsepX@#1}}%
5228 {}%
5229 \global\cslet{prevpage#1@num@outnumbering}{\@tmpb}%
5230 }%
5231 \fi%
5232 }
5233 %

```

#### 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.

```

5234 \newcommand{\wrapped@footfootmarkX}[1]{%
5235 \ifdefined\hypertarget%
5236 \hyperlink%
5237 {@bodyfootmark#1\csuse{hr@thefnmark#1}}%
5238 {\@nameuse{footfootmark#1}}%
5239 \Hy@raisedlink{%
5240 \hypertarget%
5241 {@footnotemark#1\csuse{hr@thefnmark#1}}%
5242 {}%
5243 }%
5244 \else%
5245 \@nameuse{footfootmark#1}%
5246 \fi%
5247 }%
5248 %

```

`\wrapped@bodyfootmarkX` `\wrapped@bodyfootmarkX` prints the footnote mark of the text body, wrapped in `hyperref` package's commands, if needed.

```

5249 \newcommand{\wrapped@bodyfootmarkX}[1]{%
5250 \ifdefined\hypertarget%
5251 \hyperlink%
5252 {@footnotemark#1\csuse{hr@thefnmark#1}}%
5253 {\@nameuse{bodyfootmark#1}}%
5254 \Hy@raisedlink{%
5255 \hypertarget%

```



```

5256     {bodyfootmark#1@csuse{hr@thefnmark#1}}%
5257     {}%
5258     }%
5259     \else%
5260     \@nameuse{bodyfootmark#1}%
5261     \fi%
5262 }%
5263 %

```

## 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 $\TeX$ .

That why we have defined `\ledsetnormalparstuff@common` in order to make this setting for both familiar and critical notes. This command is called by command to make specific setting to critical or familiar footnote.

```

\ledsetnormalparstuff@common \newcommand*\ledsetnormalparstuff@common{%
\Xledsetnormalparstuff      \ifluatex%
\ledsetnormalparstuffX     \ifdefstring{\footnote@luatextextdir}{TLT}{}%
                             {\textdir\footnote@luatextextdir}%
5267                             \pardir\footnote@luatexpardir%
5268                             \fi%
5269                             \csuse{\csuse{footnote@dir}}%
5270                             \normal@pars%
5271                             \parfillskip \z@ \@plus 1fil}%
5272
5273
5274 \newcommand*\Xledsetnormalparstuff}[1]{%
5275     \ledsetnormalparstuff@common%
5276     \Xsetparindent{#1}%
5277     \iftoggle{Xparindent@#1}{\hspace{\parindent}}{}%
5278 }%
5279
5280 \newcommand*\ledsetnormalparstuffX}[1]{%
5281     \ledsetnormalparstuff@common%
5282     \setparindentX{#1}%
5283     \iftoggle{parindentX@#1}{\hspace{\parindent}}{}%
5284 }%
5285 %

```

## 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.

```

5286
5287 \newdimen\old@hsize%
5288 \AtBeginDocument{\old@hsize=\hsize}%
5289
5290 \newcommand{\setXnoteswidthliketwocolumns@}[1]{%
5291   \global\let\hsize@fornote=\hsize%
5292   \global\old@hsize=\hsize%
5293   \let\old@columnwidth=\columnwidth%
5294   \iftoggle{Xnoteswidthliketwocolumns@#1}%
5295     {%
5296       \setwidthliketwocolumns%
5297       \global\let\hsize@fornote=\hsize%
5298     }%
5299   {%
5300     \let\hsize=\hsize@fornote%
5301     \let\columnwidth=\old@columnwidth%
5302   }%
5303
5304 \newcommand{\setnoteswidthliketwocolumnsX@}[1]{%
5305   \global\let\hsize@fornote=\hsize%
5306   \global\old@hsize=\hsize%
5307   \let\old@columnwidth=\columnwidth%
5308   \iftoggle{noteswidthliketwocolumnsX@#1}%
5309     {%
5310       \setwidthliketwocolumns%
5311       \global\let\hsize@fornote=\hsize%
5312     }%
5313   {%
5314     \let\hsize=\hsize@fornote%
5315     \let\columnwidth=\old@columnwidth%
5316   }%
5317
5318 %

```

`\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`.

```

5319 \newcommand{\setXnotespositionliketwocolumns@}[1]{%
5320   \iftoggle{Xnoteswidthliketwocolumns@#1}{%
5321     \csuse{setnotespositionliketwocolumns@\columns@position}%
5322   }{}%
5323 }%
5324
5325 \newcommand{\setnotesXpositionliketwocolumns@}[1]{%
5326   \iftoggle{noteswidthliketwocolumnsX@#1}{%
5327     \csuse{setnotespositionliketwocolumns@\columns@position}%
5328   }{}%
5329 }%

```

```
5330
5331 %
```

## XVII Footnotes' order

`\fnpos` and `\mpfnpos` simply place their arguments in `\@fnpos` and `\@mpfnpos`, which will be used later in the output routine.

```
\fnpos
\mpfnpos
\@fnpos
\@mpfnpos
5332 \def\@fnpos{familiar-critical}
5333 \def\@mpfnpos{critical-familiar}
5334 \newcommand{\fnpos}[1]{\xdef\@fnpos{#1}}
5335 \newcommand{\mpfnpos}[1]{\xdef\@mpfnpos{#1}}
5336 %
```

## 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@Xfootnoterule37 \newcommand{\print@Xfootnoterule}[1]{%
\print@footnoterule38 \vskip-\csuse{Xafterrule@#1}%Because count in \dimen\csuse{#1footins}
5339 \nointerlineskip%
5340 \moveleft-\leftskip\vbox{\csuse{#1footnoterule}}%
5341 \nointerlineskip%
5342 \vskip\csuse{Xafterrule@#1}%
5343 }%
5344
5345 \newcommand{\print@footnoterule}[1]{%
5346 \vskip-\csuse{afterruleX@#1}%Because count in \dimen\csuse{footins#1}
5347 \nointerlineskip%
5348 \moveleft-\leftskip\vbox{\csuse{footnoterule#1}}%
5349 \nointerlineskip%
5350 \vskip\csuse{afterruleX@#1}%
5351 }%
5352
5353 %
```

## 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.

```

5354 \newtoggle{Xprenotes@}%
5355 \toggletrue{Xprenotes@}%
5356 \newcommand{\Xprenotes@}{0pt}%
5357 \newcommand*{\Xprenotes}[1]{\renewcommand{\Xprenotes@}{#1}}%
5358 \newcommand{\preXnotes}[1]{\led@warning@preXnotes@deprecated\Xprenotes{#1}}
%For compatibility
5359 %

```

The same, but for familiar footnotes.

```

\Xprenotes@60 \newtoggle{prenotesX@}
\Xprenotes@61 \toggletrue{prenotesX@}
5362 \newcommand{\XprenotesX@}{0pt}
5363 \newcommand*{\XprenotesX}[1]{\renewcommand{\XprenotesX@}{#1}}
5364 %

```

## XIX.3 Internal commands

```

firstXseries@65 \gdef\firstXseries@{}
prepare@Xprenotes@66 \newcommand{\prepare@Xprenotes}[1]{%
5367 \ifdimequal{0pt}{\Xprenotes@}%
5368 }%
5369 {%
5370 \IfStrEq{\firstXseries@}{-}{%
5371 \global\skip\csuse{#1footins}=\Xprenotes@%

```

```

5372 \global\advance\skip\csname #1footins\endcsname by\csuse{Xafterrule@
#1}%
5373 \gdef\firstXseries@{#1}%
5374 }%
5375 {%
5376 \ifseriesbefore{#1}{\firstXseries@}%
5377 {%
5378 \global\skip\csuse{#1footins}=\csuse{Xbeforenotes@\firstXseries@}%
5379 \global\advance\skip\csname #1footins\endcsname by\csuse{Xafterrule@
#1}%
5380 \gdef\firstXseries@{#1}%
5381 }%
5382 }%
5383 }%
5384 }%
5385 }
5386 %

```

The same thing is required for familiar notes and \prenotesX.

```

firstseriesX@57 \gdef\firstseriesX@{}
prepare@prenotesX58 \newcommand{\prepare@prenotesX}[1]{%
5389 \ifdimequal{0pt}{\prenotesX@}%
5390 }%
5391 {%
5392 \IfStrEq{\firstseriesX@}{}{%
5393 \global\skip\csuse{footins#1}=\prenotesX@%
5394 \global\advance\skip\csname footins#1\endcsname by\csuse{afterruleX@
#1}%
5395 \gdef\firstseriesX@{#1}%
5396 }%
5397 {%
5398 \ifseriesbefore{#1}{\firstseriesX@}%
5399 {%
5400 \global\skip\csuse{footins#1}=\csuse{beforenotesX@\firstseriesX@}%
5401 \global\advance\skip\csname footins#1\endcsname by\csuse{afterruleX@
#1}%
5402 \gdef\firstseriesX@{#1}%
5403 }%
5404 }%
5405 }%
5406 }%
5407 }
5408 %

```

## XX Endnotes

First, check the noend option.

```

5409 \ifbool{noend@}{}{%Used instead of \ifnoend@ to prevent expansion problem
5410 %

```

## 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.

```

5411 \newcommand{\l@dend@open}[1]{%
5412   \global\booltrue{l@dend@#1}%
5413   \expandafter\immediate%
5414     \expandafter\openout%
5415       \csname l@d@#1end\endcsname%
5416       =\l@auxdir\jobname.#1end\relax%
5417 }%
5418 \newcommand{\l@dend@close}[1]{%
5419   \global\boolfalse{l@dend@#1}%
5420   \expandafter\immediate%
5421     \expandafter\closeout\csname l@d@#1end\endcsname%
5422 }%
5423 %
5424 %

```

`\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.

```

5425 \newcommand{\l@dend@stuff}{%
5426   \def\do##1{%
5427     \ifbool{l@dend@##1}{%
5428       {\l@dend@open{##1}}%
5429       \expandafter\immediate\expandafter\write\csname l@d@##1end\endcsname{\
string\l@d@section{the\section@num}\@percentchar}%
5430     }%
5431     \dolistloop{\@series}%
5432   }%
5433 %
5434 %

```

`\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.

```

5435 \global\notbool{parapparus@}{\long}\def\endprint#1#2#3#4#5#6#7{%
5436   \iftoggle{Xendfirstnote@#4}{%
5437     \global\togglefalse{Xendfirstnote@#4}%
5438     \csuse{Xendtxtbeforenotes@#4}%
5439   }{%
5440     \csuse{Xendbhooknote@#4}%
5441     \csuse{Xendnotefontsize@#4}%
5442     \hangindent=\csuse{Xendhangindent@#4}%
5443     \Xendstorelineinfo{#1}{#4}%
5444     \ifXendinsertsep@%
5445       \hskip\csuse{Xendafternote@#4}\relax%
5446       \ifcdef{preendl#4}%
5447         {\ifcsequal{preendl#4}{lineinfo@}%
5448           {\ifcempty{Xendsymnum@#4}{\csuse{Xendsep@#4}}{}}%
5449           {\csuse{Xendsep@#4}}%
5450         }%
5451         {\csuse{Xendsep@#4}}%
5452     \else%
5453       \iftoggle{Xendparagraph@#4}%
5454         {\global\Xendinsertsep@true}%
5455         {}%
5456     \fi%
5457     \xdef\@currentseries{#4}%
5458     \def\do##1{%
5459       \setkeys[mac]{truefootnoteoption}{##1}%
5460     }%
5461     \notblank{#5}{\docsvlist{#5}}{%
5462       \IfStrEq{#6}{R}{\ledRcol@true}{}%
5463       \def\@this@crossref@start{#7:start}%
5464       \def\@this@crossref@end{#7:end}%
5465       \printlineendnote{#1}{#4}%
5466       \IfStrEq{#6}{R}{\ledRcol@false}{}%
5467       \nottoggle{Xendlemmadisablefontselection@#4}%
5468         {\select@lemmafnt#1}{}%
5469         {}%
5470     \bgroup%
5471       \csuse{Xendlemmafnt@#4}%
5472       \csuse{Xendwraplemma@#4}{#2}%
5473     \egroup%
5474     \ifboolexpr{%
5475       togl {nosep@}%
5476       or test{\ifcempty{Xendlemmaseparator@#4}}%
5477     }{%
5478       {\hskip\csuse{Xendinplaceoflemmaseparator@#4}\relax}%
5479       {\nobreak%
5480         \hskip\csuse{Xendbeforelemmaseparator@#4}%
5481         \csuse{Xendlemmaseparator@#4}%
5482         \hskip\csuse{Xendafterlemmaseparator@#4}%
5483         \relax%
5484       }%

```

```

5485 \csuse{Xendwrapcontent@#4}{#3}%
5486 \nottoggle{Xendparagraph@#4}{\par}{}%
5487 \def\do##1{%
5488   \setkeys[mac]{falsefootnoteoption}{##1}%
5489 }%
5490 \notblank{#5}{\docsvlist{#5}}{}%
5491 }}%
5492
5493 \let\l@d@section=\@gobble
5494
5495 %

```

**\Xendstorelineinfo** The `\Xendstorelineinfo` macro is used to store some data about line number of the current critical endnote, data which will be reused later for the `\Xnumberonlyfirstinline` and related setting.

#1 endnote specification for the current endnote; #2 endnote series.

```

5496 \newcommand{\Xendstorelineinfo}[2]{%
5497   \l@dp@rsefootspec#1|{%
5498     \iftoggle{Xendnumberonlyfirstintwolines@#2}{%
5499       \xdef\lineinfo@{\l@d@parsedstartpage - \l@d@parsedstartline - \
5500       \l@d@parsedstartsub - \l@d@parsedendpage - \l@d@parsedendline - \
5501       \l@d@parsedendsub}%
5502     }%
5503     {\
5504       \xdef\lineinfo@{\l@d@parsedstartpage - \l@d@parsedstartline - \
5505       \l@d@parsedstartsub}%
5506     }%
5507   }%
5508 }%
5509 %

```

**\printlineendnote** This macro controls, in endnote, whether the line number is printed or not, according to the series options. Its first argument is the information about lines; its second is the series of the footnote.

```

5506 \newcommand{\printlineendnote}[2]{%
5507   \ifboolexpr{%
5508     togl {nonum@}%
5509     or togl {Xendnonumber@#2}%
5510   }%
5511   {\hspace{\csuse{Xendinplaceofnumber@#2}}}%
5512   {%
5513     \iftoggle{Xendnumberonlyfirstinline@#2}%
5514     {\ifcsdef{prevendline#2}%
5515       {\ifcsequal{prevendline#2}{lineinfo@}%
5516         {%
5517           \csuse{Xendbhookinplaceofnumber@#2}%
5518           \ifcseempty{Xendsymmlinenumber@#2}%
5519             {\hspace{\csuse{Xendinplaceofnumber@#2}}}%
5520             {\printsymmlinenumber@#2}%
5521           \csuse{Xendahookinplaceofnumber@#2}%
5522         }%
5523       }%
5524     }%
5525   }%

```



```

5525     }%
5526     {\printlineendnotearea{#1}{#2}}%We keep every time line
5527     \csxdef{prevendline#2}{\lineinfo@#1}%
5528     }%
5529 }%
5530 %

```

```

\printsymlineendnotearea \newcommand{\printsymlineendnotearea}[1]{%
5532   \hspace{\csuse{Xendbeforesymlinewidth@#1}}%
5533   \csuse{Xendnotenumfont@#1}%
5534   \ifdimequal{\csuse{Xendboxsymlinewidth@#1}}{\z@}%
5535     {\csuse{Xendsymlinewidth@#1}}%
5536     {\hbox to \csuse{Xendboxsymlinewidth@#1}}%
5537     {\csuse{Xendsymlinewidth@#1}\hfill}%
5538   }%
5539   \hspace{\csuse{Xendaftersymlinewidth@#1}}%
5540 }%
5541 %

```

`\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 on the options about repeating line numbers. The first argument is line information, the second is the notes series (A, B, C, etc.)

```

5542 \newcommand{\printlineendnotearea}[2]{%
5543   \csuse{Xendbhooklinenumber@#2}%
5544   \hspace{\csuse{Xendbeforenumber@#2}}%
5545   \bgroup%
5546     \csuse{Xendnotenumfont@#2}%
5547     \ifdimequal{\csuse{Xendboxlinenum@#2}}{Opt}%
5548       {\printendlines#1||\ifledRcol@{\Rlineflag}\fi}%
5549       {\leavevmode%
5550         \hbox to \csuse{Xendboxlinenum@#2}}%
5551       {%
5552         \IfSubStr{RC}{\csuse{Xendboxlinenumalign@#2}}{\hfill}{}%
5553         \printendlines#1||\ifledRcol@{\Rlineflag}\fi%
5554         \IfSubStr{LC}{\csuse{Xendboxlinenumalign@#2}}{\hfill}{}%
5555       }%
5556   \egroup%
5557   \hspace{\csuse{Xendafternumber@#2}}%
5558   \csuse{Xendahooklinenumber@#2}%
5559 }%
5560 %

```

## XX.2 User level commands

### XX.2.1 Inserting contents to endnotes

The `\Xendnotes` commands are defined above, when defining apparatus commands by series. Here, we define only `\toendnotes` command not specific to a series, in order to insert arbitrary code. The regular version writes an unexpanded argument, while the regular version writes a once-expanded argument.

```

\toendnotes* \newcommandx{\toendnotes}[2][1,usedefault]{%
\toendnotes* \ifboolexpr{bool{numbering} or bool{numberingR}}{%
5563   \def\do##1{%
5564     \expandafter\immediate\expandafter\write\csname l@d@##1end\endcsname%
5565     {\unexpanded{#2}\@percentchar}%
5566   }%
5567   \ifstrempy{#1}%
5568     {\dolistloop{\@series}}%
5569     {\docsvlist{#1}}%
5570   }{\led@err@toendnotes@outsidenumbering}%
5571 }%
5572 \WithSuffix\newcommandx\toendnotes*[2][1,usedefault]{%
5573   \ifboolexpr{bool{numbering} or bool{numberingR}}{%
5574     \def\do##1{%
5575       \expandafter\immediate\expandafter\write\csname l@d@##1end\endcsname%
5576       {#2\@percentchar}%
5577     }%
5578     \ifstrempy{#1}%
5579       {\dolistloop{\@series}}%
5580       {\docsvlist{#1}}%
5581     }{\led@err@toendnotes@outsidenumbering}%
5582 }%
5583 %

```

### 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.

```

5584 \newif\ifXendinsertsep@
5585 \newcommand*\doendnotes[1]{%
5586   \l@dend@close{#1}%
5587   \begingroup
5588     \csxdef{prevpagenum@#1}{}%
5589     \csxdef{prevpagerange@#1}{}%
5590     \global\toggletrue{Xendfirstnote@#1}%
5591     \makeatletter
5592     \expandafter\let\csname #1end\endcsname=\endprint
5593     \input\l@auxdir\jobname.#1end%
5594     \global\Xendinsertsep@false%
5595   \endgroup}
5596 %

```

`\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.

```

5597 \newcommand*\doendnotesbysection[1]{%
5598   \l@dend@close{#1}%
5599   \csxdef{prevpagenum@#1}{}%
5600   \csxdef{prevpagerange@#1}{}%

```

```

5601 \global\expandafter\advance\csname #1end@bysection\endcsname by 1%
5602 \begingroup%
5603   \makeatletter%
5604   \def\l@d@section##1{%
5605     \global\toggletrue{Xendfirstnote@#1}%
5606     \ifnumequal{##1}{\csname #1end@bysection\endcsname}%
5607       {\cslet{#1end}{\endprint}}%
5608       {\cslet{#1end}{\@gobbleseven}}%
5609   }%
5610   \input\l@auxdir\jobname.#1end%
5611   \global\Xendinsertsep@false%
5612 \endgroup%
5613 }%
5614 %

```

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.

```

5615 }%
5616 %

```

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.

```

\setprintendlines 17 \newif\ifl@d@Xendlinenumber%
\ifl@d@Xendlinenumber 18 \newcommand*{\setprintendlines}[6]{%
5619   \let\@annot@start@print\relax%
5620   \let\@annot@end@print\relax%
5621   \l@d@pnumfalse%
5622   \l@d@dashfalse%
5623   \l@d@elinfalse%
5624 %

```

Should we print the line number? It depends on hooks and annotations.

```

5625   \l@d@Xendlinenumbertrue%
5626   \ifboolexpr{togl{Xendnolinenumber@\@currentseries}}%
5627     or (%
5628       togl{Xendnolinenumberifannotation@\@currentseries}%
5629       and not test{\ifdefvoid{\annot@start}}%
5630       and not test{\ifdefvoid{\annot@end}}%
5631     )%
5632   }{%
5633     \l@d@Xendlinenumberfalse%
5634   }{}%
5635 %

```

We print the second page number only if the ending page number is different

```

5636 \ifnum#4=#1 \else
5637 \l@d@pnumtrue
5638 \l@d@dashtrue
5639 \fi
5640 %

```

We print the ending line number if:

1. we print the ending page number;
2. or the ending line is different from the starting line number and we didn't disable the line number printing.

If either of these conditions is true, we also print the annotation linked to the ending line number annotations.

```

5641 \ifboolexpr{%
5642   bool{\l@d@pnum}%
5643   or not test{\ifnumequal{#2}{#5}}%
5644   and bool{\l@d@Xendlinenumber}%
5645 }{%
5646   \l@d@elintrue%
5647   \l@d@dashtrue%
5648   \l@d@elintrue
5649   \l@d@dashtrue
5650   \unless\ifx\relax\annot@end%
5651   \def\@annot@end@print{%
5652     \l@wrapcs@ifnotemptybox{Xendwraplinenumannotation@\@currentseries
5653 }{\@annot@end}%
5654   }%
5655   \fi%
5656 }%
5657 {}%
5658 %

```

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.

```

5658 \ifl@d@elin%
5659 \def\@annot@start@print{%
5660   \l@wrapcs@ifnotemptybox{Xendwraplinenumannotation@\@currentseries}{\@
5661   annot@start}%
5662   }%
5663   \else%
5664     \ifx\@annot@start\@annot@end%
5665     \unless\ifx\@annot@start\relax%
5666     \def\@annot@start@print{%
5667       \l@wrapcs@ifnotemptybox{Xendwraplinenumannotation@\@currentseries
5668 }{\@annot@start}%
5669     }%
5670     \fi%
5671   \else%
5672     \ifx\@annot@end@print\relax%
5673     \def\@annot@start@print{%
5674       \l@wrapcs@ifnotemptybox{Xendwraplinenumannotation@\@currentseries
5675 }{\@

```

```

5673         \ifx\annot@start\empty\else%
5674         \annot@start%
5675         \ifdefined\linrangesep%
5676         \linrangesep%
5677         \else%
5678         \csuse{Xendlinrangeseparator@\@currentseries}%
5679         \fi%
5680     \fi%
5681     \annot@end%
5682 }%
5683 }%
5684 \else%
5685     \let\@annot@start@print\@annot@end@print%
5686     \let\@annot@end@print\relax%
5687 \fi%
5688 \fi%
5689 \fi%
5690 %

```

We print the starting sub-line if it is nonzero.

```

5691 \l@d@ssubfalse
5692 \ifnum#3=0 \else
5693     \l@d@ssubtrue
5694 \fi
5695 %

```

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.

```

5696 \l@d@eslfalse
5697 \ifnum#6=0 \else
5698     \ifnum#6=#3
5699         \ifl@d@elin \l@d@esltrue \else \l@d@eslfalse \fi
5700     \else
5701         \l@d@esltrue
5702         \l@d@dashtrue
5703     \fi
5704 \fi%
5705 %

```

```

5706 \ifl@d@dash%
5707     \ifboolexpr{togl{fulllines@} or test{\ifcsempy{Xendtwolines@
@currentseries}}}%
5708     {%
5709     {%
5710     \setistwofollowinglines{#1}{#2}{#4}{#5}%
5711     \ifboolexpr{%
5712         (%
5713         togl {Xendtwolinesbutnotmore@\@currentseries}%
5714         and not%
5715         (%
5716         bool {istwofollowinglines@}%
5717         )%
5718     )%
5719     or%

```

```

5720      (%)
5721      (not test{\ifnumequal{#1}{#4}})%
5722      and togl{Xendtwolinesonlyinsamepage@ \@currentseries}%
5723      )%
5724      }%
5725      {}%
5726      {%
5727      \l@d@dashfalse%
5728      \l@d@Xtwolinestrue%
5729      \l@d@elinfalse%
5730      \l@d@eslfalse%
5731      \ifcsempy{Xendmorethantwolines@ \@currentseries}%
5732      }%
5733      {\ifistwofollowinglines@ \else%
5734      \l@d@Xmorethantwolinestrue%
5735      \fi%
5736      }%
5737      }%
5738      }%
5739      \fi%
5740      %

```

If the `\Xendnoidenticallinenumannotation` is set for this series, we check if the ending annotation is identical to the starting. If true, we don't print the ending annotation

```

5741      \iftoggle{Xendnoidenticallinenumannotation@ \@currentseries}{%
5742      \ifx\annot@start\annot@end%
5743      \let\@annot@end@print\relax%
5744      \ifboolexpr{%Do not print the dash if we do not print the line number
5745      test{\ifdefempty{\linenumrep}}%
5746      or not bool {l@d@Xendlinenum}%
5747      }{%
5748      \l@d@dashfalse%
5749      }{}%
5750      \fi%
5751      }{}%
5752      %

```

Finally, we check for `\Xendlinenumannotationonlyfirst` and `\Xendlinenumannotationonlyfirst` and we redefine, if required, `\@annot@start@print` and `\@annot@end@print`. We also store the current line number annotations.

```

5753      \iftoggle{Xendlinenumannotationonlyfirst@ \@currentseries}{%
5754      \ifboolexpr{%
5755      (%)
5756      togl{Xendlinenumannotationonlyfirstintwo@ \@currentseries}%
5757      and test{\ifcsequal{annot@start}{prevannot@start@ \@currentseries}}%
5758      and test{\ifcsequal{annot@end}{prevannot@end@ \@currentseries}}%
5759      )%
5760      or%
5761      (%)
5762      not togl{Xendlinenumannotationonlyfirstintwo@ \@currentseries}%
5763      test{\ifcsequal{annot@start}{prevannot@start@ \@currentseries}}%
5764      )%
5765      }{%

```

```

5766     \def\@annot@start@print{%
5767         \l@wrapcs@ifnotemptybox{Xendwraplinenumannotation@\@currentseries
} {\csuse{Xendsymmlinenumannotation@\@currentseries}}%
5768     }%
5769     \let\@annot@end@print\relax%
5770     \ifboolexpr{%Do not print the dash if do not print the line number
5771         test{\ifdefempty{\linenumrep}}%
5772         or not bool {l@d@Xendlinenum}}%
5773     }{%
5774         \l@d@d@dashfalse%
5775     }{%}
5776 }{%}
5777 \global\cslet{prevannot@start@\@currentseries}{\@annot@start}%
5778 \global\cslet{prevannot@end@\@currentseries}{\@annot@end}%
5779 }%
5780 }%
5781 %

End of \setprintendlines.

5782 }%
5783 %

```

`\printendlines` Now we are ready to print it all.

```

5784 \def\printendlines#1|#2|#3|#4|#5|#6|#7|#8|{%
5785     \begingroup
5786     \setprintendlines{#1}{#2}{#3}{#4}{#5}{#6}%
5787 %

```

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.

```

5788     \ifdimequal{\csuse{Xendboxstartlinenum@\@currentseries}}{0pt}%
5789     {\bgroup}%
5790     {\leavevmode\hbox to \csuse{Xendboxstartlinenum@\@currentseries}\bgroup
\hfill}%
5791 %

```

Then, print the starting page number-

```

5792     \ifboolexpr{%
5793         (%
5794             test{\ifcsstring{prevpagenum@\@currentseries}{#1}}%
5795             and not%
5796             (togl{Xendpagenumberonlyfirstifsingl@\@currentseries} and bool{
l@d@pnum})%
5797         )%
5798         or%
5799         (%
5800             test {\ifcsstring{prevpagerange@\@currentseries}{#1-#4}}%
5801         )%
5802     }%

```

```

5803 {%
5804 \ifcempty{Xendsympagenum@\@currentseries}%
5805   {\hspace{\csuse{Xendinplaceofpagenumber@\@currentseries}}}%
5806   {\csuse{Xendsympagenum@\@currentseries}}%
5807 }%
5808 {%
5809 \wrap@edcrossref{\@this@crossref@start}{\printnpnum{#1}}%
5810 }%
5811 %

```

Then, determine what must be printed before the start line.

```

5812 \ifl@d@dash%
5813 \ifl@d@pnum%
5814   \csuse{Xendlineprefixsingle@\@currentseries}%
5815 \else%
5816   \ifcempty{Xendlineprefixmore@\@currentseries}%
5817     {\csuse{Xendlineprefixsingle@\@currentseries}}%
5818     {\csuse{Xendlineprefixmore@\@currentseries}}%
5819 \fi%
5820 \else%
5821   \csuse{Xendlineprefixsingle@\@currentseries}%
5822 \fi%
5823 %

```

Then print the starting line, followed, if needed, by the side flag and the starting sub line number, then the line number annotation.

```

5824 \ifcsstring{Xendlinenumannotationposition@\@currentseries}{before}%
5825   {\@annot@start@print}%
5826   }%
5827 \ifl@d@Xendlinenum%
5828   \wrap@edcrossref{\@this@crossref@start}{%
5829     \ifledRcol@%
5830     \linenumrepR{#2}%
5831     \else%
5832     \linenumrep{#2}%
5833     \fi%
5834   }%
5835 \fi%
5836 \iftoggle{Xendlineflag@\@currentseries}{\ifledRcol@\@Rlineflag\fi}{}%
5837 \iftoggle{Xendnotboxingsubline@\@currentseries}{\egroup}{%Close the box
5838 \ifl@d@Xendlinenum%
5839   \ifl@d@ssub%
5840     \csuse{Xendsublinesep@\@currentseries}%
5841     \wrap@edcrossref{\@this@crossref@start}{%
5842       \ifledRcol@%
5843       \sublinenumrepR{#3}%
5844       \else%
5845       \sublinenumrep{#3}%
5846       \fi%
5847     }%
5848     \fi%
5849 \fi%
5850 \ifcsstring{Xendlinenumannotationposition@\@currentseries}{after}%
5851   {\@annot@start@print}%

```



```
5852   {}%
5853 %
```

Close the box.

```
5854   \iftoggle{Xendnotboxingsubline@\@currentseries}{\egroup}%
5855 %
```

Open the box for the ending line number.

```
5856   \ifdimequal{\csuse{Xendboxendlinenum@\@currentseries}}{0pt}%
5857   {\bgroup}%
5858   {\hbox to \csuse{Xendboxendlinenum@\@currentseries}\bgroup}%
5859 %
```

Print the dash + the ending line number, or the line number range symbol.

```
5860   \ifl@d@Xtwolines%
5861   \ifl@d@Xendlinenum%
5862   \ifl@d@Xmorethantwolines%
5863   \wrap@edcrossref{\@this@crossref@end}{\csuse{Xendmorethantwolines@\@currentseries}}%
5864   \else%
5865   \wrap@edcrossref{\@this@crossref@end}{\csuse{Xendtwolines@\@currentseries}}%
5866   \fi%
5867   \fi%
5868   \else%
5869   \ifl@d@dash%
5870   \ifdefined\linerangesep%
5871   \linerangesep%
5872   \else%
5873   \csuse{Xendlinerangeseparator@\@currentseries}%
5874   \fi%
5875   \fi%
5876 %
```

Print the ending page number.

```
5877   \ifl@d@pnum%
5878   \ifcsstring{prevpagerange@\@currentseries}{#1-#4}%
5879   {%
5880   \ifcempty{Xendsympagenum@\@currentseries}%
5881   {\hspace{\csuse{Xendinplaceofpagenumber@\@currentseries}}}%
5882   {\csuse{Xendsympagenum@\@currentseries}}%
5883   }%
5884   {%
5885   \wrap@edcrossref{\@this@crossref@end}{\printpnum{#4}}%
5886   }%
5887   \fi%
5888 %
```

Print the ending line number, with, if needed, the line prefix, and followed by the side flag, the subtitle number, and line number annotation.

```
5889   \ifcsstring{Xendlinenumannotationposition@\@currentseries}{before}%
5890   {\@annot@end@print}%
5891   {}%
```

```

5892 \ifl@d@Xendlinenumber%
5893 \ifl@d@elin%
5894 \ifl@d@pnum\csuse{Xendlineprefixsingle@\@currentseries}\fi%
5895 \wrap@edcrossref{\@this@crossref@end}{%
5896 \ifledRcol%
5897 \linenumrepR{#5}%
5898 \else%
5899 \linenumrep{#5}%
5900 \fi%
5901 }%
5902 \iftoggle{Xendlineflag@\@currentseries}{\ifledRcol@\@Rlineflag\fi
5903 }{}%
5904 \fi%
5905 \ifl@d@esl%
5906 \ifl@d@elin%
5907 \csuse{Xendsublinesep@\@currentseries}%
5908 \fi%
5909 \wrap@edcrossref{\@this@crossref@end}{%
5910 \ifledRcol%
5911 \sublinenumrepR{#6}%
5912 \else%
5913 \sublinenumrep{#6}%
5914 \fi%
5915 }%
5916 \fi%
5917 \ifcsstring{Xendlinenumannotationposition@\@currentseries}{after}%
5918 {\@annot@end@print}%
5919 {}%
5920 \fi%
5921 %

```

Close the ending line box.

```

5922 \ifdimequal{\csuse{Xendboxendlinenum@\@currentseries}}{Opt}%
5923 {}%
5924 {\hfill}%Prevent underfull hbox
5925 \egroup%
5926 %

```

And, finally, save, if needed, the current page number for the Xendpagenumberonlyfirst hooks.

```

5927 \iftoggle{Xendpagenumberonlyfirst@\@currentseries}%
5928 {\iftoggle{Xendpagenumberonlyfirstintwo@\@currentseries}%
5929 {\csxdef{prevpagerange@\@currentseries}{#1-#4}}%
5930 {\csxdef{prevpagenum@\@currentseries}{#4}}%
5931 }%
5932 {}%
5933 %

```

Now, the end of \printendlines macro.

```

5934 \endgroup%
5935 }%
5936 %
5937 %

```

`\printnnum` A macro to print a page number in an endnote. Should not be override anymore

```
5938 \newcommand*\printnnum}[1]{\csuse{Xendbeforepagenumber@\@currentseries
5939 }#1\csuse{Xendafterpagenumber@\@currentseries}}
5940 %
```

## 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@`.

```
5941 \newcommand*\newseries}[1]{%
5942   \def\do##1{\newseries@{##1}}%
5943   \docsvlist{#1}
5944 }
5945 %
```

`\@series` The `\series@` macro is an etoolbox list, which contains the name of all series.

```
5946 \newcommand{\@series}{}
5947 %
```

The command `\newseries@\series` creates a new series of the footnote.

```
\newseries@48 \newcommand*\newseries@[1]{
5949 %
```

### XXI.1 Test if series is still existing

```
5950 \xifinlist{#1}{\@series}{\led@warn@SeriesStillExist{#1}}%
5951 {%
5952 %
```

### XXI.2 Init specific to reledpar

When calling `\newseries@` after having loaded `reledpar`, we need to load specific setting.

```
5953 \ifdefined\newseries@par%
5954 \newseries@par{#1}%
5955 \fi%
5956 %
```

### XXI.3 For critical footnotes

Critical footnotes are those which start with letters. We look for the `\nocritical` option of `reledmac`.

```
5957 \unless\ifnocritical@
5958 %
```

**XXI.3.1 Options**

```

5959 \newtoggle{Xlineflag@#1}
5960 \newtoggle{Xparindent@#1}
5961 \newtoggle{Xlemmadisablefontselection@#1}
5962 \csgdef{Xwrapcontent@#1}{}%
5963 \csgdef{Xbeforeinserting@#1}{}%
5964 \csgdef{Xhangindent@#1}{Opt}%
5965 \csgdef{Xragged@#1}{}%
5966 \csgdef{Xhsizetwocol@#1}{0.45 \hsize}%
5967 \csgdef{Xsizethreecol@#1}{.3 \hsize}%
5968 \csgdef{Xcolalign@#1}{\raggedright}%
5969 \csgdef{Xnotenumberfont@#1}{\normalfont}%
5970 \csgdef{Xnotefontsize@#1}{\footnotesize}%
5971 \csgdef{Xbhooknote@#1}{}%
5972 \csgdef{Xbhookgroup@#1}{}%
5973
5974 \csgdef{Xboxlinenum@#1}{Opt}%
5975 \csgdef{Xboxlinenumalign@#1}{L}%
5976
5977 \csgdef{Xboxstartlinenum@#1}{Opt}%
5978 \newtoggle{Xnotboxingsubline@#1}%
5979 \csgdef{Xboxendlinenum@#1}{Opt}%
5980
5981 \csgdef{Xboxsymlinenum@#1}{Opt}%
5982 \newtoggle{Xgroupbyline@#1}%
5983 \newtoggle{Xgroupbylineseparetwelines@#1}%
5984
5985 \newtoggle{Xnumberonlyfirstinline@#1}%
5986 \newtoggle{Xnumberonlyfirstintwelines@#1}%
5987
5988 \newtoggle{Xlinenumannotationonlyfirst@#1}%
5989 \newtoggle{Xlinenumannotationonlyfirstintwo@#1}%
5990
5991 \csgdef{Xtwolines@#1}{}%
5992 \csgdef{Xmorethantwolines@#1}{}%
5993 \csgdef{Xsublinesep@#1}{\fullstop}%
5994 \csgdef{Xpagelinesep@#1}{\csname Xsublinesep@#1\endcsname}%for
backward compatibility, call Xsublinesep@#1
5995 \newtoggle{Xtwolinesbutnotmore@#1}%
5996 \newtoggle{Xtwolinesonlyinsamepage@#1}%
5997 \newtoggle{Xonlypstart@#1}%
5998 \newtoggle{Xpstarteverytime@#1}%
5999
6000 \newtoggle{Xpstart@#1}%
6001 \newtoggle{Xpstartonlyfirst@#1}%
6002 \csdef{Xprevpstart@#1}{0}%Not a hook, but a tool for the hook
Xpstartonlyfirst
6003 \csgdef{Xpstartseparator@#1}{}%
6004
6005 \newtoggle{Xstanza@#1}%
6006 \newtoggle{Xstanzaonlyfirst@#1}%
6007 \csdef{Xprevstanza@#1}{0}%Not a hook, but a tool for the hook
Xstanzaonlyfirst

```

```

6008 \csgdef{Xstanzaseparator@#1}{}%
6009
6010 \csgdef{Xsymlinenum@#1}{}%
6011 \csgdef{Xsymlinenumannotation@#1}{}%
6012
6013 \newtoggle{Xnonote@#1}%
6014
6015 \newtoggle{Xnonumber@#1}%
6016 \newtoggle{Xnolinenum@#1}%
6017 \newtoggle{Xnolinenumannotation@#1}%
6018 \csgdef{Xbeforenumber@#1}{0pt}%
6019 \csgdef{Xtxtbeforenumber@#1}{}%
6020 \newtoggle{Xnopagenumberifcurrent@#1}%
6021 \csgdef{Xbeforepagenumber@#1}{}%
6022 \csgdef{Xafternumber@#1}{0.5em}%
6023 \newtoggle{Xnonbreakableafternumber@#1}%
6024 \csgdef{Xbeforesymlinenum@#1}{\csuse{Xbeforenumber@#1}}%
6025 \csgdef{Xaftersymlinenum@#1}{\csuse{Xafternumber@#1}}%
6026 \csgdef{Xinplaceofnumber@#1}{1em}%
6027 \global\csllet{Xlemmaseparator@#1}{\rbracket}%
6028 \csgdef{Xbeforelemmaseparator@#1}{0em}%
6029 \csgdef{Xafterlemmaseparator@#1}{0.5em}%
6030 \csgdef{Xinplaceoflemmaseparator@#1}{1em}%
6031 \csgdef{Xbeforenotes@#1}{1.2em \@plus .6em \@minus .6em}
6032 \csgdef{Xafterrule@#1}{0pt}
6033
6034
6035 \csgdef{Xtxtbeforenotes@#1}{%
6036 \newtoggle{Xtxtbeforenotes@#1@typeset}}%Not directly used by user,
but internal
6037 \newtoggle{Xtxtbeforenotesonlyonce@#1}%
6038
6039 \csgdef{Xmaxhnotes@#1}{0.8\vsizer}
6040 \newtoggle{Xnoteswidthliketwocolumns@#1}%
6041 \csgdef{Xparafootsep@#1}{}%
6042 \csgdef{Xafternote@#1}{1em plus.4em minus.4em}
6043 \csgdef{Xlinerrangeseparator@#1}{\endashchar}%
6044
6045 \csgdef{Xlemmafnt@#1}{}%
6046 \csgdef{Xwraplemma@#1}{%
6047 \csgdef{Xwidth@#1}{\hsizer}%
6048 \csgdef{Xwraplinenumannotation@#1}{\textsuperscript}%
6049 \csgdef{Xlinenumannotationposition@#1}{after}%
6050 \expandafter\csgdef{Xinnotemark@#1}##1{\ledinnotemark{##1}}%
6051 \newtoggle{Xnoidenticallinenumannotation@#1}%
6052 %

```

### XXI.3.2 Create inserts, needed to add notes in foot

As regards inserts, see chapter 15 of *The TeXbook* by D. Knuth.

```

6053 \expandafter\newinsert\csname #1footins\endcsname%
6054 \unless\ifnoledgroup%
6055 \expandafter\newinsert\csname mp#1footins\endcsname%
6056 \fi%

```

6057 %

**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.

```

6058 \global\newcommand{\parapparatus@}{\expandafter\newcommand\expandafter
*}{\expandafter\newcommand}\csname #1footnote\endcsname [2] [] {%
6059 \xdef\@currentseries{#1}%
6060 \nottoggle{Xnonote@#1}{%
6061 \advance \edtext@notes by 1%
6062 \ifedtext@secondarg{%
6063 \ifledRcol%
6064 \ifcsstring{Xonlyside@#1}{L}{\
led@error@note@called@onrightside{#1footnote}}}{%
6065 \else%
6066 \ifcsstring{Xonlyside@#1}{R}{\
led@error@note@called@onleftside{#1footnote}}}{%
6067 \fi%
6068 \begingroup%
6069 \newcommand{\content}{##2}%
6070 \ifcsstring{series@display#1}{paragraph}{%
6071 \patchcmd{\@tag}{\par}{\relax}{}{}%
6072 }{}%
6073 \ifnumberedpar{%
6074 \ifledRcol%
6075 \ifluatex%
6076 \footnotelang@lua[R]%
6077 \fi%
6078 \@ifundefined{xpg@main@language}{%if polyglossia
6079 {}%
6080 {\footnotelang@poly[R]}%
6081 \footnoteoptions@{R}{##1}{true}%
6082 \xright@appenditem{%
6083 \ifbool{indtl@innote}%
6084 {\unexpanded{\let\index\nindex}}%
6085 {}%
6086 \ifbool{indtl@notenumber}%
6087 {\unexpanded{\let\index\nindex}}%There is no note
number... so
6088 {}%
6089 \noexpand\Xnote@true%
6090 \unexpanded{\gdef\@currentnotetype{critical}}%
6091 \noexpand\prepare@Xprenotes{#1}%
6092 \noexpand\prepare@edindex@fornote{\l@d@nums}%
6093 \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.
6094 \noexpand\setcounter{stanzaR}{\the\c@stanzaR}%Save
stanzaR counter for footnote
6095 \unexpanded{\def\@this@crossref@start}{\theedtext:
start}%

```

```

6096         \unexpanded{\def\@this@crossref@end}{\theedtext:end}%
6097         \expandonce{\@beforeinsertofthisedtext}% Internal for
now, no reason to make it public
6098         \noexpand\parse@annot\l@current@annotR|%
6099         \noexpand\csuse{v#1footnote}{#1}%
6100         {{\l@d@nums}{\expandonce\@tag}{\expandonce\content}}
%
6101         \noexpand\Xnote@false%
6102         \unexpanded{\advance\@edindex@fornote\@m@ne}%
6103         \ifbool{indtl@innote}%
6104         {\unexpanded{\let\index\orig@index}}%
6105         {}%
6106         \ifbool{indtl@notenumber}%
6107         {\unexpanded{\let\index\orig@index}}%
6108         {}%
6109     }\to\inserts@listR
6110     \footnoteoptions@{R}{##1}{false}%
6111     \global\advance\insert@countR \@ne%
6112 \else%
6113     \ifluatex%
6114     \footnotelang@lua%
6115     \fi%
6116     \ifundefined{xpg@main@language}%if polyglossia
6117     {}%
6118     {\footnotelang@poly}%
6119     \footnoteoptions@{L}{##1}{true}%
6120     \xright@appenditem{%
6121     \ifbool{indtl@innote}%
6122     {\unexpanded{\let\index\nindex}}%
6123     {}%
6124     \ifbool{indtl@notenumber}%
6125     {\unexpanded{\let\index\nindex}}%There is no note
number... so
6126     {}%
6127     \noexpand\Xnote@true%
6128     \unexpanded{\gdef\@currentnotetype{critical}}%
6129     \noexpand\prepare@Xprenotes{#1}%
6130     \noexpand\prepare@edindex@fornote{\l@d@nums}%
6131     \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.
6132     \ifl@dpairing%
6133     \noexpand\setcounter{stanzaL}{\the\c@stanzaL}%Save
stanzaR counter for footnote
6134     \fi%
6135     \unexpanded{\def\@this@crossref@start}{\theedtext:
start}%
6136     \unexpanded{\def\@this@crossref@end}{\theedtext:end}%
6137     \expandonce{\@beforeinsertofthisedtext}%Internal for
now, no reason to make it public
6138     \noexpand\parse@annot\l@current@annotR|%
6139     \noexpand\csuse{v#1footnote}%
6140     {#1}%

```

```

6141         {{{\l@d@nums}}{\expandonce\@tag}{\expandonce\content
}}}%
6142         \noexpand\Xnote@false%
6143         \unexpanded{\advance\@edindex@fornote@m@ne}%
6144         \ifbool{indtl@innote}%
6145             {\unexpanded{\let\index\orig@@index}}%
6146             {}%
6147         \ifbool{indtl@notenumber}%
6148             {\unexpanded{\let\index\orig@@index}}%
6149             {}%
6150         }\to\inserts@list
6151         \global\advance\insert@count \@ne%
6152         \footnoteoptions@{L}{##1}{false}%
6153         \fi
6154     \else
6155         \csuse{v#1footnote}{#1}{0|0|0|0|0|0}{##1}%
6156         \fi%
6157     \endgroup%
6158 \else%
6159     \led@err@FootnoteNotInSecondArgEdtext{#1}%
6160 \fi%
6161 }\@@nonneed@notestruer}% End of \nottoggle{Xnnote@#1}{%
6162 \ignorespaces%
6163 }
6164 %

```

Create counter used to determine on which page the previous note was called.

```

6165     \expandafter\newcount\csname #1prevpage@num\endcsname%
6166     \expandafter\newcount\csname #1prevpage@numR\endcsname%
6167 %

```

We need to be able to modify reledmac's footnote macros and restore their

```

6168     \global\csletcs{#1@footnote}{#1footnote}
6169 %

```

### XXI.3.4 Set standard display

```

6170     \Xarrangement@normal{#1}%
6171 %

```

End of for critical footnotes.

```

6172     \fi
6173 %

```

### XXI.4 For familiar footnotes

Familiar footnotes are those which end with letters. We look for the `nofamiliar` option of `reledmac`.

```

6174     \unless\ifnofamiliar@
6175 %

```



## XXI.4.1 Options

```

6176 \newtoggle{parindentX@#1}
6177 \csgdef{wrapcontentX@#1}{}%
6178 \csgdef{hangindentX@#1}{Opt}%
6179 \csgdef{beforeinsertingX@#1}{}%
6180 \csgdef{raggedX@#1}{}%
6181 \csgdef{hsizetwocolX@#1}{0.45 \hsize}%
6182 \csgdef{hsizethreecolX@#1}{.3 \hsize}%
6183 \csgdef{colalignX@#1}{\raggedright}%
6184 \csgdef{notenumfontX@#1}{\normalfont}%
6185 \csgdef{notefontsizeX@#1}{\footnotesize}%
6186 \csgdef{bhooknoteX@#1}{}%
6187 \csgdef{bhookgroupX@#1}{}%
6188 \csgdef{afterruleX@#1}{Opt}
6189 \csgdef{beforenotesX@#1}{1.2em \@plus .6em \@minus .6em}
6190 \csgdef{maxhnotesX@#1}{0.8\vsizex}%
6191 \newtoggle{noteswidthliketwocolumnsX@#1}%
6192 \csgdef{parafootsepX@#1}{}%
6193 \csgdef{afternoteX@#1}{1em plus.4em minus.4em}
6194 \csgdef{widthX@#1}{\hsizex}%
6195 \expandafter\csgdef{innotemarkX@#1}##1{\ledinnotemark{##1}}%
6196 \csgdef{txtbeforenotesX@#1}{}%
6197 \newtoggle{nonoteX@#1}%
6198 \newtoggle{txtbeforenotesX@#1@typeset}%Not directly used by user,
but internal
6199 \newtoggle{txtbeforenotesonlyonceX@#1}%
6200 % End of for familiar footnotes.
6201 % \subsubsection{Create inserts, needed to add notes in foot}
6202 % As regards inserts, see chapter 15 of the TeXBook by D. Knuth.
6203 % \begin{macrocode}
6204 \expandafter\newinsert\csname footins#1\endcsname%
6205 \unless\ifnoledgroup%
6206 \expandafter\newinsert\csname mpfootins#1\endcsname%
6207 \fi%
6208 %

```

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.

```

6209 \global\expandafter\newcommand\csname footnote#1\endcsname[2] []{%
6210 \xdef\@currentseries{#1}%
6211 \nottoggle{nonoteX@#1}{%
6212 \beginingroup%
6213 \prepare@prenotesX{#1}%
6214 \newcommand{\content}{##2}%
6215 %
6216 %

```

If we are using the `\csquotes` package, we reset the quotation level.

```

6217 \ifdefined\csq@qlevel%
6218 \csq@qlevel=0\relax%
6219 \fi%

```

```

6220 %

```

If we prepare parallel typesetting, we cannot just increase the footnote counter. Read reledpar's handbook about that (V.1.2 p. 51). If we have a manual footnote mark, use it.

```

6221     \global\expandafter\advance\csname footnote#1@reading\
endcsname by \@ne%
6222     \ifstrempy{##1}{%
6223         \ifboolexpr{bool{l@dpairing} or bool{l@dprintingpages} or
bool{l@dprintingcolumns}}{%
6224             \ifcsdef{footnote#1reading\the\csname footnote#1@reading
\endcsname=typeset}%
6225                 {\setcounter{footnote#1}{\csuse{footnote#1reading\the
csname footnote#1@reading\endcsname=typeset}}}%
6226                 {\setcounter{footnote#1}{\the\csname footnote#1@reading\
endcsname}}}%
6227         }{%
6228             \stepcounter{footnote#1}%
6229         }%
6230     }{}%
6231 %

```

We also have to check consistency with \onlysideX setting.

```

6232     \ifledRcol%
6233         \ifcsstring{onlysideX@#1}{L}{\
led@error@note@called@onrightside{footnote#1}}{}%
6234     \else%
6235         \ifcsstring{onlysideX@#1}{R}{\
led@error@note@called@onleftside{footnote#1}}{}%
6236     \fi%
6237 %

```

And now, the feature not depending on whether we prepare parallel typesetting

```

6238     \ifstrempy{##1}%
6239         {\protected@csxdef{@thefnmark#1}{\csuse{thefootnote#1}}}%
6240         {\protected@csxdef{@thefnmark#1}{##1}}%
6241     \protected@csxdef{hr@thefnmark#1}{\the\value{footnote#1}@the
\csname footnote#1@series\endcsname}%
6242     \nottoggle{nomk@}%Nomk is set to true when using \
footnoteXnomk with \parpackage
6243         {\csuse{@footnotemark#1}}%
6244         {}%
6245     \ifluatex%
6246         \xdef\footnote@luatextextdir{\the\textdir}%
6247         \xdef\footnote@luatexpardir{\the\pardir}%
6248     \fi%
6249     \if@ledgroup%
6250         \led@set@index@fornote{#1}%
6251     \fi%
6252     \csuse{vfootnote#1}{#1}{\expandonce\content}{##1}\
m@mmf@prepare%
6253     \ifbool{indtl@innote}%
6254         {\let\index\orig@index}%
6255     {}%

```

```

6256         \ifbool{indtl@notenumber}%
6257             {\let\index\orig@index}%
6258             }%
6259         \endgroup%
6260     }{}%End of the notoggleX@#1
6261 }
6262 %

```

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 to get the correct footnote number when using parallel typesetting (V.1.2 p. 51).

The  $\TeX$  counter `\footnoteX@series` is increased each time we reset the  $\LaTeX$  counter `footnoteX`, either using `+setcounter{footnoteX}{0}+` or because of `cs@addtoreset`.

```

6263     \newcounter{footnote#1}
6264     \global\expandafter\renewcommand\csname thefootnote#1\endcsname{\
arabic{footnote#1}}
6265     \expandafter\newcount\csname footnote#1@reading\endcsname%
6266     \expandafter\newcount\csname footnote#1@series\endcsname%
6267     \global\expandafter\csname footnote#1@series\endcsname=1%
6268 %

```

Create counter used to determine on which page the previous note was called.

```

6269     \expandafter\newcount\csname prevpage#1@num\endcsname%
6270     \expandafter\newcount\csname prevpage#1@numR\endcsname%
6271     \expandafter\newcount\csname footnote#1@parafootsepX\endcsname%For
the case we use \parafootsepX outside of numbered section
6272 %

```

Add `\let\footnoteX@gobble` to `\no@expands`.

```

6273     \expandafter\gappto\expandafter\no@expands\expandafter{\expandafter\
let\csname footnote#1\endcsname@gobble}%
6274 %

```

And now, define `\footnoteXmark` and `\footnoteXtext`, equivalent to classical `\footnotemark` and `\footnotetext`.

```

6275     \expandafter\newcommand\csname footnote#1mark\endcsname [1] [] {%
6276         \begingroup%
6277             \prepare@prenotesX{#1}%
6278             \ifstrempy{##1}{%
6279                 \stepcounter{footnote#1}%
6280             }{%
6281                 \setcounter{footnote#1}{##1}%
6282             }%
6283             \protected@csxdef{@thefnmark#1}{\csuse{thefootnote#1}}%
6284             \protected@csxdef{hr@thefnmark#1}{\the\value{footnote#1}@the\
csname footnote#1@series\endcsname}%
6285             \csuse{@footnotemark#1}%
6286             \m@mf@prepare%
6287         \endgroup%
6288     }%
6289     \expandafter\newcommand\csname footnote#1text\endcsname [2] [] {%
6290         \begingroup%

```

```

6291     \csuse{vfootnote#1}{#1}{\expandonce{##2}}{##1}%
6292     \endgroup%
6293   }%
6294 %

```

Do not forget to initialize the series.

```

6295   \arrangementX@normal{#1}%
6296   \fi
6297 %

```

## XXI.5 The endnotes

Endnotes are commands like `\Xendnote`, where `X` is a series letter. First, we check for the `noend` options.

```

6298   \unless\ifnoend@
6299 %

```

### 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`

```

6300   \expandafter\newwrite\csname l@d@#1end\endcsname%
6301   \expandafter\newif\csname ifl@dend@#1\endcsname%
6302 %

```

### 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.

```

6303   \global\expandafter\newcommand\csname #1endnote\endcsname [2] [1,
6304   usedefault]{%
6305     \advance \edtext@notes by 1%
6306     \bgroup%
6307     \newlinechar='40%
6308     \newcommand{\content}{##2}%
6309     \stepcounter{labidx}%
6310     \expandafter\immediate\expandafter\write\csname l@d@#1end\
endcsname{%
6311     \unexpanded{\def\sw@list@inedtext}%
6312     {\expandafter\unexpanded\expandafter{\sw@inthisedtext}}%
6313     \@percentchar\space%Explicit space, to add a linebreak in the
output file
6314     \noexpand\parse@annot\l@current@annot|\@percentchar\space%
6315     \expandafter\string\csname #1end\endcsname%
6316     {\ifnumberedpar@l@d@nums\fi}%
6317     {\ifnumberedpar@\expandonce\@tag\fi}%

```

```

6318     {\expandonce\content}%
6319     {#1}%
6320     {\unexpanded{##1}}%
6321     {\ifledRcol R\else L\fi}%
6322     {\theedtext}%
6323     \@percentchar%
6324     }%
6325     \egroup%
6326     \ignorespaces%
6327 }%
6328 %

```

### XXI.5.3 Tools

The `\Xtoendnotes` command inserts any arbitrary content into the endnote file. It is an alias of the more generalist `\addtoendnotes`

```

6329
6330     \global\expandafter\newcommand\csname #1toendnotes\endcsname[1]{%
6331     \toendnotes[#1]{##1}%
6332     }%
6333
6334     \expandafter\WithSuffix\expandafter\newcommand\csname #1toendnotes\
6335     endcsname*[1]{%
6336     \toendnotes*[#1]{##1}%
6337     }%
6338 %

```

### 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.

```

6339
6340     \global\cslet{#1end}{\@gobbleseven}
6341 %

```

We need to store the number of times `\doendnotesbysection` is called for one series.

```

6342     \global\expandafter\newcount\csname #1end@bysection\endcsname%
6343 %

```

### XXI.5.5 The options

```

6344     \csgdef{Xendwraplemma@#1}{%
6345     \csgdef{Xendwrapcontent@#1}{}%
6346     \csgdef{Xendtwolines@#1}{}%
6347     \csgdef{Xendmorethantwolines@#1}{}%
6348     \newtoggle{Xendtwolinesbutnotmore@#1}{%

```

```

6349 \newtoggle{Xendtwolinesonlyinsamepage@#1}{}%
6350 \newtoggle{Xendlemmadisablefontselection@#1}%
6351 \csgdef{Xendnotenumfont@#1}{\normalfont}%
6352 \csgdef{Xendnotefontsize@#1}{\footnotesize}%
6353 \csgdef{Xendbhooknote@#1}{}%
6354
6355 \csgdef{Xendsublinesep@#1}{\fullstop}%
6356
6357 \csgdef{Xendbeforenumber@#1}{Opt}
6358 \csgdef{Xendafternumber@#1}{0.5em}
6359
6360 \csgdef{Xendboxlinenum@#1}{Opt}%
6361 \csgdef{Xendboxlinenumalign@#1}{L}%
6362
6363 \csgdef{Xendboxstartlinenum@#1}{Opt}%
6364 \newtoggle{Xendnotboxingsubline@#1}%
6365 \csgdef{Xendboxendlinenum@#1}{Opt}%
6366
6367 \csgdef{Xendlemmaseparator@#1}{}%
6368 \csgdef{Xendbeforelemmaseparator@#1}{0em}%
6369 \csgdef{Xendafterlemmaseparator@#1}{0.5em}%
6370 \csgdef{Xendinplaceoflemmaseparator@#1}{0.5em}%
6371
6372 \newtoggle{Xendparagraph@#1}%
6373 \csgdef{Xendafternote@#1}{1em plus.4em minus.4em}%
6374 \csgdef{Xendsep@#1}{}%
6375
6376 \csgdef{Xendinplaceofnumber@#1}{Opt}%
6377 \newtoggle{Xendnonumber@#1}%
6378 \newtoggle{Xendnolinenum@#1}%
6379 \newtoggle{Xendnolinenumifannotation@#1}%
6380
6381 \csgdef{Xendhangindent@#1}{Opt}%
6382
6383 \newtoggle{Xendnumberonlyfirstinline@#1}%
6384 \newtoggle{Xendnumberonlyfirstintwolines@#1}%
6385
6386 \newtoggle{Xendlinenumannotationonlyfirst@#1}%
6387 \newtoggle{Xendlinenumannotationonlyfirstintwo@#1}%
6388
6389 \csgdef{Xendbeforesymmlinenum@#1}{\csuse{Xendbeforenumber@#1}}%
6390 \csgdef{Xendaftersymmlinenum@#1}{\csuse{Xendafternumber@#1}}%
6391
6392 \csgdef{Xendsymmlinenum@#1}{}%
6393 \csgdef{Xendsymmlinenumannotation@#1}{}%
6394
6395 \csgdef{Xendboxsymmlinenum@#1}{Opt}%
6396
6397 \csgdef{Xendbhooklinenum@#1}{}%
6398 \csgdef{Xendehooklinenum@#1}{}%
6399 \csgdef{Xendbhookinplaceofnumber@#1}{}%
6400 \csgdef{Xendehookinplaceofnumber@#1}{}%
6401
6402 \csgdef{Xendlinrangeseparator@#1}{\endashchar}%

```

```

6403 \csgdef{Xendbeforepagenumber@#1}{p.}%
6404 \csgdef{Xendafterpagenumber@#1}{ }%
6405 \csgdef{Xendlineprefixsingle@#1}{}%
6406 \csgdef{Xendlineprefixmore@#1}{}%
6407
6408
6409 \newtoggle{Xendlineflag@#1}
6410
6411 \csgdef{Xendlemmafont@#1}{}%
6412
6413 \csgdef{Xendlinenumannotationposition@#1}{after}%
6414 \csgdef{Xendwraplinenumannotation@#1}{\textsuperscript}%
6415 \newtoggle{Xendnoidenticallinenumannotation@#1}%
6416
6417 \newtoggle{Xendpagenumberonlyfirst@#1}%
6418 \newtoggle{Xendpagenumberonlyfirstifsingle@#1}%
6419 \newtoggle{Xendpagenumberonlyfirstintwo@#1}%
6420 \csgdef{Xendsympagenum@#1}{}%
6421 \csgdef{Xendinplaceofpagenumber@#1}{Opt}%
6422
6423 \csgdef{Xendtxtbeforenotes@#1}{}%
6424 \newtoggle{Xendfirstnote@#1}%Not a hook, but used to apply
Xendtxtbeforenotes
6425
6426 %

```

End of endnotes declaration

```

6427 \fi%
6428 %

```

Dump series in \@series

```

6429 \listxadd{\@series}{#1}
6430 }
6431 }% End of \newseries
6432 %

```

## XXI.6 Init standards series (A,B,C,D,E)

```

6433 \expandafter\newseries\expandafter{\default@series}
6434 %

```

## 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.

```

6435 \newcommand{\seriesatbegin}[1]{%
6436 \StrDel{\@series}{#1}[\@series]%
6437 \edef\@new{}%
6438 \listead{\@new}{#1}%
6439 \listead{\@new}{\@series}%
6440 \xdef\@series{\@new}%

```

```
6441 }
6442 %
```

`\seriesatend` And `\seriesatend` moves the series to the end of the list.

```
6443 \newcommand{\seriesatend}[1]{%
6444   \StrDel{\@series}{#1}[\@series]%
6445   \edef\@new{%
6446     \listead{\@new}{\@series}%
6447     \listead{\@new}{#1}%
6448   }
6449 }
6450 %
```

## XXII.2 Test series order

`\ifseriesbefore` `\ifseriesbefore{<seriesA>}{<seriesB>}{<true>}{<false>}` expands to `<true>` if `<seriesA>` is printed before `<seriesB>`, or to `<false>` otherwise.

```
6451 \newcommand{\ifseriesbefore}[4]{%
6452   \StrPosition{\@series}{#1}[\@first]%
6453   \StrPosition{\@series}{#2}[\@second]%
6454   \ifnumgreater{\@second}{\@first}{#3}{#4}%
6455 }
6456 %
```

### XXII.2.1 Get the first series

In some specific case, we need to know the first series of the list of series.

```
\@getfirstseries57 \newcommand{\@getfirstseries}{%
6458   \ifdefempty{\@series}%
6459   {\xdef\@firstseries{}}%
6460   {\StrChar{\@series}{1}[\@firstseries]}%
6461 }%
6462 %
```

## 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 throughout the code, depending on 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 on the footnote display).
- #5 (optional): if not empty, and if #1 is empty, change the hook setting for pseudo-series, as `appref`.

```

6463 \newcommandx{\settoggle@series}[5][4,5,usedefault]{%
6464   \def\do##1{%
6465     \ifcsdef{etb@tgl@#2@##1}{%
6466       \global\settoggle{#2@##1}{#3}%
6467     }{%
6468       \led@warn@BadHookSeries{##1}{#2}%
6469     }%
6470     \ifstrequal{#4}{critical}{
6471       \csuse{Xarrangement@\csuse{series@display##1}}{##1}%
6472     }{
6473       \ifstrequal{#4}{familiar}{%
6474         \csuse{arrangementX@\csuse{series@displayX##1}}{##1}%
6475       }{
6476         }%
6477       \ifstrempy{#1}{%
6478         \dolistloop{\@series}%
6479         \ifstrempy{#5}{%
6480           \docsvlist{#5}%
6481         }
6482       }%
6483     }%
6484     \docsvlist{#1}%
6485   }%
6486 }
6487 %

```

`\setcommand@series` `\setcommand@series{series}{command}{value}` is a generic command to store a hook's value into commands specific to some series. The arguments are:

- #1 (mandatory): the series for which the hooks should be set. If empty, all the series will be affected.
- #2 (mandatory): the name of the hook.
- #3 (mandatory): the new value of the hook/command.
- #4 (optional): if equal to `reload`, reload the footnote setting (call `\footnormal` or `\footparagraph` or ... depending on the footnote display).
- #5 (optional): if not empty, and if #1 is empty, change the hook setting for pseudo-series, as `appref`.
- #6 (optional): the number of arguments of the hook.

```

6488 \newcommandx{\setcommand@series}[6][4,5,6=0,usedefault]{%
6489   \def\do##1{
6490     \ifcsdef{#2@##1}{%
6491       \expandafter\global\expandafter\renewcommand\csname#2@##1\endcsname
6492       [#6]{#3}%
6493     }{%
6494       \led@warn@BadHookSeries{##1}{#2}%
6495     }%
6496     \ifstrequal{#4}{critical}{%
6497       \csuse{Xarrangement@\csuse{series@display##1}}{##1}%
6498     }{}
6499     \ifstrequal{#4}{familiar}{%
6500       \csuse{arrangementX@\csuse{series@displayX##1}}{##1}%
6501     }{}%
6502   }%
6503   \ifstreempty{#1}{%
6504     \dolistloop{\@series}%
6505     \ifstreempty{#5}{%
6506       \docsvlist{#5}
6507     }
6508   }%
6509   \docsvlist{#1}%
6510 }%
6511 }%
6512 %

```

### 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`. The third argument

```

6513 \newcommandx{\newhookcommand@series}[3][2,3=0,usedefault]{%
6514   \global\expandafter\newcommand\expandafter*\csname #1\endcsname[2][ ]{%
6515     \setcommand@series{##1}{#1}{##2}[ ][#2][#3]%

```

```

6516 }%
6517 \ifstrempy{#2}{}%
6518 \def\do##1{%
6519 \global\expandafter\newcommand\expandafter*\csname #1##1\endcsname
[1]{%
6520 \csuse{#1}[##1]{###1}%
6521 }%
6522 }%
6523 \docsvlist{#2}%
6524 }%
6525 }
6526 %

```

`\newhooktoggle@series` `\newhooktoggle@series`\command names is a generic command to add new commands for a new toggle hook, like `\Xnumberonlyfirstinline`. The second argument is also used to create commands named `\<hookname><pseudoserries>`, like `\Xtwolinesbutnotmoreappref`.

```

6527 \newcommandx{\newhooktoggle@series}[2][2,usedefault]{%
6528 \global\expandafter\newcommandx\expandafter*\csname #1\endcsname [2] [1,2={
true},usedefault]{%
6529 \settoggle@series{##1}{#1}{##2} [] [#2]%
6530 }%
6531 \ifstrempy{#2}{}%
6532 \def\do##1{%
6533 \global\expandafter\newcommand\expandafter*\csname #1##1\endcsname{%
6534 \csuse{#1}[##1]%
6535 }%
6536 }%
6537 \docsvlist{#2}%
6538 }%
6539 }
6540 %

```

`\newhookcommand@toggle@reload` `\newhookcommand@toggle@reload` does the same thing as `\newhooktoggle@series` but the commands created by this macro also reload the series arrangement, depending on the type of notes

```

6541 \newcommand{\newhooktoggle@series@reload}[2]{%
6542 \global\expandafter\newcommandx\expandafter*\csname #1\endcsname [2] [1,2={
true},usedefault]{%
6543 \settoggle@series{##1}{#1}{##2} [#2]%
6544 }%
6545 }%
6546 %

```

`\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.

```

6547 \newcommand{\newhookcommand@series@reload}[2]{%
6548 \global\expandafter\newcommand\expandafter*\csname #1\endcsname [2] []{%
6549 \setcommand@series{##1}{#1}{##2} [#2]%
6550 }%
6551 }
6552 %

```

### 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.

```

6553 \unless\ifnocritical@
6554   \newhookcommand@series{Xwrapcontent}%
6555   \newhookcommand@series{Xbeforeinserting}%
6556   \newhookcommand@series{Xlemmafont}%
6557   \newhookcommand@series{Xwraplemma}%
6558   \newhooktoggle@series{Xparindent}
6559   \newhookcommand@series{Xhangindent}
6560   \newhookcommand@series{Xragged}
6561   \newhookcommand@series{Xhsizetwocol}
6562   \newhookcommand@series{Xhsizethreecol}
6563   \newhookcommand@series{Xcolalign}%
6564   \newhookcommand@series{Xnotenumfont}
6565   \newhookcommand@series{Xbhooknote}
6566   \newhookcommand@series@reload{Xbhookgroup}{critical}
6567   \newhookcommand@series{Xboxsymlinenum}%
6568   \newhookcommand@series{Xsymlinenum}
6569   \newhookcommand@series{Xsymlinenumannotation}%
6570   \newhookcommand@series{Xbeforenumber}
6571   \newhookcommand@series{Xtxtbeforenumber}
6572   \newhooktoggle@series{Xnopagenumberifcurrent}%
6573   \newhookcommand@series{Xbeforepagenumber}%
6574   \newhookcommand@series{Xafternumber}
6575   \newhookcommand@series{Xbeforesymlinenum}
6576   \newhookcommand@series{Xaftersymlinenum}
6577   \newhookcommand@series{Xinplaceofnumber}
6578   \newhookcommand@series{Xlemmaseparator}
6579   \newhookcommand@series{Xbeforelemmaseparator}
6580   \newhookcommand@series{Xafterlemmaseparator}
6581   \newhookcommand@series{Xinplaceoflemmaseparator}
6582   \newhookcommand@series{Xtxtbeforenotes}
6583   \newhooktoggle@series{Xtxtbeforenotesonlyonce}%
6584   \newhookcommand@series@reload{Xafterrule}{critical}
6585   \newhooktoggle@series{Xnumberonlyfirstinline}
6586   \newhooktoggle@series{Xnumberonlyfirstintwolines}
6587   \newhooktoggle@series{Xlinenumannotationonlyfirst}%
6588   \newhooktoggle@series{Xlinenumannotationonlyfirstintwo}%
6589   \newhooktoggle@series{Xgroupbyline}%
6590   \newhooktoggle@series{Xgroupbylineseparatetwolines}%
6591   \newhooktoggle@series{Xnonumber}
6592   \newhooktoggle@series{Xnolinenumber}%
6593   \newhooktoggle@series{Xnolinenumberifannotation}%
6594
6595   \newhooktoggle@series{Xpstart}
6596   \newhooktoggle@series{Xpstartonlyfirst}%
6597   \newhooktoggle@series{Xpstarteverytime}%
6598   \newhookcommand@series{Xpstartseparator}%
6599
6600   \newhooktoggle@series{Xstanza}%
6601   \newhooktoggle@series{Xstanzaonlyfirst}%

```

```

6602 \newhookcommand@series{Xstanzaseparator}%
6603
6604 \newhooktoggle@series{Xnonote}%
6605
6606 \newhooktoggle@series{Xonlypstart}
6607 \newhooktoggle@series{Xnonbreakableafternumber}
6608 \newhooktoggle@series{Xlemmadisablefontselection}
6609 \newhookcommand@series@reload{Xmaxhnotes}{critical}
6610 \newhookcommand@series@reload{Xbeforenotes}{critical}
6611 \newhooktoggle@series@reload{Xnoteswidthliketwocolumns}{critical}%
6612 \newhookcommand@series@reload{Xnotefontsize}{critical}
6613
6614 \newhookcommand@series{Xboxlinenum}%
6615 \newhookcommand@series{Xboxlinenumalign}%
6616
6617 \newhookcommand@series{Xboxstartlinenum}%
6618 \newhooktoggle@series{Xnotboxingsubline}%
6619 \newhookcommand@series{Xboxendlinenum}%
6620
6621 \newhookcommand@series{Xafternote}%
6622 \newhookcommand@series{Xparafootsep}
6623 \newhookcommand@series@reload{Xwidth}{critical}%
6624
6625 \newhookcommand@series{Xinnotemark} [] [1]%
6626
6627 \ifundef{\Xhsize}%
6628 {%
6629   \newcommandx{\Xhsize}[2][1,usedefault]{%
6630     \led@warning@Xhsize@deprecated%
6631     \Xwidth[#1]{#2}%
6632   }%
6633 }%
6634 {}%
6635 \fi
6636 %

```

Now, hooks for critical notes which also apply to crossreferencing and line numbering at the sides of the page.

```

6637 \newhooktoggle@series{Xlineflag}[appref,SEref]
6638 \newhookcommand@series{Xtwolines}[appref,SEref]
6639 \newhookcommand@series{Xmorethantwolines}[appref,SEref]
6640 \newhookcommand@series{Xsublinesep}[appref,SEref,side]%
6641 \newhookcommand@series{Xpagelinesep}[appref,SEref,side]%
6642 \newhooktoggle@series{Xtwolinesbutnotmore}[appref,SEref]
6643 \newhooktoggle@series{Xtwolinesonlyinsamepage}[appref,SEref]
6644 \newhookcommand@series{Xlinerrangeseparator}[appref,SEref]
6645 \newhookcommand@series{Xlinenumannotationposition}[side,appref,SEref]%
6646 \newhookcommand@series{Xwraplinenumannotation}[side,ref,appref,SEref]%
6647 \newhooktoggle@series{Xnoidenticallinenumannotation}[side,ref,appref,SEref]
6648 %

```

### XXII.3.5 Options for familiar notes

Before generating the optional commands for familiar notes, we check the `\nofamiliar` option.

```

6649 \unless\ifnofamiliar@
6650   \newhookcommand@series{wrapcontentX}%
6651   \newhookcommand@series{beforeinsertingX}%
6652   \newhooktoggle@series{parindentX}
6653   \newhookcommand@series{hangindentX}
6654   \newhookcommand@series{raggedX}
6655   \newhookcommand@series{hsizetwocolX}
6656   \newhookcommand@series{hsizethreecolX}
6657   \newhookcommand@series{colalignX}%
6658   \newhookcommand@series{notenumfontX}
6659   \newhookcommand@series{bhooknoteX}
6660   \newhookcommand@series@reload{bhookgroupX}{familiar}
6661   \newhookcommand@series@reload{beforenotesX}{familiar}
6662   \newhookcommand@series@reload{maxhnotesX}{familiar}
6663   \newhooktoggle@series@reload{noteswidthliketwocolumnsX}{familiar}%
6664   \newhookcommand@series@reload{afterruleX}{familiar}
6665   \newhookcommand@series@reload{notefontsizeX}{familiar}
6666   \newhookcommand@series{afternoteX}
6667   \newhookcommand@series{parafootsepX}
6668   \newhookcommand@series{txtbeforenotesX}%
6669   \newhooktoggle@series{txtbeforenotesonlyonceX}%
6670   \newhookcommand@series@reload{widthX}{familiar}%
6671   \newhookcommand@series{innotemarkX}[] [1]%
6672   \newhooktoggle@series{nonoteX}%
6673   \ifundef{\hsizeX}%
6674     {%
6675       \newcommandx{\hsizeX}[2] [1,usedefault] {%
6676         \led@warning@hsizeX@deprecated%
6677         \widthX [#1] {#2}%
6678       }%
6679     }%
6680   {}%
6681 \fi
6682 %

```

### 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.

```

6683 \unless\ifnoend@
6684   \newhookcommand@series{Xendwraplemma}
6685   \newhookcommand@series{Xendwrapcontent}
6686   \newhookcommand@series{Xendnotenumfont}
6687   \newhookcommand@series{Xendlemmafont}%
6688   \newhookcommand@series{Xendbhooknote}
6689
6690   \newhookcommand@series{Xendboxlinenum}%
6691   \newhookcommand@series{Xendboxlinenumalign}%
6692

```

```

6693 \newhookcommand@series{Xendboxstartlinenum}%
6694 \newhooktoggle@series{Xendnotboxingsubline}%
6695 \newhookcommand@series{Xendboxendlinenum}%
6696
6697 \newhookcommand@series{Xendnotefontsize}
6698 \newhooktoggle@series{Xendlemmadisablefontselection}
6699 \newhookcommand@series{Xendlemmaseparator}
6700 \newhookcommand@series{Xendbeforelemmaseparator}
6701 \newhookcommand@series{Xendafterlemmaseparator}
6702 \newhookcommand@series{Xendinplaceoflemmaseparator}
6703
6704 \newhookcommand@series{Xendbeforenumber}%
6705 \newhookcommand@series{Xendafternumber}%
6706
6707 \newhooktoggle@series{Xendparagraph}
6708 \newhookcommand@series{Xendafternote}
6709 \newhookcommand@series{Xendsep}
6710
6711 \newhookcommand@series{Xendinplaceofnumber}%
6712 \newhooktoggle@series{Xendnonumber}%
6713 \newhooktoggle@series{Xendnolinenum}%
6714 \newhooktoggle@series{Xendnolinenumifannotation}%
6715
6716 \newhooktoggle@series{Xendnumberonlyfirstinline}%
6717 \newhooktoggle@series{Xendnumberonlyfirstintwoline}%
6718
6719 \newhooktoggle@series{Xendlinenumannotationonlyfirst}%
6720 \newhooktoggle@series{Xendlinenumannotationonlyfirstintwo}%
6721
6722 \newhookcommand@series{Xendsymmlinenum}%
6723 \newhookcommand@series{Xendbeforesymmlinenum}%
6724 \newhookcommand@series{Xendaftersymmlinenum}%
6725 \newhookcommand@series{Xendboxsymmlinenum}%
6726
6727 \newhookcommand@series{Xendsymmlinenumannotation}%
6728
6729 \newhookcommand@series{Xendbhooklinenum}%
6730 \newhookcommand@series{Xendahooklinenum}%
6731 \newhookcommand@series{Xendbhookinplaceofnumber}%
6732 \newhookcommand@series{Xendahookinplaceofnumber}%
6733
6734 \newhookcommand@series{Xendhangindent}%
6735
6736 \newhooktoggle@series{Xendpagenumberonlyfirst}%
6737 \newhooktoggle@series{Xendpagenumberonlyfirstifsingle}%
6738 \newhooktoggle@series{Xendpagenumberonlyfirstintwo}%
6739 \newhookcommand@series{Xendsympagenum}%
6740 \newhookcommand@series{Xendinplaceofpagenumber}%
6741
6742 \newhookcommand@series{Xendtxtbeforenotes}%
6743
6744 \fi
6745 \newhooktoggle@series{Xendlineflag}[apprefwithpage,SErefwithpage]
6746 \newhookcommand@series{Xendtwoline}[apprefwithpage,SErefwithpage]

```

```

6747 \newhookcommand@series{Xendmorethantwolines}[apprefwithpage,SErefwithpage]
6748 \newhooktoggle@series{Xendtwolinesbutnotmore}[apprefwithpage,SErefwithpage]
6749 \newhooktoggle@series{Xendtwolinesonlyinsamepage}[apprefwithpage,
SErefwithpage]
6750 \newhookcommand@series{Xendlinerangeseparator}[apprefwithpage,SErefwithpage
]
6751 \newhookcommand@series{Xendbeforepagenumber}[apprefwithpage,SErefwithpage,
SErefonlypage]
6752 \newhookcommand@series{Xendafterpagenumber}[apprefwithpage,SErefwithpage]
6753 \newhookcommand@series{Xendlineprefixsingle}[apprefwithpage,SErefwithpage]
6754 \newhookcommand@series{Xendlineprefixmore}[apprefwithpage,SErefwithpage]
6755 \newhookcommand@series{Xendsublinesep}[apprefwithpage,SErefwithpage]
6756
6757 \newhookcommand@series{Xendlinenumannotationposition}[apprefwithpage,
SErefwithpage]%
6758 \newhookcommand@series{Xendwraplinenumannotation}[apprefwithpage,
SErefwithpage]%
6759 \newhooktoggle@series{Xendnoidenticallinenumannotation}[apprefwithpage,
SErefwithpage]%
6760 %

```

#### XXII.4 Hooks for a particular footnote

`\newhooktoggle@specific` `\newhooktoggle@specific` is a generic command to create boolean hook specific to a note.

```

6761 \newcommand{\newhooktoggle@specific}[1]{%
6762   \newtoggle{#1}%
6763   \listgadd{\hooktoggle@specific}{#1}%
6764   \define@key[mac]{truefootnoteoption}{#1}[]{\global\settoggle{#1}{true}}%
When enabling footnote option
6765   \define@key[mac]{falsefootnoteoption}{#1}[]{\global\settoggle{#1}{false
}}
6766 }
6767 %

```

`\newhookarg@specific` `\newhookarg@specific` is a generic command to create argumen hook specific to a note.

```

6768 \newcommand{\newhookarg@specific}[1]{%
6769   \listgadd{\hookarg@specific}{#1}%
6770   \define@key[mac]{truefootnoteoption}{#1}{\csgdef{#1}{##1}}%When enabling
footnote option
6771   \define@key[mac]{falsefootnoteoption}{#1}{\global\csundef{#1}}%When
disabling footnote option
6772 }
6773 %

```

`\hooktoggle@specific` `\add@hooktoggle@specific@to@cs` The `\add@hooktoggle@specific@to@cs` macro stores to a macro all the current values of hook toggle specific to a command. It is useful for the `\Xgroupbyline` option, which does not immediately add notes to the inserts list. It uses the `\hooktoggle@specific` list.



```

6774 \def\hooktoggle@specific{}%
6775 \newcommand{\add@hooktoggle@specific@to@cs}[1]{%
6776   \def\do##1{%
6777     \iftoggle{##1@}{%
6778       \ifcsdef{##1}{%
6779         \csgappto{##1}{\toggletrue{##1@}}%
6780       }{%
6781         \csgdef{##1}{\toggletrue{##1@}}%
6782       }%
6783     }%
6784     \ifcsdef{##1}{%
6785       \csgappto{##1}{\togglefalse{##1@}}%
6786     }{%
6787       \csgdef{##1}{\togglefalse{##1@}}%
6788     }%
6789   }%
6790 }%
6791 \dolistloop{\hooktoggle@specific}%
6792 }%
6793 %

```

`\hookarg@specific` The same, but for optional argument of critical footnotes with assigned value.

`\add@hookarg@specific@to@cs`

```

6794 \def\hookarg@specific{}%
6795 \newcommand{\add@hookarg@specific@to@cs}[1]{%
6796   \def\do##1{%
6797     \ifcvoid{##1@}{}%
6798     \ifcsdef{##1}{%
6799       \csxappto{##1}{%
6800         \noexpand\csdef%
6801           {##1@}{\csname##1@\endcsname}%
6802       }%
6803     }{%
6804       \csxdef{##1}{%
6805         \noexpand\csdef%
6806           {##1@}{\csname##1@\endcsname}%
6807       }%
6808     }%
6809   }%
6810 }%
6811 \dolistloop{\hookarg@specific}%
6812 }%
6813 %

```

And now, we define some hooks specific to a note.

```

6814 \newhooktoggle@specific{fulllines}%
6815 \newhooktoggle@specific{nonum}
6816 \newhooktoggle@specific{nosep}
6817 \newhooktoggle@specific{noprefix}%
6818 \newhooktoggle@specific{prefixmore}%
6819 \newhookarg@specific{linrangesep}
6820 %

```

`\linerangesep@` `\linerangesep@` is defined by the option `linerangesep` 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.

```
6821 %
```

`\nomk@` `\nomk@` toggle is used by `reledpar` to remove the footnote mark in the text when using `\footnoteXmk`. Read `reledpar` handbook.

```
6822 \newtoggle{nomk@}%
```

```
6823 %
```

## XXII.5 Alias

`\Xnolemmaseparator` `\Xnolemmaseparator` [*series*] is just an alias for `\Xlemmaseparator` [*series*] {}.

```
6824 \newcommand*{\Xnolemmaseparator}[1][1]{\Xlemmaseparator[#1]{}}
```

```
6825 %
```

## XXIII Output routine

Now we begin the output routine and associated things.

### XXIII.1 Extra footnotes output

With luck we might only have to change `\@makecol` and `\@reinserts` of  $\TeX$ 's kernel. Since `reledmac`, we use `etoolbox`'s patching commands instead of overriding. It should provides better compatibility with other package which modify these commands

`\l@ddoxtrafeet` `\l@ddoxtrafeet` is the code extending `\@makecol` to cater for the extra `reledmac` feet. We have two categories of extra footnotes. By default, we order the footnote inserts so that the regular footnotes of  $\TeX$  are first, then familiar familiar footnotes and finally the critical footnotes.

```
6826 \newcommand*{\l@ddoxtrafeet}{%
6827   \IfStrEq{familiar-critical}{\@fnpos}
6828     {\do@feetX\do@Xfeet}%
6829     {%
6830   \IfStrEq{critical-familiar}{\@fnpos}%
6831     {\do@Xfeet\do@feetX}%
6832     {%
6833       \setbox\@outputbox \vbox{%
6834         \unvbox\@outputbox%
6835         \do@feet@custom@order}{\@fnpos}%
6836       }%
6837     }%
6838   }%
6839 }%
6840
6841 %
```

`\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.

```

6842 \newcommand{\do@feet@custom@order}[2]{%
6843   \def\do##1{%
6844     \edef\@notesseries{\@firstoftwo##1}%
6845     \edef\@notetype{\@secondoftwo##1}%
6846     \ifdefstring{\@notetype}{critical}%
6847       {\csuse{#1append@Xnotes}{\@notesseries}}%
6848       {\ifdefstring{\@notetype}{familiar}%
6849         {\csuse{#1append@notesX}{\@notesseries}}%
6850         {}}%
6851     }%
6852   }%
6853   \expandafter\docsvlist\expandafter{#2}%
6854 }%
6855 %

```

`\do@Xfeet` `\do@Xfeet` is the code extending `\@makecol` to cater to the extra critical feet.

```

6856 \newcommand*{\do@Xfeet}{%
6857   \setbox\@outputbox \vbox{%
6858     \unvbox\@outputbox
6859     \@opXfeet}}
6860 %

```

`\@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
6861 \newcommand{\append@Xnotes}[1]{%
6862   \ifvoid\csuse{#1footins}\else%
6863     \global\skip\csuse{#1footins}=\csuse{Xbeforenotes@#1}%
6864     \global\advance\skip\csuse{#1footins} by\csuse{Xafterrule@#1}%
6865     \print@Xnotes{#1}%
6866   \fi%
6867 }%
6868 %

```

The normal way to add one series, `\print@Xnotes`, is replaced by `reledpar` when using `\Pages`.

```

6869 \newcommand\print@Xnotes[1]{%
6870   \xdef\@currentseries{#1}%
6871   \csuse{#1footstart}{#1}%
6872   \csuse{#1footgroup}{#1}%%
6873 }%
6874 %

```

We print all series of notes by looping on them. We check before printing them that they are not voided.

```

6875 \newcommand*{\@opXfeet}{%
6876   \unless\ifnocritical@%
6877   \gdef\firstXseries@{}%
6878   \def\do##1{%

```

```

6879     \append@Xnotes{##1}%
6880     }%
6881     \dolistloop{\@series}%
6882     \fi%
6883 }%
6884 %

```

**\loddodoreintrafeet** \loddodoreintrafeet is the code catering for the extra footnotes within \@reinserts. We use the same category and ordering as in \loddoxtrafeet.

```

6885 \newcommand*{\loddodoreintrafeet}{%
6886   \IfStrEq{familiar-critical}{\@fnpos}
6887   {\@doreinfeetX\X@doreinfeet}%
6888   {%
6889   \IfStrEq{critical-familiar}{\@fnpos}%
6890   {\X@doreinfeet\@doreinfeetX}%
6891   {\@doreinfeetX\X@doreinfeet}%
6892   }%
6893 }
6894 %
6895 %

```

**\X@doreinfeet** \X@doreinfeet is the code for catering for the extra critical footnotes within \@reinserts.

```

6896 \newcommand*{\X@doreinfeet}{%
6897   \unless\ifnocritical@%
6898   \def\do##1{%
6899     \ifvoid\csuse{##1footins}\else%
6900     \insert\csuse{##1footins}{\unvbox\csuse{##1footins}}%
6901     \fi}%
6902   \dolistloop{\@series}
6903   \fi}%
6904 }
6905 %
6906 %

```

**\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.

```

6907 \@doreinfeetX \newcommand{\append@notesX}[1]{%
6908   \ifvoid\csuse{footins#1}\else%
6909   \global\skip\csuse{footins#1}=\csuse{beforenotesX@#1}%
6910   \global\advance\skip\csuse{footins#1} by\csuse{afterruleX@#1}%
6911   \print@notesX{#1}%
6912   \fi%
6913 }%
6914 %

```

The normal way to print one series of notes. \print@Xnotes is replaced by reledpar when using \Pages.

```

6915 \newcommand\print@notesX[1]{%
6916   \xdef\@currentseries{#1}%
6917   \csuse{footstart#1}{#1}%

```

```

6918 \cuse{footgroup#1}{#1}%
6919 }%
6920 %

```

We print all the series of notes by looping over them. We check before printing them that they are not voided.

```

6921 \newcommand*\do@feetX}{%
6922   \unless\ifnofamiliar%
6923   \gdef\firstseriesX@{}%
6924   \setbox\@outputbox \vbox{%
6925     \unvbox\@outputbox%
6926     \def\do##1{%
6927       \append@notesX{##1}%
6928     }%
6929     \dolistloop{\@series}%
6930   \fi%
6931 }%
6932
6933 \newcommand{\@doreinfeetX}{%
6934   \unless\ifnofamiliar%
6935   \def\do##1{%
6936     \ifvoid\cuse{footins##1}\else
6937     \insert%
6938     \cuse{footins##1}
6939     {\unvbox\cuse{footins##1}}%
6940   \fi%
6941   }%
6942   \dolistloop{\@series}%
6943 \fi%
6944 }%
6945
6946 %

```

## 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.)

```

6947 \@ifclassloaded{memoir}{%
6948 %

```

memoir is loaded so we use memoir's built in hooks.

```

6949   \g@addto@macro{\m@doextrafeet}{\l@ddoxtrafeet}%
6950   \g@addto@macro{\m@doreinextrafeet}{\l@ddoreinextrafeet}%
6951 }{%
6952 %

```

memoir has not been loaded, so patch `\@makecol` and `\@reinserts`. If the fancyhdr package < version 3.8 has been loaded, we patch the `\latex@makecol` command, because this package redefines the standard `\@makecol` in the preamble to call `\latex@makecol`, which has been `\let` to `\@makecol`. If this package is not loaded, we directly patch

\@makecol. If the fancyhdr package  $\geq$  version 3.8, we also directly patch \@makecol, because fancyhdr does its own patch \@AtBeginDocument.

```

6953 \ifboolexpr{%
6954   test{\@ifpackageloaded{fancyhdr}}%
6955   and test {\ifdef{\latex@makecol}}%
6956 }{%
6957   \patchcmd%
6958     {\latex@makecol}%
6959     {\xdef\@freelist{\@freelist\@midlist}}%
6960     {\xdef\@freelist{\@freelist\@midlist}\l@ddoxtrafeet}%
6961     {}%
6962     {\led@error@fail@patch@@makecol}%
6963 }{%
6964   \patchcmd%
6965     {\@makecol}%
6966     {\xdef\@freelist{\@freelist\@midlist}}%
6967     {\xdef\@freelist{\@freelist\@midlist}\l@ddoxtrafeet}%
6968     {}%
6969     {\led@error@fail@patch@@makecol}%
6970 }%
6971
6972 \patchcmd%
6973   {\@reinserts}%
6974   {\ifvbox}%
6975   {\l@ddodoreinxtrafeet\ifvbox}%
6976   {}%
6977   {\led@error@fail@patch@@reinserts}%
6978 }
6979
6980 %

```

It turns out that \@doclearpage also needs modifying.

`\if@led@nofoot` We have to check if there are any leftover feet.

```

6981 \newif\if@led@nofoot
6982
6983 %

```

```

6984 \@ifclassloaded{memoir}{%
6985 %

```

If the memoir class is loaded, we hook into its modified \@doclearpage.

```

\@mem@extranofeet 86 \g@addto@macro{\@mem@extranofeet}{%%
6987   \def\do#1{%
6988     \unless\ifnocritical@%
6989       \ifvoid\csuse{#1footins}\else\@mem@nofootfalse\fi%
6990     \fi%
6991     \unless\ifnofamiliar@%
6992       \ifvoid\csuse{footins#1}\else\@mem@nofootfalse\fi%
6993     \fi%
6994   }
6995   \dolistloop{\@series}%

```

```

6996 }%
6997 }{%
6998 %

```

As memoir is not loaded we have patch \@doclearpage.

```

\@led@testifnofoot \newcommand*{\@led@testifnofoot}{%
\@doclearpage     \@led@nofoottrue%
7001               \ifvoid\footins\else%
7002                 \@led@nofootfalse%
7003               \fi%
7004               \def\do##1{%
7005                 \unless\ifnocritical@%
7006                   \ifvoid\cuse{##1footins}\else%
7007                     \@led@nofootfalse%
7008                   \fi%
7009                 \fi%
7010                 \unless\ifnofamiliar@%
7011                   \ifvoid\cuse{footins##1}\else%
7012                     \@led@nofootfalse%
7013                   \fi%
7014                 \fi%
7015               }%
7016               \dolistloop{\@series}%
7017             }%
7018
7019             \pretocmd%
7020               {\@doclearpage}%
7021               {\@led@testifnofoot}%
7022               {}%
7023               {\led@error@fail@patch@@doclearpage}%
7024
7025             \patchcmd%
7026               {\@doclearpage}%
7027               {\ifvoid\footins}%
7028               {\if@led@nofoot}%
7029               {}%
7030               {\led@error@fail@patch@@doclearpage}%
7031
7032           }
7033
7034 %

```

## 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. 125](#)). The `sameparallelpagenumber` option allows the same page number on both left and right side. The `prevpnotnumbered` 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  $\text{\the page}$ , in order to use the value of the  $\text{\par@page}$  counter instead of the value of the page counter.

This counter will be increased in a patched version of  $\text{\LaTeX}$ 's  $\text{\@outputpage}$  macro, as is the page counter in this macro. However, this increase will take account of the options.

$\text{\par@patch@thepage}$  and  $\text{\par@patch@pagenumbering}$  patches  $\text{\the page}$  in order to use the value of  $\text{\par@page}$  counter and not the value of page. It must be called after any redefinition of  $\text{\the page}$ . That is why we insert it at the end of the  $\text{\LaTeX}$  macro  $\text{\pagenumbering}$ , which is called by some  $\text{\xxxmatter}$  commands. In cases when we are using the memoir class, we insert it at the end of  $\text{\@mempnum}$ . When using  $\text{\pagenumbering}$ , we also need to reset the  $\text{\par@page}$  counter. Consequently, we put  $\text{\par@patch@thepage}$  and counter reset in  $\text{\par@patch@pagenumbering}$ . We also call  $\text{\par@patch@thepage}$  at the beginning of the document.

```

7035
7036 \newcommand{\par@patch@thepage}{%
7037   \ifboolexpr{%
7038     bool{sameparallelpagenumber}%
7039     or bool{prevpgnotnumbered}%
7040   }%
7041   {%
7042     \patchcmd{\the page}%
7043       {page}{\par@page}%
7044       {}%
7045       {\led@error@fail@patch@thepage}%
7046   }-}%
7047 }%
7048
7049 \newcommand{\par@patch@pagenumbering}{%
7050   \setcounter{\par@page}{1}%
7051   \par@patch@thepage%
7052 }%
7053
7054 \ifboolexpr{%
7055   bool {l@dmemoir}%
7056   and test {\ifdef{\@mempnum}}%Only with memoir < 3.8
7057 }{%
7058   \apptocmd{\@mempnum}%
7059     {\par@patch@pagenumbering}%
7060     {}%
7061     {\led@error@fail@patch@mempnum}%
7062   }%
7063   {%
7064     \pretocmd{\pagenumbering}%
7065       {\par@patch@pagenumbering}%
7066       {}%
7067       {\led@error@fail@patch@pagenumbering}%
7068     }%
7069
7070 \AtBeginDocument{\par@patch@thepage}%

```



```
7071 %
```

`\@outputpage` As its name says, `\@outputpage` is a  $\LaTeX$  macro called in the output routine. It is this macro which increases the page counter. We patch it in order to increase, conditionally, the `par@page` counter, and to set correctly the page number sent to the auxiliary files (`.1`, `.2`, etc.).

```
7072 \AtBeginDocument{%
7073   \set@this@c@page%
7074   \apptocmd{\@outputpage}{%
7075     \ifsameparallelpagenuumber%
7076     \ifl@dprintingpages%
7077       \ifodd@c@page\else%
7078         \stepcounter{par@page}%
7079       \fi%
7080     \else%
7081       \stepcounter{par@page}%
7082     \fi%
7083   \else%
7084     \stepcounter{par@page}%
7085   \fi%
7086 }%
7087 {}%
7088 {\led@error@fail@patch@outputpage}%
7089 }%
7090 \catcode`\#=12%
7091 \AtBeginDocument{%
7092   \apptocmd{\setcounter}{%
7093     \IfStrEq{#1}{page}{\set@this@c@page}{}%
7094   }%
7095   {}%
7096   {\led@error@fail@patch@setcounter}%
7097   \apptocmd{\addtocounter}{%
7098     \IfStrEq{#1}{page}{\set@this@c@page}{}%
7099   }%
7100   {}%
7101   {\led@error@fail@patch@addtocounter}%
7102   \ifpackageloaded{calc}{%
7103     \apptocmd{\stepcounter}{%Because calc.sty does not use addtocounter in
stepcounter
7104       \IfStrEq{#1}{page}{\set@this@c@page}{}%
7105     }%
7106     {}%
7107     {\led@error@fail@patch@steptocounter}%
7108   }{}%
7109 }%
7110 \catcode`\#=6%
7111 %
```

`\thepar@page` The `par@page` counter.

```
7112 \newcounter{par@page}%
7113 \setcounter{par@page}{1}%
7114 %
```

## 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.

```
7115 \list@create{\labelref@list}
7116 %
```

`\zz@@@` Two convenience macros to zero three / four labeling counters in one go.

```
\zz@@@
7117 \newcommand*\zz@@@{000|000|000}% Set three counters to zero in one go
7118 \newcommand*\zz@@@{000|000|000|000}% Set four counters to zero in one go
7119
7120 %
```

`\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.

```
7121 \newcommand*\edlabel}[1]{%
7122   \leavevmode%
7123   \@bsphack%
7124   \ifboolexpr{bool{ledRcol} or bool{ledRcol@}}{%
7125     \ifXnote%
7126     \protected@write\@auxout{%
7127       {\string\l@dmake@labelsR\space\thepage|\l@dparsedstartline|\
l@dparsedstartsub|\annot@start|\the\c@pstartR|{#1}}%
7128     \ifdef{\hypertarget}%
7129     {\Hy@raisedlink{\hypertarget{#1}{}}}%
7130     }%
7131   \else%
7132     \write\linenum@outR{\string\@lab}%
7133     \ifx\labelref@listR\empty%
7134     \xdef\label@refs{\zz@@@}%
```

<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](mailto:jdb43@cam.ac.uk)) via the `ctt` thread “ledmac cross referencing”, 25 August 2003.

```

7135 \else%
7136 \gl@p\labelref@listR\to\label@refs%
7137 \fi%
7138 \ifvmode%
7139 \advancelabel@refs%
7140 \fi%
7141 %

```

Use code from the kernel `\label` command to write the correct page number. Also define an `hypertarget` if `hyperref` package is loaded.

```

7142 \protected@write\@auxout{}%
7143 {\string\l@dmake@labelsR\space\thepage|\label@refs|\the\c@pstartR
|{#1}}%
7144 \ifdef{\hypertarget}%
7145 {\Hy@raisedlink{\hypertarget{#1}{}}}%
7146 {}%
7147 \fi%
7148 }{%
7149 \ifXnote@%
7150 \ifl@dpairing%,pstart or pstartL?
7151 \protected@write\@auxout{}%
7152 {\string\l@dmake@labels\space\thepage|\l@dparsedstartline|\
l@dparsedstartsub||\annot@start|\the\c@pstartL|{#1}}%
7153 \ifdef{\hypertarget}%
7154 {\Hy@raisedlink{\hypertarget{#1}{}}}%
7155 {}%
7156 \else%
7157 \protected@write\@auxout{}%
7158 {\string\l@dmake@labels\space\thepage|\l@dparsedstartline|\
l@dparsedstartsub||\annot@start|\the\c@pstart|{#1}}%
7159 \ifdef{\hypertarget}%
7160 {\Hy@raisedlink{\hypertarget{#1}{}}}%
7161 {}%
7162 \fi%
7163 \else%
7164 \write\linenum@out{\string\@lab}%
7165 \ifx\labelref@list\empty%
7166 \xdef\label@refs{\zz@@@}%
7167 \else%
7168 \gl@p\labelref@list\to\label@refs%
7169 \fi%
7170 \ifvmode%
7171 \advancelabel@refs%
7172 \fi%
7173 \ifl@dpairing%,Pstart or PstartL?
7174 \protected@write\@auxout{}%
7175 {\string\l@dmake@labels\space\thepage|\label@refs|\the\c@pstartL
|{#1}}%
7176 \ifdef{\hypertarget}%
7177 {\Hy@raisedlink{\hypertarget{#1}{}}}%
7178 {}%
7179 \else%
7180 \protected@write\@auxout{}%
7181 {\string\l@dmake@labels\space\thepage|\label@refs|\the\c@pstart

```

```

|{#1}}%
7182     \ifdef{\hypertarget}%
7183       {\Hy@raisedlink{\hypertarget{#1}{}}}%
7184       }%
7185     \fi%
7186   \fi%
7187 }%
7188 \@esphack}%
7189
7190 %

```

`\advancelabel@refs` In cases where `\edlabel` is the first element in a paragraph, we have a problem with line counts, because line counts change only at the first horizontal box of the paragraph. Hence, we need to test `\edlabel` if it occurs at the start of a paragraph. To do so, we use `\ifvmode`. If the test is true, we must advance by one unit the amount of text we write into the `.aux` file. We do so with the `\advancelabel@refs` command.

```

7191 \newcounter{line}%
7192 \newcounter{subline}%
7193 \newcounter{absline}%
7194 \newcommand{\advancelabel@refs}{%
7195   \setcounter{line}{\expandafter\labelrefsparseline\label@refs}%
7196   \stepcounter{line}%
7197   \setcounter{absline}{\expandafter\labelrefsparseabsline\label@refs}%
7198   \stepcounter{absline}%
7199   \ifsublines%
7200     \setcounter{subline}{\expandafter\labelrefsparsesubline\label@refs}
7201   %
7202     \stepcounter{subline}{1}%
7203     \def\label@refs{\theline|\thesubline|\theabsline}%
7204   \else%
7205     \def\label@refs{\theline|0|\theabsline}%
7206   \fi%
7207 }
7208 \def\labelrefsparseline#1|#2|#3{#1}%
7209 \def\labelrefsparsesubline#1|#2|#3{#2}%
7210 \def\labelrefsparseabsline#1|#2|#3{#3}%
7211 %

```

`\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.

```

7211 \newcommand*{\l@dmake@labels}{%
7212 \def\l@dmake@labels#1|#2|#3|#4|#5|#6|#7{%
7213   \expandafter\ifx\csname the@label\csuse{XR@prefix}#7\endcsname%
7214     \relax%
7215   \else%

```

```

7216 \led@warn@DuplicateLabel{\csuse{XR@prefix}#7}%
7217 \fi
7218 \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
7219 \csgdef{the@label\csuse{XR@prefix}#7}{#1|#2|#3|#4|#5|#6|\relax}%
7220 \ignorespaces}
7221
7222 %

```

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.

```

7223 \AtBeginDocument{%
7224 \def\l@dmake@labels#1|#2|#3|#4|#5|#6|#7{%
7225 }
7226
7227 %

```

**\@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.

```

7228
7229 \newcommand*{\@lab}{%
7230 \ifledRcol
7231 \xright@appenditem{\linenumr@p{\line@numR}}|{%
7232 \ifsublines@ \sublinenumr@p{\subline@numR}\else 0\fi|{%
7233 \the\absline@numR|{%
7234 \current@annot%
7235 }%
7236 \to\labelref@listR
7237 \else
7238 \xright@appenditem{%
7239 \linenumr@p{\line@num}}|{%
7240 \ifsublines@ \sublinenumr@p{\subline@num}\else 0\fi|{%
7241 \the\absline@num|{%
7242 \current@annot%
7243 }%
7244 \to\labelref@list
7245 \fi}
7246 %

```

**\applabel** \applabel, if called in \edtext will insert automatically both a starting and an ending label for the current edtext lines.

```

7247 \newcommand*{\applabel}[1]{%
7248 \if@edtext@secondarg%
7249 %

```



```

7287     \fi%
7288     \fi%
7289 %

```

Use the L<sup>A</sup>T<sub>E</sub>X standard hack for label.

```

7290     \@esphack%
7291 %

```

If using hyperref, store the label in a list for the current edtext, to call later an hyper-target for the end of the lemma.

```

7292     \ifdef{\hypertarget}%
7293     {%
7294         \ifcsundef{applabel@thisedtext@the\@edtext@level}{\global\
cslet{applabel@thisedtext@the\@edtext@level}{\relax}}{}%
7295         \listcsgadd{applabel@thisedtext@the\@edtext@level}{#1}%
7296     }%
7297     {}%
7298 %

```

Warning if \applabel is called outside of \edtext.

```

7299     \else%
7300         \led@warn@AppLabelOutSecondArgEdtext{#1}%
7301     \fi%
7302 %

```

End of \applabel

```

7303 }%
7304 %

```

**\edlabels** \edlabels and \edlabelE are just used to mark the beginning and the end of a passage.

```

\edlabelE
\edlabelSE
7305 \newcommand{\edlabels}[1]{%
7306     \edlabel{#1:start}%
7307 }
7308 \newcommand{\edlabelE}[1]{%
7309     \edlabel{#1:end}%
7310 }
7311 \newcommand{\edlabelSE}[1]{%
7312     \edlabels{#1}%
7313     \edlabelE{#1}%
7314 }
7315 %

```

**\wrap@edcrossref** \wrap@edcrossref is called around all reledmac crossref commands, except those which start with x. It adds the hyperlink.

```

7316 \newrobustcmd{\wrap@edcrossref}[2]{%
7317     \ifdef{\hyperlink}%
7318         {\hyperlink{#1}{#2}}%
7319         {#2}%
7320 }
7321 %

```

`\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<sup>A</sup>T<sub>E</sub>X already defines a `\pageref`, so changing the name to `\edpageref`.

```
7322 \newcommand*\edpageref}[1]{\l@dref@undefined{#1}\wrap@edcrossref{#1}{\l@
7323 \l@dgetref@num{1}{#1}}
7324
7325 %
```

`\edlineref` If the specified label exists, `\lineref` gives its line number.

`\xlineref`

```
7326 \newcommand*\edlineref}[1]{%
7327 \l@dref@undefined{#1}%
7328 \wrap@edcrossref{#1}{%
7329 \providetoggle{label@#1@ledRcol}%Required for the first run, when the
7330 label has not yet been parsed on the .aux file
7331 \iftoggle{label@#1@ledRcol}%
7332 {\linenumrepR{\l@dgetref@num{2}{#1}}}%
7333 {\linenumrep{\l@dgetref@num{2}{#1}}}%
7334 \xflagref{#1}%
7335 }%
7336 \newcommand*\xlineref}[1]{\l@dgetref@num{2}{#1}}%
7337
7338 %
```

`\sublineref` If the specified label exists, `\sublineref` gives its sub-line number.

`\xsublineref`

```
7339 \newcommand*\sublineref}[1]{%
7340 \l@dref@undefined{#1}%
7341 \wrap@edcrossref{#1}{%
7342 \providetoggle{label@#1@ledRcol}%Required for the first run, when the
7343 label has not yet been parsed on the .aux file
7344 \iftoggle{label@#1@ledRcol}%
7345 {\sublinenumrepR{\l@dgetref@num{3}{#1}}}%
7346 {\sublinenumrep{\l@dgetref@num{3}{#1}}}%
7347 }%
7348 \newcommand*\xsublineref}[1]{\l@dgetref@num{3}{#1}}%
7349
7350 %
```

`\xabslineref` If the specified label exists, `\xabslineref` gives its absolute line number. This is generally used only by `reledmac` internal macros.

```
7351 \newcommand*\xabslineref}[1]{\l@dgetref@num{4}{#1}}%
7352 %
```



`\annotationref` If the specified label exists, `\edannotationref` gives the line number annotation with which it is associated.

`\xannotationref`

```

7353 \newcommand*\annotationref}[1]{%
7354   \l@dref@undefined{#1}%
7355   \wrap@edcrossref{#1}{%
7356     \l@wrap@ifnotemptybox{\Xwraplinenumannotation@ref}{%
7357       \l@dgetref@num{5}{#1}%
7358     }%
7359   }%
7360 }%
7361 \newcommand*\xannotationref}[1]{%
7362   \l@dgetref@num{5}{#1}%
7363 }%
7364 %

```

`\pstartref` If the specified label exists, `\pstartref` gives its pstart number.

`\xpstartref`

```

7365 \newcommand*\pstartref}[1]{\l@dref@undefined{#1}\wrap@edcrossref{#1}{\
7366   \l@dgetref@num{6}{#1}}}%
7367 \newcommand*\xpstartref}[1]{\l@dgetref@num{6}{#1}}%
7368 %

```

`\xflagref` `\xflagref` finds the side flag of any ref defined with `\edlabel`.

```

7369 \newcommand*\xflagref}[1]{\l@dgetref@num{7}{#1}}%
7370 %

```

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.

```

7371 \newcommand*\l@dref@undefined}[1]{%
7372   \expandafter\ifx\csname the@label#1\endcsname\relax
7373   \led@warn@RefUndefined{#1}%
7374   \fi}
7375 %
7376 %

```

`\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.

```

7377 \newcommand*\l@dgetref@num}[2]{%
7378   \expandafter
7379   \ifx\csname the@label#2\endcsname \relax
7380     000%

```

```

7381 \else
7382 \expandafter\expandafter\expandafter
7383 \l@dlabel@parse\csname the@label#2\endcsname|#1%
7384 \fi}
7385
7386 %

```

`\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`.)

```

7387 \newcommand*\l@dlabel@parse{}
7388 \def\l@dlabel@parse#1|#2|#3|#4|#5|#6|#7|#8{%
7389 \ifcase #8%
7390 \or #1%
7391 \or #2%
7392 \or #3%
7393 \or #4%
7394 \or #5%
7395 \or #6%
7396 \or #7%
7397 \fi}
7398 %

```

`\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.

```

7399 \newcommand*\xxref}[2]{%
7400 {%
7401 \expandafter\ifx\csname the@label#1\endcsname\relax%
7402 \let\@tempa\zz@@%
7403 \def\@tempc{}%
7404 \else%
7405 \def\@tempa{%
7406 \l@dgetref@num{1}{#1}|%
7407 \l@dgetref@num{2}{#1}|%
7408 \l@dgetref@num{3}{#1}%
7409 }%
7410 \edef\@tempc{%
7411 \l@dgetref@num{5}{#1}%
7412 }%
7413 \fi%
7414 \expandafter\ifx\csname the@label#2\endcsname\relax%

```

```

7415 \let\@tempb\zz@@@%
7416 \def\@tempd{}%
7417 \else%
7418 \def\@tempb{%
7419 \l@dgetref@num{1}{#2}|%
7420 \l@dgetref@num{2}{#2}|%
7421 \l@dgetref@num{3}{#2}%
7422 }%
7423 \edef\@tempd{%
7424 \l@dgetref@num{5}{#2}%
7425 }%
7426 \fi%
7427 \global\appto\@beforeinsertofthisedtext{%
7428 \def\@this@crossref@start{#1}%
7429 }%
7430 \global\appto\@beforeinsertofthisedtext{%
7431 \def\@this@crossref@end{#2}%
7432 }%
7433 \linenum{\@tempa|\@tempb}%
7434 \edef\@tempe{\@tempc|\@tempd}%
7435 \expandafter\lineannot\expandafter{\@tempe}%
7436 }%
7437 }%
7438
7439
7440 %

```

`\appref` `\SEref`, `\apprefwithpage`, `\SErefwithpage` and `\SEonlypage` print cross-ref to some start / end lines defined by specific commands. It prints the lines as they should be printed in the apparatus (critical notes for not suffixed versions, endnotes for suffixed versions).

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`.

```

7441 \def\Xtwolines@appref{}%
7442 \def\Xtwolines@SEref{}%
7443
7444 \def\Xmorethantwolines@appref{}%
7445 \def\Xmorethantwolines@SEref{}%
7446
7447 \def\Xlinerangeseparator@appref{\endashchar}%
7448 \def\Xlinerangeseparator@SEref{\endashchar}%
7449
7450 \def\Xsublinesep@appref{\fullstop}%
7451 \def\Xsublinesep@SEref{\fullstop}%
7452
7453 \def\Xpagelinesep@appref{\fullstop}%
7454 \def\Xpagelinesep@SEref{\fullstop}%
7455
7456
7457 \newtoggle{Xtwolinesbutnotmore@appref}%
7458 \newtoggle{Xtwolinesbutnotmore@SEref}%
7459

```

```

7460 \newtoggle{Xtwolinesonlyinsamepage@appref}%
7461
7462 \newtoggle{Xtwolinesonlyinsamepage@SEref}%
7463
7464 \newtoggle{Xlineflag@appref}%
7465 \toggletrue{Xlineflag@appref}%Here exception
7466 \newtoggle{Xlineflag@SEref}%
7467 \toggletrue{Xlineflag@SEref}%%Here exception
7468
7469 \newtoggle{Xlinenumannotationonlyfirst@SEref}%Never changed, just for the
test in \setprintlines
7470 \newtoggle{Xlinenumannotationonlyfirst@appref}%Idem
7471 \newtoggle{Xnolinenumber@SEref}%Idem
7472 \newtoggle{Xnolinenumber@appref}%Idem
7473 \newtoggle{Xnolinenumberifannotation@SEref}%Idem
7474 \newtoggle{Xnolinenumberifannotation@appref}%Idem
7475
7476
7477 \def\Xlinenumannotationposition@appref{after}%
7478 \def\Xlinenumannotationposition@SEref{after}%
7479
7480 \def\Xwraplinenumannotation@appref{\textsuperscript}%
7481 \def\Xwraplinenumannotation@SEref{\textsuperscript}%
7482 \newtoggle{Xnoidenticallinenumannotation@appref}%
7483 \newtoggle{Xnoidenticallinenumannotation@SEref}%
7484
7485 \def\Xendtwolines@apprefwithpage{}%
7486 \def\Xendtwolines@SErefwithpage{}%
7487
7488 \def\Xendmoreethantwolines@apprefwithpage{}%
7489 \def\Xendmoreethantwolines@SErefwithpage{}%
7490
7491 \def\Xendlinerangeseparator@apprefwithpage{\endashchar}
7492 \def\Xendlinerangeseparator@SErefwithpage{\endashchar}
7493 \def\Xendlinerangeseparator@SErefonlypage{\endashchar}
7494
7495 \def\Xendbeforepagenumber@apprefwithpage{p.}%
7496 \def\Xendbeforepagenumber@SErefwithpage{p.}%
7497 \def\Xendbeforepagenumber@SEonlypage{p.}%
7498
7499 \def\Xendafterpagenumber@apprefwithpage{ }%
7500 \def\Xendafterpagenumber@SErefwithpage{ }%
7501
7502
7503 \def\Xendlineprefixsingle@apprefwithpage{}%
7504 \def\Xendlineprefixsingle@SErefwithpage{}%
7505
7506 \def\Xendlineprefixmore@apprefwithpage{}%
7507 \def\Xendlineprefixmore@SErefwithpage{}%
7508
7509 \newtoggle{Xendtwolinesbutnotmore@apprefwithpage}%
7510 \newtoggle{Xendtwolinesbutnotmore@SErefwithpage}%
7511
7512 \def\Xendsublinesep@apprefwithpage{\fullstop}%

```

```

7513 \def\Xendsublinesep@SErefwithpage{\fullstop}%
7514
7515 \newtoggle{Xendtwolinesonlyinsamepage@apprefwithpage}%
7516 \newtoggle{Xendtwolinesonlyinsamepage@SErefwithpage}%
7517
7518 \newtoggle{Xendlineflag@apprefwithpage}
7519 \toggletrue{Xendlineflag@apprefwithpage}%Here, exception
7520 \newtoggle{Xendlineflag@SErefwithpage}
7521 \toggletrue{Xendlineflag@SErefwithpage}%Here, exception
7522
7523
7524 \def\Xendlinenumannotationposition@apprefwithpage{after}%
7525 \def\Xendlinenumannotationposition@SErefwithpage{after}%
7526
7527 \def\Xendlinenumannotationposition@apprefwithpage{after}%
7528 \def\Xendlinenumannotationposition@apprefwithpage@SErefwithpage{after}%
7529
7530 \def\Xendwraplinenumannotation@apprefwithpage{\textsuperscript}%
7531 \def\Xendwraplinenumannotation@SErefwithpage{\textsuperscript}%
7532
7533 \newtoggle{Xendnoidenticallinenumannotation@apprefwithpage}%
7534 \newtoggle{Xendnoidenticallinenumannotation@SErefwithpage}%
7535
7536
7537 \newtoggle{Xendnolinenumber@SErefwithpage}%Never changed, only for test in
  \setprintendlines
7538 \newtoggle{Xendnolinenumber@apprefwithpage}%Idem
7539 \newtoggle{Xendnolinenumberifannotation@SErefwithpage}%Idem
7540 \newtoggle{Xendnolinenumberifannotation@apprefwithpage}%Idem
7541
7542 %

```

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.

```

7543
7544 \newtoggle{Xendlinenumannotationonlyfirst@SEref}%Never changed, just for
  the test in \setprintendlines
7545 \newtoggle{Xendlinenumannotationonlyfirst@SErefwithpage}%Idem
7546 \newtoggle{Xendlinenumannotationonlyfirst@appref}%Idem
7547 \newtoggle{Xendlinenumannotationonlyfirst@apprefwithpage}%Idem
7548
7549 \gdef\Xboxstartlinenum@appref{Opt}
7550 \gdef\Xboxstartlinenum@SEref{Opt}
7551 \newtoggle{Xnotboxingsubline@appref}%
7552 \newtoggle{Xnotboxingsubline@SEref}%
7553
7554 \gdef\Xboxendlinenum@appref{Opt}
7555 \gdef\Xboxendlinenum@SEref{Opt}
7556
7557 \gdef\Xendboxstartlinenum@apprefwithpage{Opt}
7558 \gdef\Xendboxstartlinenum@SErefwithpage{Opt}
7559 \newtoggle{Xendnotboxingsubline@apprefwithpage}%
7560 \newtoggle{Xendnotboxingsubline@SErefwithpage}%

```

```

7561 \gdef\Xendboxendlinenum@apprefwithpage{Opt}
7562 \gdef\Xendboxendlinenum@SErefwithpage{Opt}
7563
7564
7565 \newtoggle{Xendpagenumberonlyfirst@apprefwithpage}
7566 \newtoggle{Xendpagenumberonlyfirst@SErefwithpage}
7567
7568 \newtoggle{Xendpagenumberonlyfirstifsingle@apprefwithpage}
7569 \newtoggle{Xendpagenumberonlyfirstifsingle@SErefwithpage}
7570
7571 \newtoggle{Xendpagenumberonlyfirstintwo@apprefwithpage}
7572 \newtoggle{Xendpagenumberonlyfirstintwo@SErefwithpage}
7573
7574 \gdef\Xendsympagenum@apprefwithpage{}
7575 \gdef\Xendsympagenum@SErefwithpage{}
7576
7577 \gdef\Xendinplaceofpagenumber@apprefwithpage{}
7578 \gdef\Xendinplaceofpagenumber@SErefwithpage{}
7579
7580 %

```

Now, declare the default values of \@apprefprefixsingle and \@apprefprefixmore, \@SErefprefix, \@SErefprefixmore and the commands which defines them.

```

7581 \newcommand\@apprefprefixsingle{}%
7582 \newcommand\@SErefprefixsingle{}%
7583
7584 \newcommand\@apprefprefixmore{}%
7585 \newcommand\@SErefprefixmore{}%
7586
7587 \newcommand{\setapprefprefixsingle}[1]{%
7588   \gdef\@apprefprefixsingle{#1}%
7589 }
7590 \newcommand{\setSErefprefixsingle}[1]{%
7591   \gdef\@SErefprefixsingle{#1}%
7592 }
7593
7594 \newcommand{\setapprefprefixmore}[1]{%
7595   \gdef\@apprefprefixmore{#1}%
7596 }
7597 \newcommand{\setSErefprefixmore}[1]{%
7598   \gdef\@SErefprefixmore{#1}%
7599 }
7600
7601 %

```

And not \setSErefonlypageprefixsingle and \setSErefonlypageprefixmore.

```

7602 \newcommand{\setSErefonlypageprefixsingle}[1]{%
7603   \gdef\SErefonlypage@prefixsingle{#1}%
7604 }%
7605 \newcommand{\setSErefonlypageprefixmore}[1]{%
7606   \gdef\SErefonlypage@prefixmore{#1}%
7607 }%
7608 %

```

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.

```

7609
7610 \newcommandx{\appref}[2][1,usedefault]{\reformatted@{#1}{#2}{appref}}
7611 \newcommandx{\SEref}[2][1,usedefault]{\reformatted@{#1}{#2}{SEref}}
7612
7613 \newcommandx{\apprefwithpage}[2][1,usedefault]{\reformattedwithpage@
7614 {#1}{#2}{appref}}
7615 \newcommandx{\SErefwithpage}[2][1,usedefault]{\reformattedwithpage@
7616 {#1}{#2}{SEref}}
7617
7618 \newcommand{\reformatted@}[3]{%
7619   \def\do##1{%
7620     \setkeys[mac]{truefootnoteoption}{##1}%
7621   }%
7622   \notblank{#1}{\docsvlist{#1}}{ }%
7623   \xdef\currentseries{#3}%
7624   \iftoggle{noprefix@}{ }%
7625   {%
7626     \ifcsemt{#3prefixmore}%
7627     {\@apprefprefixsingle}%
7628     {%
7629       \IfEq{\xlineref{#2:start}}{\xlineref{#2:end}}%
7630       {\iftoggle{prefixmore@}%
7631         {\csuse{#3prefixmore}}%
7632         {\csuse{#3prefixsingle}}%
7633       }%
7634       {\csuse{#3prefixmore}}%
7635     }%
7636   }%
7637   \ifboolexpr{%
7638     test{\ifcsundef{the@label#2:start}}%
7639     or test{\ifcsundef{the@label#2:end}}%
7640   }%
7641   {\led@warn@pairRefUndefined{#2}\nfss@text{\reset@font\bfseries ??}}%
7642   {%
7643     \def\@this@crossref@start{#2:start}%
7644     \def\@this@crossref@end{#2:end}%
7645     \xdef\annot@start{\xannotationref{#2:start}}%
7646     \xdef\annot@end{\xannotationref{#2:end}}%
7647     \printlines\xpageref{#2:start}|\xlineref{#2:start}|\xsublineref{#2:
7648 start}|\xpageref{#2:end}|\xlineref{#2:end}|\xsublineref{#2:end}|\relax|\
7649 xflagref{#2:start}|%
7650   }%
7651   \def\do##1{%
7652     \setkeys[mac]{falsefootnoteoption}{##1}%
7653   }%
7654   \notblank{#1}{\docsvlist{#1}}{ }%

```

```

7653 }%
7654
7655 \newcommand{\reformattedwithpage@}[3]{%
7656   \def\do##1{%
7657     \setkeys[mac]{truefootnoteoption}{##1}%
7658   }%
7659   \notblank{#1}{\docsvlist{#1}}{}%
7660   \xdef\@currentseries{#3withpage}%
7661   \ifboolexpr{%
7662     test{\ifcsundef{the@label#2:start}}%
7663     or test{\ifcsundef{the@label#2:end}}%
7664   }%
7665   {\led@warn@pairRefUndefined{#2}\nfss@text{\reset@font\bfseries ??}}%
7666   {%
7667     \def\@this@crossref@start{#2:start}%
7668     \def\@this@crossref@end{#2:end}%
7669     \xdef\annot@start{\xannotationref{#2:start}}%
7670     \xdef\annot@end{\xannotationref{#2:end}}%
7671     \printendlines\xpageref{#2:start}|\xlineref{#2:start}|\xsublineref{#2:
start}|\xpageref{#2:end}|\xlineref{#2:end}|\xsublineref{#2:end}|\relax|\
xflagref{#2:start}|%
7672   }%
7673   \def\do##1{%
7674     \setkeys[mac]{falsefootnoteoption}{##1}%
7675   }%
7676   \notblank{#1}{\docsvlist{#1}}{}%
7677 }%
7678
7679 \newcommand{\reformattedonlypage@}[3]{%
7680   \def\do##1{%
7681     \setkeys[mac]{truefootnoteoption}{##1}%
7682   }%
7683   \notblank{#1}{\docsvlist{#1}}{}%
7684   \xdef\@currentseries{#3onlypage}%
7685   \ifboolexpr{%
7686     test{\ifcsundef{the@label#2:start}}%
7687     or test{\ifcsundef{the@label#2:end}}%
7688   }%
7689   {\led@warn@pairRefUndefined{#2}\nfss@text{\reset@font\bfseries ??}}%
7690   {\ifnumequal{\xpageref{#2:end}}{\xpageref{#2:start}}%
7691     {%
7692       \ifcsvoid{#3onlypage@prefixsingle}%
7693       }%
7694     {\csletcs{Xendbeforepagenumber@#3onlypage}{#3onlypage@prefixsingle
}}%
7695     \printnpnum{%
7696       \wrap@edcrossref{#2:start}{\xpageref{#2:start}}%
7697     }%
7698   }%
7699   {%
7700     \ifcsvoid{#3onlypage@prefixmore}%
7701     }%
7702     {\csletcs{Xendbeforepagenumber@#3onlypage}{#3onlypage@prefixmore}}%
7703   \ifdefined\linrangesep%

```



```

7704     \printnpnum{%
7705         \wrap@edcrossref{#2:start}{\xpageref{#2:start}}%
7706         \linerrangesep@%
7707         \wrap@edcrossref{#2:end}{\xpageref{#2:end}}%
7708     }%
7709     \else%
7710         \printnpnum{%
7711             \wrap@edcrossref{#2:start}{\xpageref{#2:start}}%
7712             \cuse{Xendlinerrangeseparator@{\currentseries}}%
7713             \wrap@edcrossref{#2:end}{\xpageref{#2:end}}%
7714         }%
7715         \fi%
7716     }%
7717 }%
7718 \def\do##1{%
7719     \setkeys[mac]{falsefootnoteoption}{##1}%
7720 }%
7721 \notblank{#1}{\docsvlist{#1}}{ }%
7722 }%
7723 %

```

`\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`.

```

7724 \newcommand*{\edmakelabel}[2]{\expandafter\xdef\csname the@label#1\
endcsname{#2}}
7725
7726 %

```

(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. 140 and V.9 p. 105), 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`.

```

7727 \pretocmd{\XR@test}%
7728     {\XR@test@mac+++#1#2#3#4+++}%
7729     {}%
7730     {}%
7731 %

```

`\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`.

```
7732 \long\def\xR@test@mac+++#1+++{\XR@test@mac@test#1}
7733 %
```

`\XR@test@mac@test` And finally, `\XR@test@mac@test` does the job. This code is based on the `\XR@test` macro of the `xr` package. However, note that the `\XR@prefix` is not called here, but it is integrated directly in `\l@dmake@labels` and `\l@dmake@labelsR`.

```
7734 \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
7735 \ifx#1\l@dmake@labels%
7736 \l@dmake@labels#2%
7737 \else
7738 \ifx#1\l@dmake@labelsR%
7739 \l@dmake@labelsR #2%
7740 \fi%
7741 \fi%
7742 }%
7743 %
```

## 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.

```
7744 \pretocmd{\@xympar}%
7745 {\ifnumberedpar@
7746 \led@warn@NoMarginpars
7747 \@esphack
7748 \else}%
7749 {}%
7750 {}%
7751
7752 \apptocmd{\@xympar}%
7753 {\fi}%
7754 {}
7755 {}
7756
7757 %
```

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@dmakesidenote@margin` returns the number associated to sidenote margin:

**left:** 0

**right:** 1

**outer:** 2

**inner:** 3

```

7758 \newcount\sidenote@margin
7759 \newcommand*\sidenotemargin}[1]{%
7760   \l@dgetsidenote@margin{#1}%
7761   \ifnum\l@dtempcntb>\m@ne
7762     \ifledRcol
7763       \global\sidenote@marginR=\l@dtempcntb
7764     \else
7765       \global\sidenote@margin=\l@dtempcntb
7766     \fi
7767   \fi}}
7768 \newcommand*\l@dgetsidenote@margin}[1]{%
7769   \def\@tempa{#1}\def\@tempb{left}%
7770   \ifx\@tempa\@tempb
7771     \l@dtempcntb \z@
7772   \else
7773     \def\@tempb{right}%
7774     \ifx\@tempa\@tempb
7775       \l@dtempcntb \@ne
7776     \else
7777       \def\@tempb{outer}%
7778       \ifx\@tempa\@tempb
7779         \l@dtempcntb \tw@
7780       \else
7781         \def\@tempb{inner}%
7782         \ifx\@tempa\@tempb
7783           \l@dtempcntb \thr@@
7784         \else
7785           \led@warn@BadSidenotemargin
7786           \l@dtempcntb \m@ne
7787         \fi
7788       \fi
7789     \fi
7790   \fi}
7791 \sidenotemargin{right}
7792
7793 %

```

`\l@dlp@rbox` We need two boxes to store sidenote texts.

`\l@drp@rbox`

```

7794 \newbox\l@dlp@rbox
7795 \newbox\l@drp@rbox
7796
7797 %

```

`\ledlsnotewidth` `\ledrsnotewidth` These specify the width of the left/right boxes (initialised to `\marginparwidth`), their distance from the text (initialised to `\linenumsep`), and the fonts used.

`\ledlsnotesep`

`\ledrsnotesep`

```

7798 \newdimen\ledlsnotewidth \ledlsnotewidth=\marginparwidth

```

`\ledlsnotefontsetup`

`\ledrsnotefontsetup`

```

7799 \newdimen\ledrsnotewidth \ledrsnotewidth=\marginparwidth
7800 \newdimen\ledlsnotesep \ledlsnotesep=\linenumsep
7801 \newdimen\ledrsnotesep \ledrsnotesep=\linenumsep
7802 \newcommand*\ledlsnotefontsetup{\raggedleft\footnotesize}
7803 \newcommand*\ledrsnotefontsetup{\raggedright\footnotesize}
7804
7805 %

```

`\ledleftnote` `\ledrightnote`, `\ledinnernote`, `\ledouternote` are the user commands for left, right, inner and outer sidenotes. The two last ones are just alias for the two first ones, depending on the page number. `\ledsidenote{<text>}` is the command for a moveable sidenote.

`\ledsidenote` All these macro call the macro `\ledsidenote@`, as they have the same structure.

```

\ledsidenote@
7806 \newcommand*\ledleftnote}[1]{%
7807   \ledsidenote@{\l@dlsnote{#1}}%
7808 }%
7809 \newcommand*\ledrightnote}[1]{%
7810   \ledsidenote@{\l@drsnote{#1}}%
7811 }%
7812 \newcommand*\ledsidenote}[1]{%
7813   \ledsidenote@{\l@dcsnote{#1}}%
7814 }%
7815 \newcommand*\ledinnernote}[1]{%
7816   \ledsidenote@{\l@disnote{#1}}%
7817 }%
7818 \newcommand*\ledouternote}[1]{%
7819   \ledsidenote@{\l@dosnote{#1}}%
7820 }%
7821 \newcommand*\ledsidenote@[1]{%
7822   \edtext{#1}%
7823   \advance\edtext@notes@start by 1%
7824   #1%
7825 }%+
7826 }%
7827 %

```

`\l@dlsnote` . The “footnotes” for left, right, and moveable sidenotes. The whole scheme is reminiscent of the critical footnotes code.

```

\l@drsnote
\l@dcsnote
\l@desnote
\l@disnote
7828 \newif\ifrightnoteup
7829 \rightnoteuptrue
7830
7831 \newcommand*\l@dlsnote}[1]{%
7832   \begingroup%
7833   \newcommand{\content}{#1}%
7834   \ifnumberedpar@
7835     \ifledRcol%
7836       \xright@appenditem{\noexpand\l@dlsnote{\expandonce\content}}%
7837         \to\inserts@listR
7838       \global\advance\insert@countR \@ne%
7839     \else%
7840       \xright@appenditem{\noexpand\l@dlsnote{\expandonce\content}}%
7841         \to\inserts@list

```

```

7842     \global\advance\insert@count \@ne%
7843     \fi
7844 \fi%
7845 \ignorespaces%
7846 \endgroup%
7847 }%
7848
7849 \newcommand*\l@drsnote}[1]{%
7850   \begingroup%
7851   \newcommand{\content}{#1}%
7852   \ifnumberedpar@
7853     \ifledRcol%
7854       \xright@appenditem{\noexpand\vl@drsnote{\expandonce\content}}%
7855         \to\inserts@listR
7856       \global\advance\insert@countR \@ne%
7857     \else%
7858       \xright@appenditem{\noexpand\vl@drsnote{\expandonce\content}}%
7859         \to\inserts@list
7860       \global\advance\insert@count \@ne%
7861     \fi
7862   \fi\ignorespaces%
7863   \endgroup%
7864 }%
7865
7866 \newcommand*\l@dcsnote}[1]{%
7867   \begingroup%
7868   \newcommand{\content}{#1}%
7869   \ifnumberedpar@
7870     \ifledRcol%
7871       \xright@appenditem{\noexpand\vl@dcsnote{\expandonce\content}}%
7872         \to\inserts@listR
7873       \global\advance\insert@countR \@ne%
7874     \else%
7875       \xright@appenditem{\noexpand\vl@dcsnote{\expandonce\content}}%
7876         \to\inserts@list
7877       \global\advance\insert@count \@ne%
7878     \fi
7879   \fi\ignorespaces%
7880   \endgroup%
7881 }%
7882
7883 \newcommand*\l@disnote}[1]{%
7884   \begingroup%
7885   \newcommand{\content}{#1}%
7886   \ifnumberedpar@%
7887     \ifledRcol%
7888       \xright@appenditem{\noexpand\vl@disnote{\expandonce\content}}%
7889         \to\inserts@listR%
7890       \global\advance\insert@countR \@ne%
7891     \else%
7892       \xright@appenditem{\noexpand\vl@disnote{\expandonce\content}}%
7893         \to\inserts@list%
7894       \global\advance\insert@count \@ne%
7895     \fi%

```

```

7896 \fi\ignorespaces%
7897 \endgroup%
7898 }%
7899
7900 \newcommand*{\l@dosnote}[1]{%
7901 \begingroup%
7902 \newcommand{\content}{#1}%
7903 \ifnumberedpar@%
7904 \ifledRcol%
7905 \xright@appenditem{\noexpand\l@dosnote{\expandonce\content}}%
7906 \to\inserts@listR%
7907 \global\advance\insert@countR \@ne%
7908 \else%
7909 \xright@appenditem{\noexpand\l@dosnote{\expandonce\content}}%
7910 \to\inserts@list%
7911 \global\advance\insert@count \@ne%
7912 \fi%
7913 \fi\ignorespaces%
7914 \endgroup%
7915 }%
7916
7917 %

```

`\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` `\vl@dosnote` 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 on the page.

```

7918 \newcommand*{\vl@dlsnote}[1]{%
7919 \ifledRcol%
7920 \@l@dttempcntb=\sidenote@marginR%
7921 \ifnum\@l@dttempcntb>\@ne%
7922 \advance\@l@dttempcntb by\page@numR%
7923 \fi%
7924 \else%
7925 \@l@dttempcntb=\sidenote@margin%
7926 \ifnum\@l@dttempcntb>\@ne%
7927 \advance\@l@dttempcntb by\page@num%

```

```

7928     \fi%
7929 \fi%
7930 \ifodd\@l@tempcntb%
7931   \listgadd{\l@dcsnotetext@l}{#1}%
7932 \else%
7933   \listgadd{\l@dcsnotetext}{#1}%
7934 \fi
7935 }
7936 \newcommand*\vl@drsnote}[1]{%
7937   \ifledRcol%
7938     \@l@tempcntb=\sidenote@marginR%
7939     \ifnum\@l@tempcntb>\@ne%
7940       \advance\@l@tempcntb by\page@numR%
7941     \fi%
7942 \else%
7943   \@l@tempcntb=\sidenote@margin%
7944   \ifnum\@l@tempcntb>\@ne%
7945     \advance\@l@tempcntb by\page@num%
7946   \fi%
7947 \fi%
7948 \ifodd\@l@tempcntb%
7949   \listgadd{\l@dcsnotetext}{#1}%
7950 \else%
7951   \listgadd{\l@dcsnotetext@r}{#1}%
7952 \fi%
7953 }
7954 \newcommand*\vl@dcsnote}[1]{\listgadd{\l@dcsnotetext}{#1}}
7955
7956 \newcommand{\vl@disnote}[1]{%
7957   \ifledRcol%
7958     \@tempcnta=\page@numR%
7959   \else%
7960     \@tempcnta=\page@num%
7961   \fi%
7962   \ifodd\@tempcnta% ODD => right page => inner side = left side
7963     \vl@dlsnote{#1}%
7964   \else%
7965     \vl@drsnote{#1}%
7966   \fi%
7967 }%
7968
7969 \newcommand{\vl@dosnote}[1]{%
7970   \ifledRcol%
7971     \@tempcnta=\page@numR%
7972   \else%
7973     \@tempcnta=\page@num%
7974   \fi%
7975   \ifodd\@tempcnta% ODD => right page => outer side = right side
7976     \vl@drsnote{#1}%
7977   \else%
7978     \vl@dlsnote{#1}%
7979   \fi%
7980 }%
7981

```

7982 %

`\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.

```
7983 \newcommand*{\setl@dlp@rbox}[1]{%
7984   \begingroup%
7985     \parindent\z@\hsize=\ledlsnotewidth%
7986     \ledlsnotefontsetup%We kept it outside of the vbox, because can affect
the ragging
7987     \global\setbox\l@dlp@rbox%
7988     \ifleftnoteup%
7989       =\vbox to\z@{\ledlsnotefontsetup\vss #1}}%We put \
ledlsnotefontsetup inside footnote because required for color command. Note
the {} to keep setting local.
7990     \else%
7991       =\vbox to 0.70\baselineskip{\ledlsnotefontsetup\strut#1\vss}}%
7992     \fi%
7993   \endgroup%
7994 }
7995
7996 \newcommand*{\setl@drp@rbox}[1]{%
7997   \begingroup%
7998     \parindent\z@\hsize=\ledrsnotewidth%
7999     \ledrsnotefontsetup%We kept it outside of the vbox, because can affect
the ragging
8000     \global\setbox\l@drp@rbox%
8001     \ifrighnoteup%
8002       =\vbox to\z@{\ledrsnotefontsetup\vss#1}}%We put \ledrsnotefontsetup
inside footnote because required for color command. Note the {} to keep
setting local.
8003     \else%
8004       =\vbox to0.7\baselineskip{\ledrsnotefontsetup\strut#1\vss}}%
8005     \fi%
8006   \endgroup%
8007 }%
8008 \newif\ifleftnoteup
8009 \leftnoteuptrue
8010 %
```

`\@sidenoteseq` This macro is used to separate sidenotes of the same line.

```
8011 \newcommand{\setsidenoteseq}[1]{\gdef\@sidenoteseq{#1}}
8012 \newcommand{\@sidenoteseq}{, }
8013 %
```

`\affixside@note` This macro puts any moveable sidenote text into the left or right sidenote box, depending on which margin it is meant to go in. It's a very much stripped down version of `\affixline@num`.

Before do it, we concatenate all moveable sidenotes of the line, using `\@sidenoteseq` as separator. It is the result that we put on the sidenote.



```

8014 \newcommand*\affixside@note}{%
8015   \prepare@edindex@fornote{\the\page@num|\the\line@num|\the\subline@num|\
the\page@num|\the\line@num|\the\subline@num]}%
8016   \def\sidenotecontent@{}%
8017   \numgdef\itemcount@{0}%
8018   \def\do##1{%
8019     \ifnumequal{\itemcount@}{0}%
8020     {%
8021     \appto\sidenotecontent@{##1}}% Not print not separator before
the 1st note
8022     {\appto\sidenotecontent@{\@sidenotesep ##1}%
8023     }%
8024     \numgdef\itemcount@{\itemcount@+\@ne}%
8025   }%
8026   \dolistloop{\l@dcsnotetext}%
8027   \ifnumgreater\itemcount@{1}{\led@err@ManySidenotes}{}%
8028 %

```

And we do the same for left and right notes (not movable).

```

8029 \gdef\@templ@d{%
8030 \gdef\@templ@n{\l@dcsnotetext\l@dcsnotetext@1\l@dcsnotetext@r}%
8031 \ifx\@templ@d\@templ@n \else%
8032 \iftwocolumn%
8033 \iffirstcolumn%
8034 \setl@dlp@rbox{##1}{\sidenotecontent@}%
8035 \else%
8036 \setl@drp@rbox{\sidenotecontent@}%
8037 \fi%
8038 \else%
8039 \@l@dttempcntb=\sidenote@margin%
8040 \ifnum\@l@dttempcntb>\@ne%
8041 \advance\@l@dttempcntb by\page@num%
8042 \fi%
8043 \ifodd\@l@dttempcntb%
8044 \setl@drp@rbox{\sidenotecontent@}%
8045 \gdef\sidenotecontent@{}%
8046 \numgdef\itemcount@{0}%
8047 \dolistloop{\l@dcsnotetext@l}%
8048 \ifnumgreater\itemcount@{1}{\led@err@ManyLeftnotes}{}%
8049 \setl@dlp@rbox{\sidenotecontent@}%
8050 \else%
8051 \setl@dlp@rbox{\sidenotecontent@}%
8052 \gdef\sidenotecontent@{}%
8053 \numgdef\itemcount@{0}%
8054 \dolistloop{\l@dcsnotetext@r}%
8055 \ifnumgreater\itemcount@{1}{\led@err@ManyRightnotes}{}%
8056 \setl@drp@rbox{\sidenotecontent@}%
8057 \fi%
8058 \fi%
8059 \fi%
8060 \advance\@edindex@fornote@m@ne%
8061 }
8062 %

```

## XXVII Minipages and such

We can put footnotes into minipages. The preparatory code has been set up earlier, all that remains is to ensure that it is available inside a minipage box. This requires some alteration to the kernel code, specifically the `\@iiiminipage` and `\endminipage` macros. We will arrange this so that additional series can be easily added.

`\l@dfheetbeginmini` These will be the hooks in `\@iiiminipage` and `\endminipage`.  
`\l@dfheetendmini` They can be extended to handle other things if necessary.

```

8063 \ifnoledgroup@else%
8064 \newcommand*{\l@dfheetbeginmini}{\@ledgrouptrue\l@dedbeginmini\
l@dfambeginmini}
8065 \newcommand*{\l@dfheetendmini}{%
8066   \IfStrEq{critical-familiar}{\@mpfnpos}%
8067     {\l@dedendmini\l@dfamendmini}%
8068     {%
8069       \IfStrEq{familiar-critical}{\@mpfnpos}%
8070         {\l@dfamendmini\l@dedendmini}%
8071         {\do@feet@custom@order{mp@}{\@mpfnpos}}}%
8072   }%
8073 }%
8074 %

```

`\l@dedbeginmini` These handle the initiation and closure of critical footnotes in a minipage environment.  
`\l@dedendmini`  
`\mp@append@Xnotes`

```

8075 \newcommand*{\l@dedbeginmini}{%
8076   \unless\ifnocritical@%
8077     \def\do##1{%
8078       \csletcs{v##1footnote}{mpv##1footnote}%
8079     }%
8080     \dolistloop{\@series}%
8081   \fi%
8082 }
8083 \newcommand*{\l@dedendmini}{%
8084   \unless\ifnocritical@%
8085     \ifl@dpairing%
8086       \ifledRcol%
8087         \flush@notesR%
8088       \else%
8089         \flush@notes%
8090       \fi%
8091     \fi
8092     \def\do##1{%
8093       \mp@append@Xnotes{##1}%
8094     }%
8095     \dolistloop{\@series}%
8096   \fi%
8097 }%
8098 \newcommand{\mp@append@Xnotes}[1]{%
8099 \ifvoid\csuse{mp#1footins}\else%
8100 \ifl@dpairing%
8101 \ifparledgroup%
8102 \ifledRcol%

```

```

8103 \dingdef{\parledgroup@beforenotesR}{\parledgroup@beforenotesR+\skip
\@nameuse{mp#1footins}}%
8104 \else%
8105 \dingdef{\parledgroup@beforenotesL}{\parledgroup@beforenotesL+
skip\@nameuse{mp#1footins}}%
8106 \fi%
8107 \fi%
8108 \fi%
8109 \ifcsstring{series@display#1}{paragraph}{-}{%
8110 \setbox\@nameuse{mp#1footins}=\vbox{%
8111 \cuse{Xnotefontsize@#1}%
8112 \ifcsdef{Xhsize\cuse{series@display#1}@#1}{%
8113 \hsize\cuse{Xhsize\cuse{series@display#1}@#1}%
8114 }-}{%
8115 \noindent\cuse{Xtxtbeforenotes@#1}%
8116 \unvbox\@nameuse{mp#1footins}%
8117 \@parboxrestore%
8118 }%
8119 }%
8120 \cuse{mp#1footgroup}{#1}%
8121 \fi%
8122 }%
8123 %

```

`\l@dfambeginmini` These handle the initiation and closure of familiar footnotes in a minipage environment.

```

\l@dfamendmini
\mp@append@notesX
8124 \newcommand*{\l@dfambeginmini}{%
8125 \unless\ifnofamiliar%
8126 \def\do##1{\csletcs{vfootnote##1}{mpvfootnote##1}}%
8127 \dolistloop{\@series}%
8128 \fi%
8129 }%
8130
8131 \newcommand*{\l@dfamendmini}{%
8132 \unless\ifnofamiliar%
8133 \def\do##1{%
8134 \mp@append@notesX{##1}%
8135 }%
8136 \dolistloop{\@series}%
8137 \fi%
8138 }%
8139 \newcommand{\mp@append@notesX}[1]{%
8140 \ifvoid\cuse{mpfootins#1}\else%
8141 \ifcsstring{series@displayX#1}{paragraph}{-}{%
8142 \setbox\@nameuse{mpfootins#1}=\vbox{%
8143 \cuse{notefontsizeX@#1}%
8144 \ifcsdef{hsize\cuse{series@display#1}X@#1}{%
8145 \hsize\cuse{hsize\cuse{series@display#1}X@#1}%
8146 }-}{%
8147 \noindent\cuse{txtbeforenotesX@#1}%
8148 \unvbox\@nameuse{mpfootins#1}%
8149 \@parboxrestore%
8150 }%
8151 }%

```

```

8152 \csuse{mpfootgroup#1}{#1}%
8153 \fi%
8154 }%
8155 %

```

`\@iiiminipage` This is our extended form of the kernel `\@iiiminipage` defined in `ltboxes.dtx`.

```

8156 \patchcmd%
8157 {\@iiiminipage}%
8158 {\let\@footnotetext\@mpfootnotetext}%
8159 {\let\@footnotetext\@mpfootnotetext\l@dfetbeginmini}%
8160 {}%
8161 {\led@error@fail@patch@\@iiiminipage}%
8162 %

```

`\endminipage` This is our extended form of the kernel `\endminipage` defined in `ltboxes.dtx`.

```

8163 \patchcmd%
8164 {\endminipage}%
8165 {\footnoterule}%
8166 {\footnoterule\l@advance@parledgroup@beforenormalnotes}%
8167 {}%
8168 {\led@error@fail@patch@endminipage}
8169
8170 \patchcmd%
8171 {\endminipage}%
8172 {\@minipagefalse}%
8173 {\l@dfetendmini\@minipagefalse}%
8174 {}%
8175 {\led@error@fail@patch@endminipage}
8176
8177 %

```

`\l@dunboxmpfoot` `\@l@dunboxmpfoot` insert normal footnotes for `ledgroup`.  
`\l@advance@parledgroup@beforenormalnotes`

```

8178 \newcommand*{\l@dunboxmpfoot}{%
8179 \vskip\skip\@mpfootins
8180 \normalcolor
8181 \footnoterule
8182 \l@advance@parledgroup@beforenormalnotes
8183 \unvbox\@mpfootins%
8184 }
8185 %

```

When using parallel `ledgroup`, we need to store the vertical space added before footnote, in order to compensate them between left and right pages.

```

8186 \newcommand{\l@advance@parledgroup@beforenormalnotes}{%
8187 \ifparledgroup
8188 \ifl@dpairing
8189 \ifledRcol
8190 \dingdef{\parledgroup@beforenotesR}{\parledgroup@beforenotesR+
skip\@mpfootins}
8191 \else

```

```

8192     \dingdef{\parledgroup@beforenotesL}{\parledgroup@beforenotesL+\
skip\@mpfootins}
8193     \fi
8194     \fi
8195     \fi
8196   }
8197 %

```

`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.

```

8198
8199 \newenvironment{ledgroup}{%
8200   \resetprevpage@num%
8201   \def\@mpfn{mpfootnote}\def\thempfn{\thempfootnote}\c@mpfootnote\z@%
8202   \let\@footnotetext\@mpfootnotetext
8203   \l@dfetbeginmini%
8204 }{%
8205   \par
8206   \unskip
8207   \ifvoid\@mpfootins\else
8208     \l@dunboxmpfoot
8209   \fi
8210   \l@dfetendmini%
8211   \@ledgroupfalse%
8212 }
8213
8214
8215 %

```

`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.

```

8216 \newenvironment{ledgroupsize}[2][1]{%
8217 %

```

Set the various text measures.

```

8218   \hsize #2\relax
8219 %

```

Initialize fills for centering.

```

8220   \let\ledllfill\hfil
8221   \let\ledrlfill\hfil
8222   \def\@tempa{#1}\def\@tempb{1}%
8223 %

```

Left adjusted numbered lines

```

8224     \ifx\@tempa\@tempb
8225     \let\ledllfill\relax
8226   \else
8227     \def\@tempb{r}%

```

```
8228 \ifx\@tempa\@tempb
8229 %
```

Right adjusted numbered lines

```
8230 \let\ledrlfill\relax
8231 \fi
8232 \fi
8233 %
```

Set up the footnoting.

```
8234 \def\@mpfn{mpfootnote}\def\thempfn{\thempfootnote}\c@mpfootnote\z@
8235 \let\@footnotetext\@mpfootnotetext
8236 \l@dfteetbeginmini%
8237 }{
8238 \par
8239 \unskip
8240 \ifvoid\@mpfootins\else
8241 \l@dunboxmpfoot
8242 \fi
8243 \l@dfteetendmini%
8244 }
8245 %
8246 %
```

Close the \ifnoledgroup@\else.

```
8247 \fi%
8248 %
```

`\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`.

```
8249 \newif\ifledgroupnotesL@
8250 \newif\ifledgroupnotesR@
8251 %
```

## 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`.

```
8252 \AtBeginDocument{%
8253 \unless\ifl@imakeidx%
8254 \ifpackageloaded{imakeidx}{\led@error@PackageAfterEledmac{imakeidx}}{}
8255 %
8256 \fi%
8257 \unless\ifl@indextools%
8258 \ifpackageloaded{indextools}{\led@error@PackageAfterEledmac{indextools}}{}%
8259 \fi%
```

```

8259 \unless\ifl@footmisc%
8260 \ifpackageloaded{footmisc}{\led@error@PackageAfterEledmac{footmisc}}{}
%
8261 \fi%
8262 }
8263 %

```

## XXVIII.2 Auxiliary macros for `\edindex`

`\pagelinesep` In order to get a correct line number we have to use the label/ref mechanism. These macros are for that.

```

\edindexlab
\c@labidx
8264 \newcommand{\pagelinesep}{-}
8265 \newcommand{\edindexlab}{$&}
8266 \newcounter{labidx}
8267 \setcounter{labidx}{0}
8268
8269 %

```

`\doedindexlabel` This macro sets an `\edlabel`.

```

8270 \newcommand{\doedindexlabel}{%
8271 \stepcounter{labidx}%
8272 \edlabel{\edindexlab\thelabidx}%
8273 }
8274
8275 %

```

`\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`.

```

8276 \newcounter{pageline}%
8277 \renewcommand{\thepageline}{%
8278 \thepage%
8279 \pagelinesep%
8280 \xlineref{\edindexlab\thelabidx}%
8281 }
8282 %

```

`\thestartpageline` These macros make up the page/line start/end number when the `\edindex` command is called in critical notes.

```

\theendpageline
8283 \newcommand{\thestartpageline}{%
8284 \l@dparsedstartpage%
8285 \pagelinesep%
8286 \l@dparsedstartline%
8287 }
8288 \newcommand{\theendpageline}{%
8289 \l@dparsedendpage%
8290 \pagelinesep%
8291 \l@dparsedendline%
8292 }
8293 %

```

### 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.

```
8294 \newcount\@edindex@fornote@
8295 %
```

`\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 |.

```
8296 \newcommand{\prepare@edindex@fornote}[1]{%
8297   \l@dp@rsefootspec#1|}%
8298   \advance\@edindex@fornote@\@ne%
8299 }
8300 %
```

`\get@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.

```
8301 \newcommand{\get@edindex@ledinnote@command}{%
8302   \ifxindy%
8303     \gdef\@ledinnote@command{%
8304       ledinnote\thelabidx%
8305     }%
8306     \ifxindyhyperref%
8307       \immediate\write\eledmac@xindy@out{%
8308         (define-attributes ("ledinnote\thelabidx"))^^J
8309         \space\space(markup-locref^^J
8310           \eledmacmarkuplocrefdepth^^J
8311           :open "\string\ledinnote[\edindexlab\thelabidx]{\@index@command
8312             }{\@currentseries}{\@currentnotetype}"^^J
8313             :close "}"^^J
8314             :attr "ledinnote\thelabidx"^^J
8315           )
8316         }%
8317       \else%
8318         \immediate\write\eledmac@xindy@out{%
8319           (define-attributes ("ledinnote\thelabidx"))^^J
8320           \space\space(markup-locref^^J
8321             \eledmacmarkuplocrefdepth^^J
8322             :open "\string\ledinnote{\@index@command}{\@currentseries}{\
8323               \@currentnotetype}"^^J
8324             :close "}"^^J
8325             :attr "ledinnote\thelabidx"^^J
8326           )
8327         }%
8328       \fi%
8329     }%
8330   }
```



If we do not use xindy option, `\ledinnote@command` will produce something like `ledinnote{formattingcommand}`.

```

8328 \else%
8329 \gdef\ledinnote@command{%
8330 ledinnote[\edindexlab\thelabidx]{\@index@command}{\@currentseries}{\@
@currentnotetype}%
8331 }%
8332 \fi%
8333 }
8334 %

```

## 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 |.

```

8335 \def\get@index@command#1|#2+{%
8336 \gdef\@index@txt{#1}%
8337 \gdef\@index@command{#2}%
8338 \xdef\@index@parenthesis{}%
8339 \IfBeginWith{\@index@command}{(}{%
8340 \StrGobbleLeft{\@index@command}{1}{\@index@command@}%
8341 \global\let\@index@command\@index@command@%
8342 \xdef\@index@parenthesis{(%}%
8343 }{}%
8344 \IfBeginWith{\@index@command}{)}{%
8345 \StrGobbleLeft{\@index@command}{1}{\@index@command@}%
8346 \global\let\@index@command\@index@command@%
8347 \xdef\@index@parenthesis{)%}%
8348 }{}%
8349 }
8350 %

```

## XXVIII.5 Code for the formatted index

`\ledinnote` `\ledinnotemark` These macros are used to specify that an index reference points to a note. Arguments of `\ledinnote` are:

- #1 (optional): the label for the hyperlink;
- #2: command applied to the number;
- #3: the note series;
- #4: the type of note (critical/familiar).
- #5: the number itself.

```

8351 \newcommandx{\ledinnote}[5][1,usedefault]{%
8352 \IfStrEq{#4}{critical}{%
8353 \def\tmp{Xinnotemark}%
8354 }{%
8355 \def\tmp{innotemarkX}%
8356 }%

```

```

8357 \ifboolexpr{%
8358   test{\ifdefequal{\iftrue}{\ifHy@hyperindex}}%
8359   or%
8360   bool {xindyhyperref@}%
8361 }%
8362 {%
8363   \csuse{#2}{\hyperlink{#1}{\csuse{\tmp @#3}{#5}}}%
8364 }%
8365 {%
8366   \csuse{#2}{\csuse{\tmp @#3}{#5}}%
8367 }%
8368 }%
8369 \newcommand{\ledinnotemark}[1]{#1\emph{n}}%
8370 %

```

### 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 \newcommandx{\@wredindex}[2][1=\expandonce\jobname,usedefault]{%#1 = the
8371 index name, #2 = the text
8372 \begingroup%
8373   \let\emph\@firstofone%
8374   \let\textbf\@firstofone%
8375   \let\textit\@firstofone%
8376   \let\textmd\@firstofone%
8377   \let\textnormal\@firstofone%
8378   \let\textrm\@firstofone%
8379   \let\textsc\@firstofone%
8380   \let\textsf\@firstofone%
8381   \let\textsl\@firstofone%
8382   \let\texttt\@firstofone%
8383   \let\textup\@firstofone%
8384   \xdef\@tmp{#2}%To be used in IfSubStr instead of #2 directly. Avoid
some expansion bugs (for example with \edindex{textsc{something}})
8385 \endgroup%
8386 \ifl@imakeidx%
8387   \ifnum\@edindex@fornote@>\z@%
8388     \IfSubStr[1]{\@tmp}{|}{\get@index@command#2+}{\get@index@command#2|+}
%
8389     \get@edindex@ledinnote@command%
8390     \expandafter\imki@wrindexentry{#1}{\@index@txt|(\@ledinnote@command
}{\thestartpageline}%
8391     \expandafter\imki@wrindexentry{#1}{\@index@txt|)\@ledinnote@command
}{\theendpageline}%

```

```

8392 \else%
8393 \get@edindex@hyperref{#2}%
8394 \imki@wrindexentry{#1}{\@index@txt\@edindex@hyperref}{\thepageline}%
8395 \fi%
8396 \else%
8397 \ifnum\@edindex@fornote@>\z0%
8398 \IfSubStr[1]{\@tmp}{|}{\get@index@command#2+}{\get@index@command#2|+}
%
8399 \get@edindex@ledinnote@command%
8400 \expandafter\protected@write\@indexfile{%
8401 {\string\indexentry{\@index@txt|(\@ledinnote@command){\thestartpageline}
8402 }%
8403 \expandafter\protected@write\@indexfile{%
8404 {\string\indexentry{\@index@txt|)\@ledinnote@command}{\theendpageline}
8405 }%
8406 \else%
8407 \protected@write\@indexfile{%
8408 {\string\indexentry{#2}{\thepageline}
8409 }%
8410 \fi%
8411 \fi%
8412 \endgroup
8413 \@esphack%
8414 }
8415 %

```

Need to add the definition of `\edindex` to `\makeindex`, and initialise `\edindex` to do nothing.

```

8416 \pretocmd{\makeindex}{%
8417 \def\edindex{%
8418 \ifboolexpr{bool{numbering} or bool{numberingR} or bool{
l@dprintingpages} or bool{l@dprintingcolumns}}{%
8419 \bsphack%
8420 \doedindexlabel%
8421 \begingroup%
8422 \@sanitize%
8423 \@wredindex%
8424 }%
8425 {%
8426 \led@warn@edinde@outsidenumbering%
8427 \index%
8428 }%
8429 }%
8430 }%
8431 }%
8432 {\led@error@fail@patch@makeindex}%
8433 \newcommand{\edindex}[1]{\bsphack\esphack}
8434 \newcommand{\dummy@edindex}[2][1=\expandonce\jobname,usedefault]{}%
8435 \newcommand{\disable@edindex}{\let\old@edindex\edindex\let\edindex\
dummy@edindex}%
8436 \newcommand{\restore@edindex}{\let\edindex\old@edindex}%
8437 %

```

## XXVIII.7 Hyperlink

`\hyperlinkformat` `\hyperlinkformat` command is to be used to have both a internal hyperlink and a format, when indexing.

```
8438 \newcommand{\hyperlinkformat}[3]{%
8439   \ifstrempy{#1}%
8440     {\hyperlink{#2}{#3}}%
8441     {\csuse{#1}{\hyperlink{#2}{#3}}}%
8442   }}
8443 %
```

`\hyperlinkR` `\hyperlinkR` command is to be used to create a internal hyperlink and `\ledRflag`, when indexing.

```
8444 \newcommand{\hyperlinkR}[2]{%
8445   \hyperlink{#1}{#2\@Rlineflag}%
8446 }%
8447 %
8448 %
```

`\hyperlinkformatR` `\hyperlinkformatR` command is to be used to create a internal hyperlink, a format and a `\@Rlineflag`, when indexing.

```
8449 \newcommand{\hyperlinkformatR}[3]{%
8450   \hyperlinkformat{#1}{#2}{#3\@Rlineflag}%
8451 }%
8452 %
8453 %
```

`\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`).

```
8454 \newcommand{\get@edindex@hyperref}[1]{%
8455 %
```

We have to disable temporary spaces to work through a `xstring` bug (or feature?)

```
8456 \edef\temp@{%
8457   \catcode`\ =9 %space need for catcode
8458   \detokenize{#1}%For active character in unicode
8459   \catcode`\ =10 % space need for catcode
8460 }%
8461 %
```

Now, we define `\@edindex@hyperref` if the hyperindex of `hyperref` is enabled.

```
8462 \ifdefequal{\iftrue}{\ifHy@hyperindex}{%
8463   \IfSubStr{\temp@}{|}%
8464     {\get@index@command#1+%
8465     \ifledRcol%
8466       \gdef\@edindex@hyperref{|\@index@parenthesis %space kept
8467       hyperlinkformatR{\@index@command}%
8468       {\edindexlab\thelabidx}}%
8469     \else%
8470       \gdef\@edindex@hyperref{|\@index@parenthesis %space kept
```

```

8471     hyperlinkformat{\@index@command}%
8472     {\edindexlab\thelabidx}}%
8473     \fi%
8474   }%
8475   {\get@index@command#1|+%
8476     \ifledRcol%
8477     \gdef\@edindex@hyperref{|hyperlinkR{\edindexlab\thelabidx}}%
8478     \else%
8479     \gdef\@edindex@hyperref{|hyperlink{\edindexlab\thelabidx}}%
8480     \fi%
8481   }%
8482 }%
8483 %

8484 % If we use both xindy and hyperref, first get the \protect\cs{
index@command} command.
8485 % Then define \protect\cs{@edindex@hyperref} in the form \verb+eledmacXXX+
8486 %   \begin{macrocode}
8487 {\ifxindyhyperref%
8488   \IfSubStr{\temp@}{|}%
8489   {\get@index@command#1+%
8490   {\get@index@command#1|+%
8491   \gdef\@edindex@hyperref{|eledmac\thelabidx}}%
8492 %

```

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`.

```

8493   \IfStrEq{\@index@parenthesis}{(}%
8494   {%
8495     \csxdef{xindyparenthesis@\@index@txt}{\thelabidx}%
8496     \gdef\@edindex@hyperref{| (eledmac\thelabidx}%
8497   }%
8498   {}%
8499 %

```

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`.

```

8500   \IfStrEq{\@index@parenthesis}{)}%
8501   {%
8502     \xdef\@edindex@hyperref{|)eledmac\csuse{xindyparenthesis@\@index@txt}}%
8503     \global\csundef{xindyparenthesis@\@index@txt}%
8504   }%
8505 %

```

Write in the `.xdy` file the attributes of the location.

```

8506   {%
8507     \immediate\write\eledmac@xindy@out{%
8508       (define-attributes ("eledmac\thelabidx"))^^J
8509     \space\space(markup-locref^^J
8510     \eledmacmarkuplocrefdepth^^J

```

```

8511         :open "\string\hyperlink%
8512             \ifledRcol R\fi%
8513             {\edindexlab\thelabidx}%
8514             {\ifdefempty{\@index@command}%
8515              {}}%
8516             {\@backslashchar\@index@command}%
8517             {"^^J
8518         :close "}"^^J
8519         :attr "eledmac\thelabidx"^^J
8520     )
8521 }%
8522 }%
8523 %

```

And now, in any other case.

```

8524     \else%
8525     \gdef\@index@txt{#1}%
8526     \gdef\@edindex@hyperref{}%
8527     \fi%
8528 }%
8529 }
8530 %

```

### 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 on the option of the `indextools` package. Its only argument is the note series.

```

8531 \newcommand{\led@set@index@fornote}[1]{%
8532   \ifbool{indtl@innote}%
8533     {\let\index\nindex}%
8534     {}}%
8535   \ifbool{indtl@notenumber}%
8536     {%
8537     \renewcommand{\index}[2][\indtl@jobname]{%
8538       \orig@@index[##1]{%
8539         ##2|innotenumber{\this@footnoteX@reading}}%
8540       }%
8541     }%
8542   }%
8543   {}}%
8544 }%
8545 %

```

`\led@reinit@index@fornote` The `\led@reinit@index@fornote` just reset the default value of `\index`.

```

8546 \newcommand{\led@reinit@index@fornote}{%
8547   \ifbool{indtl@innote}%
8548     {\let\index\orig@@index}%
8549     {}}%
8550   \ifbool{indtl@notenumber}%

```

```

8551 {\let\index\orig@index}%
8552 {}%
8553 }%
8554 %

```

## XXIX Glossaries

Here, we define the `\gls`-like commands prefixed by `ed`, only if the package `glossaries` is loaded.

```

8555 \AtBeginDocument{%
8556   \ifpackageloaded{glossaries}{%
8557     \ifpackageloaded{glossaries-extra}{\
8558       led@error@require@glossariesextra}%
8559     \newif\ifl@noexpands@edgl%A boolean switched to true in \no@expands
8560   }%

```

First those which arguments are [*options*]{*label*}[*insert*].

```

8560 \gdef\edglsomo{}%
8561 \listgadd{\edglsomo}{gls}%
8562 \listgadd{\edglsomo}{Gls}%
8563 \listgadd{\edglsomo}{GLS}%
8564 \listgadd{\edglsomo}{glspl}%
8565 \listgadd{\edglsomo}{Glspl}%
8566 \listgadd{\edglsomo}{GLSpl}%
8567 \listgadd{\edglsomo}{glstext}%
8568 \listgadd{\edglsomo}{Glstext}%
8569 \listgadd{\edglsomo}{GLStext}%
8570 \listgadd{\edglsomo}{Glsfirst}%
8571 \listgadd{\edglsomo}{GLSfirst}%
8572 \listgadd{\edglsomo}{glsplural}%
8573 \listgadd{\edglsomo}{Glsplural}%
8574 \listgadd{\edglsomo}{GLSplural}%
8575 \listgadd{\edglsomo}{glsfirstplural}%
8576 \listgadd{\edglsomo}{Glsfirstplural}%
8577 \listgadd{\edglsomo}{GLSfirstplural}%
8578 \listgadd{\edglsomo}{glsname}%
8579 \listgadd{\edglsomo}{Glsname}%
8580 \listgadd{\edglsomo}{GLSname}%
8581 \listgadd{\edglsomo}{glssymbol}%
8582 \listgadd{\edglsomo}{Glsymbol}%
8583 \listgadd{\edglsomo}{GLSsymbol}%
8584 \listgadd{\edglsomo}{glsdesc}%
8585 \listgadd{\edglsomo}{Glsdesc}%
8586 \listgadd{\edglsomo}{GLSdesc}%
8587 \listgadd{\edglsomo}{glsuseri}%
8588 \listgadd{\edglsomo}{Glsuseri}%
8589 \listgadd{\edglsomo}{GLSuseri}%
8590 \listgadd{\edglsomo}{glsuserii}%
8591 \listgadd{\edglsomo}{Glsuserii}%
8592 \listgadd{\edglsomo}{GLSuserii}%
8593 \listgadd{\edglsomo}{glsuseriii}%
8594 \listgadd{\edglsomo}{Glsuseriii}%

```

```

8595 \listgadd{\edglsomo}{GLSuseriii}%
8596 \listgadd{\edglsomo}{glsuseriv}%
8597 \listgadd{\edglsomo}{Glsuseriv}%
8598 \listgadd{\edglsomo}{GLSuseriv}%
8599 \listgadd{\edglsomo}{glsuserv}%
8600 \listgadd{\edglsomo}{Glsuserv}%
8601 \listgadd{\edglsomo}{GLSuserv}%
8602 \listgadd{\edglsomo}{glsuservi}%
8603 \listgadd{\edglsomo}{Glsuservi}%
8604 \listgadd{\edglsomo}{GLSuservi}%
8605 \renewcommand{\do}[1]{%
8606 \expandafter\DeclareRobustCommandx\csname ed#1\endcsname[3][1,3,
usedefault]{%
8607 \doedindexlabel%
8608 \ifl@noexpands@edgl%
8609 \csname#1\endcsname[counter=pageline,noindex,##1]{##2}{##3}%
8610 \else%
8611 \ifdef{\hypertarget}{%
8612 \Hy@raisedlink@left{\hypertarget{pageline.\thepageline}{}}%
8613 }{}%
8614 \csname#1\endcsname[counter=pageline,##1]{##2}{##3}%
8615 \fi%
8616 }%
8617 \expandafter\WithSuffix\expandafter\DeclareRobustCommandx\csname ed
#1\endcsname*[3][1,3,usedefault]{%
8618 \doedindexlabel%
8619 \ifl@noexpands@edgl%
8620 \csname#1\endcsname*[counter=pageline,noindex,##1]{##2}{##3}%
8621 \else%
8622 \csname#1\endcsname*[counter=pageline,##1]{##2}{##3}%
8623 \ifdef{\hypertarget}{%
8624 \Hy@raisedlink@left{\hypertarget{pageline.\thepageline}{}}%
8625 }{}%
8626 \fi%
8627 }%
8628 }%
8629 \dolistloop{\edglsomo}%
8630 %

```

Then those which arguments are [*options*]{*label*}{*link text*}.

```

8631 \gdef\edglsomm{}%
8632 \listgadd\edglsomm{glsdisp}%
8633 \listgadd\edglsomm{glslink}%
8634 \renewcommand{\do}[1]{%
8635 \expandafter\DeclareRobustCommandx\csname ed#1\endcsname[3][1,
usedefault]{%
8636 \doedindexlabel%
8637 \ifl@noexpands@edgl%
8638 \csname#1\endcsname[counter=pageline,noindex,##1]{##2}{##3}%
8639 \else%
8640 \ifdef{\hypertarget}{%
8641 \Hy@raisedlink@left{\hypertarget{pageline.\thepageline}{}}%
8642 }{}%
8643 \csname#1\endcsname[counter=pageline,##1]{##2}{##3}%

```



```

8644     \fi%
8645   }%
8646   \expandafter\WithSuffix\expandafter\DeclareRobustCommand\csname ed
#1\endcsname*[3][1,usedefault]{%
8647     \doedindexlabel%
8648     \ifl@noexpands@edgl%
8649     \csname#1\endcsname*[counter=pageline,noindex,##1]{##2}{##3}%
8650   \else%
8651     \ifdef{\hypertarget}{%
8652       \Hy@raisedlink@left{\hypertarget{pageline.\thepageline}{}}%
8653     }{%
8654       \csname#1\endcsname*[counter=pageline,##1]{##2}{##3}%
8655     \fi%
8656   }%
8657 }%
8658 \dolistloop{\edglsomm}%
8659 %

```

Then those which arguments are [*options*]{*label*}.

```

8660 \gdef\edglsom{%
8661 \listgadd\edglsom{glsadd}%
8662 \renewcommand{\do}[1]{%
8663   \expandafter\DeclareRobustCommand\csname ed#1\endcsname [2][1,
usedefault]{%
8664     \doedindexlabel%
8665     \ifl@noexpands@edgl%
8666     \csname#1\endcsname[counter=pageline,noindex,##1]{##2}%
8667   \else%
8668     \ifdef{\hypertarget}{%
8669       \Hy@raisedlink@left{\hypertarget{pageline.\thepageline}{}}%
8670     }{%
8671       \csname#1\endcsname[counter=pageline,##1]{##2}%
8672     \fi%
8673   }%
8674   \expandafter\WithSuffix\expandafter\DeclareRobustCommand\csname ed
#1\endcsname*[2][1,usedefault]{%
8675     \doedindexlabel%
8676     \ifl@noexpands@edgl%
8677     \csname#1\endcsname*[counter=pageline,noindex,##1]{##2}%
8678   \else%
8679     \ifdef{\hypertarget}{%
8680       \Hy@raisedlink@left{\hypertarget{pageline.\thepageline}{}}%
8681     }{%
8682       \csname#1\endcsname*[counter=pageline,##1]{##2}%
8683     \fi%
8684   }%
8685 }%
8686 \dolistloop{\edglsom}%
8687 %

```

\l@noexpands@edgl is a macro to switch \ifl@noexpands@edgl to true in \no@expands. Its definition is made dynamically.

```
\l@noexpands@edgl88 \def\l@noexpands@edgl{\l@noexpands@edgltrue}%
```

```

8689 }{\let\l@noexpands@edgl\relax}%
8690 }%
8691 %

```

## 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.

```

8692 \def\@hangingsymbol{
8693 \newcommand*\sethangingsymbol}[1]{%
8694 \gdef\@hangingsymbol{#1}%
8695 }%
8696 \newif\ifinstanza
8697 %

```

`\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.

```

8698 \newif\ifinserthangingsymbol
8699 \newcommand{\inserthangingsymbol}{%
8700 \ifinserthangingsymbol%
8701 \ifinstanza%
8702 \@hangingsymbol%
8703 \fi%
8704 \fi%
8705 }
8706 %

```

### 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.

```

8707 \newcommand*\ampersand{\char` \&}
8708
8709 %

```

### 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.

```
8710 \chardef\body=\catcode`\@
8711 \catcode`\@=11
8712 \chardef\next=\catcode`\&
8713 \catcode`\&=\active
8714
8715 %
```

### 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.

```
8716 \newcount\stanza@count
8717 \newlength{\stanzaindentbase}
8718 \setlength{\stanzaindentbase}{20pt}
8719
8720 %
```

`\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.

```
8721 \def\strip@szacnt#1,#2|{\def\@tempb{#1}\def\@tempa{#2|}}
8722 \newcommand*\setstanzavalues[2]{\def\@tempa{#2,|}%
8723 \stanza@count\z@
8724 \def\next{\expandafter\strip@szacnt\@tempa
8725 \ifx\@tempb\empty\let\next\relax\else
8726 \expandafter\mathchardef\csname #1@\number\stanza@count
8727 @\endcsname\@tempb\relax
8728 \advance\stanza@count\@ne\fi\next}%
8729 \next}
8730
8731 %
```

`\setstanzaindents` In the original EDMAC, `\setstanzavalues{sza}{⟨...⟩}` had to be called to set the in-  
`\setstanzapenalties` dents, and similarly `\setstanzavalues{szp}{⟨...⟩}` to set the penalties. `\setstanzaindents` and `\setstanzapenalties` macros are a convenience to give the user one less thing to worry about (misspelling the first argument).

```
8732 \newcommand*\setstanzaindents[1]{\setstanzavalues{sza}{#1}}
8733 \newcommand*\setstanzapenalties[1]{\setstanzavalues{szp}{#1}}
8734 %
8735 %
```

`\managestanza@modulo` Since the `stanzaindentsrepetition` counter can be used when the indentation is repeated every  $n$  verses.

The `startstanzaindentsrepetition` counter tells in which line of verse the repetition of indent starts.

The `\managestanza@modulo` macro modifies the counter `stanza@modulo`. The command adds 1 to `stanza@modulo`, but if `stanza@modulo` is equal to the `stanzaindentsrepetition` counter, the command restarts it.

```

8736 \newcounter{stanzaindentsrepetition}
8737 \newcounter{startstanzaindentsrepetition}%
8738 \setcounter{startstanzaindentsrepetition}{1}%
8739 \newcount\stanza@modulo
8740
8741 \newcommand*\managestanza@modulo[0]{%
8742   \advance\stanza@modulo\@ne%
8743   \ifnum\stanza@modulo>\value{stanzaindentsrepetition}%
8744     \stanza@modulo\@ne%
8745   \fi%
8746 }
8747 %

```

`\stanzaindent` The macro `\stanzaindent`, when called at the beginning of a verse, changes the indentation normally defined for this verse by `\setstanzaindent`. The starred version `\stanzaindent*` skips the current verse for the repetition of stanza indent.

```

8748 \newcommand{\stanzaindent}[1]{%
8749   \hspace{\dimexpr#1\stanzaindentbase-\parindent\relax}%
8750   \ignorespaces%
8751 }%
8752 \WithSuffix\newcommand\stanzaindent*[1]{%
8753   \stanzaindent{#1}%
8754   \global\advance\stanza@modulo-\@ne%
8755   \ifnum\stanza@modulo=0%
8756     \global\stanza@modulo=\value{stanzaindentsrepetition}%
8757   \fi%
8758   \ignorespaces%
8759 }%
8760 %

```

### XXX.5 Numbering stanza

Here, macro for numbering stanza. First, the stanza counter.

```

\thestanza 8761 \newcounter{stanza}
8762 \renewcommand{\thestanza}{%
8763   \textbf{\arabic{stanza}}%
8764 }
8765 %

```

`\ifnumberstanza` Then, macro to activate automatically numbering of stanza.

```

8766 \newif\ifnumberstanza%
8767 %

```

`\@insertstanzanumber` Now, macro called at the first line of of verse of a stanza.

```

8768 \newcommand{\@insertstanzanumber}[0]{%
8769   \ifnumberstanza%
8770   \ifl@dpairing%
8771     \ifledRcol%
8772     \stanzanumwrapper{\thestanzaR}%
8773   \else%
8774     \stanzanumwrapper{\thestanzaL}%
8775   \fi%
8776 \else%
8777   \stanzanumwrapper{\thestanza}%
8778 \fi%
8779 \setline{1}%
8780 \fi%
8781 }%
8782 %

```

`\@advancestanzanumber` Also a command to advance the counter of stanza.

```

8783 \newcommand{\@advancestanzanumber}[0]{%
8784   \ifnumberstanza%
8785   \ifl@dpairing%
8786     \ifledRcol%
8787     \addtocounter{stanzaR}{1}%
8788   \else%
8789     \addtocounter{stanzaL}{1}%
8790   \fi%
8791 \else%
8792   \addtocounter{stanza}{1}%
8793 \fi%
8794 \fi%
8795 }%
8796 %

```

`\stanzanumwrapper` And finally, the wrapper for stanza number

```

8797 \newcommand{\stanzanumwrapper}[1]{%
8798   \flagstanza{#1}%
8799 }%
8800 %

```

## XXX.6 Stanza number in note

Here, the command called when printing stanza number in notes.

```

8801 \newcommand{\printstanza}[0]{%
8802   \ifboolexpr{bool{l@dpairing} or bool{l@dprintingpages} or bool{
8803     \ifledRcol%
8804     \thestanzaR%
8805   \else%
8806     \thestanzaL%
8807   \fi%

```

```

8808 }{ %
8809   \thestanza %
8810 } %
8811 }
8812 %

```

### 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.

```

8813 \newcommandx{\stanza@line}[2][1,2,usedefault]{ %
8814   \stanza@indent %
8815   \pstart[#1][#2]\stanza@hang\ignorespaces %
8816 } %
8817 \newcommand{\stanza@indent}{ %
8818   \ifboolexpr{ %
8819     test {\ifnumequal{\value{stanzaindentsrepetition}}{0}} %
8820     or %
8821     test {\ifnumless{\number\stanza@count}{\value{ %
8822       startstanzaindentsrepetition}}}} %
8823     \ifcsdef{sza@\number\stanza@count @} %
8824       { %
8825         \parindent=\csname sza@\number\stanza@count @\endcsname\ %
8826         stanzaindentbase %
8827         } %
8828         \led@err@StanzaIndentNotDefined %
8829         } %
8830     } %
8831     \ifcsdef{sza@\number\stanza@modulo @} %
8832       \parindent=\csname sza@ %
8833         \the\numexpr %
8834           \stanza@modulo %
8835           - 1 %
8836           + \value{startstanzaindentsrepetition}} %
8837       \relax %
8838       @\endcsname\stanzaindentbase %
8839       \managestanza@modulo %
8840     } %
8841     { %
8842       \led@err@StanzaIndentNotDefined %
8843       } %
8844   } %
8845 } %
8846 \def\stanza@hang{ %

```

```

8847 \leavevmode\startlock%
8848 \hangindent\ifRTL-\fi\expandafter%
8849 \csname sza@0@\endcsname\stanzaindentbase%
8850 \hangafter\@ne%
8851 }%
8852
8853 \def\sza@penalty{\count@\csname szp@\number\stanza@count @\endcsname
8854 \ifnum\count@>@\M\advance\count@-\@M\penalty-\else
8855 \penalty\fi\count@}
8856 %

```

**\@startstanza** Now we have the components of the `\stanza` macro, which appears at the start of a group of lines. This macro initializes the count and checks to see if hanging indentation and penalties are to be included. Hanging indentation suspends the line count, so that the enumeration is by verse line rather than by print line. If the print line count is desired, invoke `\let\startlock\relax` and do the same for `\endlock`. Here and above we have used `\xdef` to make the stored macros take up a bit less space, but it also makes them more obscure to the reader. Lines of the stanza are delimited by ampersands `&`. The last line of the stanza must end with `\&`.

```

8857 \xdef\@startstanza[#1][#2]{%
8858 \noexpand\instanzatrue\expandafter
8859 \begingroup%
8860 \catcode`\noexpand\&\active%
8861 \unexpanded{\newdimen\parindent@beforestanza%
8862 \parindent@beforestanza=\parindent}%Keep in memory the standard \
parindent
8863 \global\stanza@count\@ne\stanza@modulo\@ne
8864 \noexpand\ifnum\expandafter\noexpand\csname sza@0@\endcsname=\z@%
8865 \def\noexpand\stanza@hang{\unexpanded{\leavevmode\null}}%
8866 \let\noexpand\endlock\relax%
8867 \noexpand\else%
8868 \interlinepenalty\@M%
8869 \unexpanded{%
8870 \ifboolexpr{bool {RTL} and bool{xetex}}{%
8871 \leftskip\z@ plus 1fil\relax%
8872 }{%
8873 \rightskip\z@ plus 1fil\relax%
8874 }%
8875 }%
8876 \noexpand\fi%
8877 \noexpand\ifnum\expandafter\noexpand\csname szp@0@\endcsname=\z@%
8878 \let\noexpand\sza@penalty%
8879 \relax\noexpand\fi%
8880 \def\noexpand&{%
8881 \noexpand\newverse[] []}%
8882 \def\noexpand\&{\noexpand\@stopstanza}%
8883 \noexpand\@advancestanza@number%
8884 \noexpand\stanza@line[#1][#2]%
8885 \noexpand\@insertstanza@number%
8886 \let\par\relax\ignorespaces%No paragraph in verses
8887 }
8888
8889 \newcommandx{\stanza}[2][1,2,usedefault]{%

```

```

8890 \ifboolexpr{%
8891   not test{\ifdefvoid{\at@every@stanza}}%
8892   and test{\ifstrempy{#1}}%
8893   and test{\ifstrempy{#2}}%
8894   {\@startstanza[] [\at@every@stanza] \@start@every@stanza}%
8895   {\@startstanza[#1] [#2] \@start@every@stanza}%
8896 }%
8897
8898 \newcommandx{\@stopstanza}[2][1,2,usedefault]{%
8899   \unskip%
8900   \endlock%
8901   \ifboolexpr{%
8902     not test{\ifdefvoid{\at@every@stop@stanza}}%
8903     and test{\ifstrempy{#1}}%
8904     and test{\ifstrempy{#2}}%
8905     {\before@every@stop@stanza\pend[] [\at@every@stop@stanza]}%
8906     {\before@every@stop@stanza\pend[#1] [#2]}%
8907   \endgroup%
8908   \instanzafalse%
8909   \global\undef\parindent@beforestanza%
8910 }
8911
8912 \newcommand{\AtEveryStopStanza}[1]{%
8913   \ifstrempy{#1}%
8914     {\gdef\at@every@stop@stanza{}}%
8915     {\gdef\at@every@stop@stanza{\noindent#1}}%
8916 }%
8917 \WithSuffix\newcommand\AtEveryStopStanza*[1]{%
8918   \ifstrempy{#1}%
8919     {\gdef\at@every@stop@stanza{}}%
8920     {\gdef\at@every@stop@stanza{#1}}%
8921 }%
8922 \def\at@every@stop@stanza{}%
8923
8924 \newcommand{\AtEveryStanza}[1]{%
8925   \ifstrempy{#1}%
8926     {\gdef\at@every@stanza{}}%
8927     {\gdef\at@every@stanza{\noindent#1}}%
8928 }%
8929 \WithSuffix\newcommand\AtEveryStanza*[1]{%
8930   \ifstrempy{#1}%
8931     {\gdef\at@every@stanza{}}%
8932     {\gdef\at@every@stanza{#1}}%
8933 }%
8934
8935
8936
8937 \newcommand{\AtStartEveryStanza}[1]{%
8938   \ifstrempy{#1}%
8939     {\gdef\at@start@every@stanza{}}%
8940     {\gdef\at@start@every@stanza{#1}}%
8941 }%
8942 \def\at@start@every@stanza{}%
8943

```



```

8944 \newcommand{\BeforeEveryStopStanza}[1]{%
8945   \ifstrempy{#1}%
8946     {\gdef\before@every@stop@stanza{}}%
8947     {\gdef\before@every@stop@stanza{#1}}%
8948 }%
8949 \def\before@every@stop@stanza{}%
8950
8951 \newcommandx*\newverse}[4][1,2,3,4,usedefault]{%
8952   \unskip%
8953   \endlock\pend[#1][#3]\sza@penalty\global%
8954   \advance\stanza@count\@ne\stanza@line[#2][#4]%
8955   }
8956
8957 %

```

`\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`.

```

8958 \newcommand*\flagstanza}[2][\stanzaindentbase]{%
8959   \hskip -#1\llap{#2}\hskip #1\ignorespaces}
8960
8961 %

```

## 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.

```

8962 \catcode`\&=\next
8963 \catcode`\@=\body
8964 \setstanzavalues{szp}{0}
8965
8966 %

```

## XXX.9 Dealing with antilabe

`\firsthalf@antilabe@wd` If the `antilabe` option is enabled, after having vboxed the content of `\pstart...pend` structure, we get the width of the last line of the `\pend`.

```

\firsthalf@antilabe@wd
\antilabe
8967 \ifantilabe@%
8968   \newdimen\firsthalf@antilabe@wd%
8969   \newcommand\get@firsthalf@antilabe@wd{%
8970     \ifboolexpr{bool {instanza} or bool {inastanzaL} or bool {inastanzaR}}{
%
8971       \vbadness=10000%
8972       \splittopskip=\z@%
8973       \ifl@dpairing%
8974         \ifledRcol%
8975         \setbox0=\copy\csname l@dRcolrawbox\the\l@dnumpstartsR\endcsname%

```

```

8976     \else%
8977         \setbox0=\copy\csname l@dLcolrawbox\the\l@dnumpstartsL\endcsname%
8978     \fi%
8979     \else%
8980         \setbox0=\copy\raw@text%
8981     \fi%
8982     \dimen0=\dimexpr\ht0+\dp0-\baselineskip\relax%
8983     \ifdim\dimen0>0pt%
8984         \setbox1=\vsplit0 to \dimen0%
8985     \fi%
8986     \setbox0=\vbox{\unvbox0\global\setbox1=\lastbox}%
8987     \setbox1=\hbox{\noindent\unhbox1}%
8988     \global\firsthalf@antilabe@wd=\wd1%
8989 }{}%
8990 }%
8991 \def\beforeantilabe{}%
8992 \def\afterantilabe{}%
8993 \newcommand{\antilabe}[1]{%
8994     \beforeantilabe\hbox to\firsthalf@antilabe@wd{#1\hfill}\afterantilabe%
8995 }%
8996 \else%
8997     \let\get@firsthalf@antilabe@wd\relax%
8998 \fi%
8999 %

```

## 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
9000 \def\msdata@c{}%
9001 \def\msdata@cR{}%
9002 \newcommand{\msdata}[1]{%
9003     \leavevmode%
9004     \unless\ifstopmsdata@inserted@%
9005         \stopmsdata%
9006         \led@warning@msdatawithoutstop%
9007     \fi%
9008     \global\stopmsdata@inserted@false%
9009     \unless\ifledRcol%
9010         \numgdef{\msdata@c}{\msdata@c+1}%
9011         \ifdef{\hypertarget}{%
9012             \edlabel{\msdata@c:start:msdata}%
9013         }{}%
9014         \protected@write\linenum@out{}{%
9015             \string\@msd{#1}%
9016         }%
9017     \else%
9018         \numgdef{\msdata@cR}{\msdata@cR+1}%
9019         \ifdef{\hypertarget}{%
9020             \edlabel{\msdata@cR:start:msdata}%

```

```

9021   }{}%
9022   \protected@write\linenum@outR{}{%
9023     \string\@msd{#1}%
9024   }%
9025   \fi%
9026 }%
9027 %

```

**\stopmsdata** The user level `\stopmsdata` command only writes information about the end of manuscripts data in numbered auxiliary file.

```

9028 \newcommand{\stopmsdata}[0]{%
9029   \leavevmode%
9030   \unless\ifledRcol%
9031     \protected@write\linenum@out{}{%
9032       \string\@stopmsd%
9033     }%
9034     \ifdef{\hypertarget}{%
9035       \edlabel{\msdata@c:end:msdata}%
9036     }{}%
9037   \else%
9038     \protected@write\linenum@outR{}{%
9039       \string\@stopmsd%
9040     }%
9041     \ifdef{\hypertarget}{%
9042       \edlabel{\msdata@cR:end:msdata}%
9043     }{}%
9044   \fi%
9045   \global\stopmsdata@inserted@true%
9046 }%
9047 %

```

**\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`

```

9048 \newif\ifstopmsdata@inserted@%
9049 %

```

## 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?

```

9050 \newcommand{\setmsdataseries}[1]{%
9051   \gdef\@msdata@series{#1}%
9052 }%
9053 \def\@msdata@series{A}%
9054 %

```

**\setmsdataposition** The label for the manuscripts data.

```

9055 \def\ms@data@position{msdata-regular}%
9056 \newcommand{\setmsdataposition}[1]{%
9057   \gdef\ms@data@position{#1}%
9058 }%
9059 %

```

`\setmsdatalabel` The label for the manuscripts data.

```

9060 \def\ms@data@label{Ms.}%
9061 \newcommand{\setmsdatalabel}[1]{%
9062   \gdef\ms@data@label{#1}%
9063 }%
9064 %

```

### XXXI.3 Counters and lists

`\@msd@c` `\@msd@c` is a counter incremented at each `\@msd` read in auxiliary file.

```

9065 \numdef{\@msd@c}{0}
9066 \numdef{\@msd@cR}{0}
9067 %

```

`\add@msd@` `\add@msd@` is a counter incremented at each `\add@msd@data`, that is at each time we prepare the insertion of manuscripts data footnote.

```

9068 \numdef{\add@msd@c}{0}%
9069 \numdef{\add@msd@cR}{0}%
9070 %

```

`\@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@msd@data` and looped on `\insert@msd@data`, then emptied.

```

9071 \def\@msdata@list{}%
9072 %

```

### XXXI.4 Auxiliary file macros

`\@msd` The `\@msd` macro is written in the auxiliary file. It just defines three macros by `\msd@data` 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.

```

9073 \newcommand{\@msd}[1]{%
9074   \unless\ifledRcol%
9075     \global\numdef{\@msd@c}{\@msd@c+\@one}%
9076     \csgdef{\@msd@data@\@msd@c @data}{#1}%
9077     \csxdef{\@msd@data@\@msd@c @linenumber}{\the\line@num}%
9078     \csxdef{\@msd@data@\@msd@c @sublinenumber}{\the\subline@num}%
9079     \csxdef{\@msd@data@\@msd@c @abslinenumber}{\the\absline@num}%
9080     \xright@appenditem{\the\absline@num}\to\actionlines@list%
9081     \xright@appenditem{-1010}\to\actions@list%
9082   \else%

```

```

9083 \global\newcommand{\@msd@cR}{\@msd@cR+\@ne}%
9084 \csgdef{\@msd@cR @dataR}{\#1}%
9085 \csxdef{\@msd@cR @linenumberR}{\the\line@numR}%
9086 \csxdef{\@msd@cR @sublinenumberR}{\the\subline@numR}%
9087 \csxdef{\@msd@cR @abslinenumberR}{\the\absline@numR}%
9088 \xright@appenditem{\the\absline@numR}\to\actionlines@listR%
9089 \xright@appenditem{-1010}\to\actions@listR%
9090 \fi%
9091 }%
9092 %

```

`\@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.

```

9093 \newcommand{\@stopmsd}[0]{%
9094   \unless\ifledRcol%
9095     \ifcsundef{\@msd@c @stoplinenumberR}{%
9096       \csxdef{\@msd@c @stopabslinenumberR}{\the\absline@numR}%
9097       \csxdef{\@msd@c @stoplinenumberR}{\the\line@numR}%
9098       \csxdef{\@msd@c @stopsublinenumberR}{\the\subline@numR}%
9099     }%
9100   \else%
9101     \ifcsundef{\@msd@cR @stoplinenumberR}{%
9102       \csxdef{\@msd@cR @stopabslinenumberR}{\the\absline@numR}%
9103       \csxdef{\@msd@cR @stoplinenumberR}{\the\line@numR}%
9104       \csxdef{\@msd@cR @stopsublinenumberR}{\the\subline@numR}%
9105     }%
9106   }%
9107 \fi%
9108 }%
9109 %

```

## 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`.

```

9110 \newcommand{\add@msdata}{%
9111   \bgroup%
9112   \normalfont%
9113   \unless\ifledRcol%
9114     \numgdef{\@add@msd@c}{\@add@msd@c+\@ne}%
9115     \ifcsdef{\@msd@c @add@msd@c @data}{%
9116       \letcs{\@data}{\@msd@c @add@msd@c @data}%
9117       \edef\l@d@nums{%
9118         000| Start page = we don't print it
9119         \csuse{\@msd@c @add@msd@c @linenumberR}| Start line number
9120         \csuse{\@msd@c @add@msd@c @sublinenumberR}| Start subline number
9121         000| End page number, we don't print it
9122         \ifnumless{\csuse{\@msd@c @stopabslinenumberR}}{\csuse{
@lastabsline@forpage@the\page@num}}%

```

```

9123     {\csuse{@msdata@add@msd@c @stoplinenumber}}%End line number if
in the same page
9124     {\csuse{@lastline@forpage@the\page@num}}%Otherwise, last
number of the page
9125     |%
9126     \ifnumless{\csuse{@msdata@add@msd@c @stopabslinenumber}}{\csuse{
@lastabsline@forpage@the\page@num}}%
9127     {\csuse{@msdata@add@msd@c @stopsublinenumber}}%End subline
number if in the same page
9128     {\csuse{@lastsubline@forpage@the\page@num}}%Otherwise, last
subnumber of the page
9129     |%
9130     \edfont@info%Font
9131     }%
9132     \@msd@options@fullpagefalse%
9133     \iffirstlineofpage%Try if the data are for the full page. If yes
, will add options to the list.
9134     \unless\if@msdata@insertedfrompreviouspage%
9135     \ifnumless{\csuse{@lastabsline@forpage@the\page@num}}{\csuse
{@msdata@add@msd@c @stopabslinenumber}+\@one}%
9136     {%
9137     \numdef{\@tmp}{\add@msd@c+\@one}%
9138     \ifcsdef{@msdata@\@tmp @abslinenumber}%
9139     {\ifnumequal{\csuse{@msdata@\@tmp @abslinenumber}}{\csuse{
@lastabsline@forpage@the\page@num}}%
9140     }%
9141     {\@msd@options@fullpagetrue}%
9142     }%
9143     {\@msd@options@fullpagetrue}%
9144     }%
9145     }%
9146     \fi%
9147     \fi%
9148     \listxadd{\@msdata@list}{%
9149     \@msd@options@iffullpage%
9150     \ifluatex%
9151     \csxdef{footnote@luatextextdir}{\the\textdir}%
9152     \csxdef{footnote@luatexpardir}{\the\pardir}%
9153     \fi%
9154     \csdef{@this@crossref@start}{\add@msd@c:start:msdata}%
9155     \csdef{@this@crossref@end}{\add@msd@c:end:msdata}%
9156     \unexpanded{%
9157     \def\annot@start{}%
9158     \def\annot@end{}%
9159     }%
9160     \noexpand\csuse{v\@msdata@series footnote}{\@msdata@series}{\
expandonce\l@d@nums}{\ms@data@label}{\expandonce\data}}%
9161     \reset@msd@options@iffullpage%
9162     }%
9163     }%
9164     {}%
9165     \else%
9166     \numgdef{\add@msd@cR}{\add@msd@cR+\@one}%
9167     \ifcsdef{@msdata@\add@msd@cR @dataR}{%

```

```

9168     \letcs{\@data}{@msdata@add@msd@cR @dataR}%
9169     \edef\l@d@nums{%
9170         000|% Start page = we don't print it
9171         \csuse{@msdata@add@msd@cR @linenumberR}|% Start line number
9172         \csuse{@msdata@add@msd@cR @sublinenumberR}|% Start subline
number, for now, not used
9173         000|% End page number, we don't print it
9174         \ifnumless{\csuse{@msdata@add@msd@cR @stopabslinenumberR}}{\
csuse{@lastline@forpageR@the\page@numR}}}%
9175         {\csuse{@msdata@add@msd@cR @stoplinenumberR}}}%End line number
if in the same page
9176         {\csuse{@lastline@forpageR@the\page@numR}}}%Otherwise, last
number of the page
9177         |%
9178         \ifnumless{\csuse{@msdata@add@msd@cR @stopabslinenumberR}}{\
csuse{@lastline@forpageR@the\page@numR}}}%
9179         {\csuse{@msdata@add@msd@cR @stopsublinenumberR}}}%End subline
number if in the same page
9180         {\csuse{@lastsubline@forpageR@the\page@numR}}}%Otherwise, last
number of the page
9181         000|% End sub line number, for now, not used
9182         \edfont@info%Font
9183     }%
9184     \@msd@options@fullpagefalse%
9185     \if@firstlineofpageR%
9186         \unless@if@msdata@insertedfrompreviouspage%
9187         \ifnumless{\csuse{@lastabsline@forpageR@the\page@numR}}{\
csuse{@msdata@add@msd@c @stopabslinenumberR}+\@ne}%
9188         {%
9189         \numdef{\@tmp}{\add@msd@cR+\@ne}%
9190         \ifcsdef{@msdata@\@tmp @abslinenumberR}%
9191         {\ifnumequal{\csuse{@msdata@\@tmp @abslinenumberR}}{\csuse{
@lastabsline@forpageR@the\page@numR}}}%
9192         {}%
9193         {\@msd@options@fullpagetrue}%
9194     }%
9195     {\@msd@options@fullpagetrue}%
9196 }%
9197 {}%
9198 \fi%
9199 \fi%
9200 \listxadd{\@msdata@list}{%
9201 \@msd@options@iffullpage%
9202 \ifluatex%
9203 \csxdef{footnote@luatextextdir}{\the\textdir}%
9204 \csxdef{footnote@luatexpardir}{\the\pardir}%
9205 \fi%
9206 \csdef{@this@crossref@start}{\add@msd@cR:start:msdata}%
9207 \csdef{@this@crossref@end}{\add@msd@cR:end:msdata}%
9208 \unexpanded{%
9209 \def\annot@start{}%
9210 \def\annot@end{}%
9211 }%

```

```

9212 \noexpand\csuse{v\@msdata@series footnote}{\@msdata@series}{\{
expandonce\l@d@nums}{\ms@data@label}{\expandonce\@data}}%
9213 \reset@msd@options@iffullpage%
9214 }%
9215 }%
9216 {}%
9217 \fi%
9218 \egroup%
9219 }%
9220 %

```

`\if@msdata@insertedfrompreviouspage` The `\if@msdata@insertedfrompreviouspage` boolean is set to TRUE if `reledmac` automatically inserts data from previous page in the first line of a page.

```

9221 \newif\if@msdata@insertedfrompreviouspage%
9222 %

```

`\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.

```

9223 \newcommand{\add@msdata@firstlineofpage}{%
9224 \bgroup%
9225 \normalfont%
9226 \unless\ifledRcol{%
9227 \ifcsdef{@msdata@\add@msd@c @data}{%
9228 \ifnumless{\the\absline@num-\@ne}{\csuse{@msdata@\add@msd@c
@stopabslinenumber}}}%
9229 {%
9230 \global\@msdata@insertedfrompreviouspagetrue%
9231 \letcs{\@data}{@msdata@\add@msd@c @data}%
9232 \edef\l@d@nums{%
9233 000|% Start page = we don't print it
9234 \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
9235 000|% Start subline number, for now, not used
9236 000|% End page number, we don't print it
9237 \ifnumless{\csuse{@msdata@\add@msd@c @stopabslinenumber}}{\csuse{
@lastabsline@forpage@\the\page@num}}}%
9238 {\csuse{@msdata@\add@msd@c @stoplinenumber}}%End line number if
in the same page
9239 {\csuse{@lastline@forpage@\the\page@num}}%Otherwise, last
number of the page
9240 |%
9241 000|% End sub line number, for now, not used
9242 \edef@info%Font
9243 }%
9244 \@msd@options@fullpagefalse%
9245 \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
9246 {%
9247 \numdef{\@tmp}{\add@msd@c+\@ne}%
9248 \ifcsdef{@msdata@\@tmp @abslinenumber}%

```



```

9249     {\ifnumequal{\csuse{@msdata@\tmp @abslinenumber}}{\csuse{
@lastabsline@forpage@\the\page@num}}}%
9250     {}%
9251     {\@msd@options@fullpagetrue}%
9252     }%
9253     {\@msd@options@fullpagetrue}%
9254     }%
9255     {}%
9256     \listxadd{\@msdata@list}{%
9257     \@msd@options@iffullpage%
9258     \ifluatex%
9259     \csxdef{footnote@luatextextdir}{\the\textdir}%
9260     \csxdef{footnote@luatexpardir}{\the\pardir}%
9261     \fi%
9262     \csdef{@this@crossref@start}{\add@msd@c:start:msdata}%
9263     \csdef{@this@crossref@end}{\add@msd@c:end:msdata}%
9264     \unexpanded{%
9265     \def\annot@start{}}%
9266     \def\annot@end{}}%
9267     }%
9268     \noexpand\csuse{v\@msdata@series footnote}{\@msdata@series}{\{
expandonce\l@d@nums}{\ms@data@label}{\expandonce\data}}%
9269     \reset@msd@options@iffullpage%
9270     }%
9271     }%
9272     {\global\@msdata@insertedfrompreviouspagefalse}%
9273     }{}%
9274     \else%
9275     \ifcsdef{@msdata@\add@msd@cR @dataR}{%
9276     \ifnumless{\the\absline@numR-\@one}{\csuse{@msdata@\add@msd@cR
@stopabslinenumberR}}%
9277     {%
9278     \global\@msdata@insertedfrompreviouspagetrue%
9279     \letcs{@data}{@msdata@\add@msd@cR @dataR}%
9280     \edef\l@d@nums{%
9281     000| Start page = we don't print it
9282     \numexpr\the\line@numR+\@one\relax| Start line number = first
line of the page. As \add@msdata@firstlineofpage is called before line
number has been incremented, we increment it for printing
9283     000| Start subline number, for now, not used
9284     000| End page number, we don't print it
9285     \ifnumless{\csuse{@msdata@\add@msd@cR @stopabslinenumberR}}{\{
csuse{@lastline@forpageR@\the\page@numR}}}%
9286     {\csuse{@msdata@\add@msd@cR @stoplinenumberR}}%End line number
if in the same page
9287     {\csuse{@lastline@forpageR@\the\page@numR}}%Otherwise, last
number of the page
9288     |%
9289     000| End sub line number, for now, not used
9290     \edfont@info%Font
9291     }%
9292     \@msd@options@fullpagefalse%
9293     \ifnumless{\csuse{@lastabsline@forpageR@\the\page@numR}}{\csuse{
@msdata@\add@msd@cR @stopabslinenumberR}+\@one}%

```

```

9294     {%
9295     \numdef{\@tmp}{\add@msd@cR+\@ne}%
9296     \ifcsdef{@msdata@\@tmp @abslinenumberR}%
9297     {\ifnumequal{\csuse{@msdata@\@tmp @abslinenumberR}}{\csuse{
@lastabsline@forpageR@\the\page@numR}}}%
9298     }%
9299     {\@msd@options@fullpagetrue}%
9300     }%
9301     {\@msd@options@fullpagetrue}%
9302     }%
9303     }%
9304     \listxadd{\@msdata@list}{%
9305     \@msd@options@iffullpage%
9306     \ifluatex%
9307     \csxdef{footnote@luatextextdir}{\the\textdir}%
9308     \csxdef{footnote@luatexpardir}{\the\pardir}%
9309     \fi%
9310     \csdef{@this@crossref@start}{\add@msd@cR:start:msdata}%
9311     \csdef{@this@crossref@end}{\add@msd@cR:end:msdata}%
9312     \unexpanded{%
9313     \def\annot@start{%
9314     \def\annot@end{%
9315     }%
9316     \noexpand\csuse{v\@msdata@series footnote}{\@msdata@series}{\{
expandonce\l@d@nums}{\ms@data@label}{\expandonce\@data}}%
9317     \reset@msd@options@iffullpage%
9318     }%
9319     }%
9320     {\global\@msdata@insertedfrompreviouspagefalse}%
9321     }%
9322     \fi%
9323     \egroup%
9324     }%
9325     %

```

### 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@msdata26 \newcommand{\insert@msdata}{%
9327     \def\do##1{##1}%
9328     \dolistloop{\@msdata@list}%
9329     \global\let\@msdata@list\relax%
9330     }%
9331     %

```

### 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.

```

9332 \newif\if@msd@options@fullpage%
9333 \newcommand{\@msd@options@iffullpage}[0]{%
9334   \if@msd@options@fullpage%
9335     \noexpand\toggletrue{nonum@}%
9336     \ifdefvoid{\ms@data@label}%
9337       {\noexpand\toggletrue{nosep@}}%
9338     }%
9339   \fi%
9340 }%
9341 \newcommand{\reset@msd@options@iffullpage}[0]{%
9342   \noexpand\togglefalse{nonum@}%
9343   \noexpand\togglefalse{nosep@}%
9344 }%
9345 %

```

## XXXII Arrays and tables

### XXXII.1 Preamble: macro as environment

The following is borrowed, and renamed, from the `amsmath` package. See also the CTT thread “`eeq` and `amstex`”, 1995/08/31, started by Keith Reckdahl and ended definitively by David M. Jones.

Several of the `[math]` macros scan their body twice. This means we must collect all text in the body of an environment form before calling the macro.

`\@emptytoks` This is actually defined in the `amsgen` package.

```

9346 \newtoks\@emptytoks
9347
9348 %

```

The rest is from `amsmath`.

`\l@denbody` A token register to contain the body.

```

9349 \newtoks\l@denbody
9350
9351 %

```

`\addtol@denbody` `\addtol@denbody{arg}` adds `arg` to the token register `\l@denbody`.

```

9352 \newcommand{\addtol@denbody}[1]{%
9353   \global\l@denbody\expandafter{\the\l@denbody#1}}
9354 %

```

`\if@edtabular` A boolean set to true inside a `reledmac`’ tabular environment.

```

9355 \newif\if@edtabular%
9356 %

```

`\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`.

```

9357 \newcommand{\l@dcollect@body}[1]{%
9358   \ifvmode\else%
9359   \\\%Ensure that we do not have spurious indentation at the very first
line, caused by some text before.
9360   \fi%
9361   \l@denvironmentbody{\expandafter#1\expandafter{\the\l@denvironmentbody}}%
9362   \edef\processl@denvironmentbody{\the\l@denvironmentbody\noexpand\end{\@currentenv}}%
9363   \l@denvironmentbody\@emptytoks \def\l@dbegin@stack{b}%
9364   \begingroup
9365   \@etabulartrue%
9366   \expandafter\let\csname\@currentenv\endcsname\l@dcollect@body
9367   \edef\processl@denvironmentbody{\expandafter\noexpand\csname\@currentenv
endcsname}%
9368   \processl@denvironmentbody%
9369   }%
9370
9371 %

```

`\l@dpush@begins` When adding a piece of the current environment's contents to `\l@denvironmentbody`, we scan it to check for additional `\begin` tokens, and add a "b" to the stack for any that we find.

```

9372 \def\l@dpush@begins#1\begin#2{%
9373   \ifx\end#2\else b\expandafter\l@dpush@begins\fi}
9374
9375 %

```

`\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.

```

9376 \def\l@dcollect@@body#1\end#2{%
9377   \edef\l@dbegin@stack{\l@dpush@begins#1\begin\end
\expandafter\@gobble\l@dbegin@stack}%
9378   \ifx\@empty\l@dbegin@stack
9379     \endgroup
9380     \@checkend{#2}%
9381     \addtol@denvironmentbody{#1}%
9382   \else
9383     \addtol@denvironmentbody{#1\end{#2}}%
9384   \fi
9385   \processl@denvironmentbody % A little tricky! Note the grouping
9386 }
9387
9388 %
9389

```

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#{%
  \@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.

```

9390 \newcommand*\l@dtabnoexpands}{%
9391   \let\rtab=0%
9392   \let\ctab=0%
9393   \let\ltab=0%
9394   \let\rtabtext=0%
9395   \let\ltabtext=0%
9396   \let\ctabtext=0%
9397   \let\edbeforetab=0%
9398   \let\edaftertab=0%
9399   \let\edatleft=0%
9400   \let\edatright=0%
9401   \let\edvertline=0%
9402   \let\edvertdots=0%
9403   \let\edrowfill=0%
9404 }
9405
9406 %

```

`\disable@familiarnotes` Macros to disable and restore familiar notes, to prevent them from printing multiple times in edtabularx and edarrayx environments.

`\restore@familiarnotes`

```

9407 \newcommand{\disable@familiarnotes}{%
9408   \unless\ifnofamiliar%
9409     \def\do##1{%
9410       \csletcs{footnote@##1}{footnote##1}%
9411       \expandafter\renewcommand \csname footnote##1\endcsname[1]{%
9412         \protected@csxdef{thefnmark##1}{\csuse{thefootnote##1}}%
9413         \csuse{footnotemark##1}%
9414       }%
9415     }%
9416     \dolistloop{\@series}%
9417   \fi%
9418 }%
9419 \newcommand{\restore@familiarnotes}{%
9420   \unless\ifnofamiliar%
9421     \def\do##1{%
9422       \csletcs{footnote##1}{footnote@##1}%
9423     }%
9424     \dolistloop{\@series}%
9425   \fi%
9426 }%
9427
9428 %

```

`\disable@sidenotes` The same for sidenotes.

`\restore@sidenotes`

```

9429 \newcommand{\disable@sidenotes}{%
9430   \let\@ledrightnote\ledrightnote%
9431   \let\@ledleftnote\ledleftnote%
9432   \let\@ledsidenote\ledsidenote%
9433   \let\ledrightnote@gobble%
9434   \let\ledleftnote@gobble%
9435   \let\ledsidenote@gobble%
9436 }%

```

```

9437 \newcommand{\restore@sidenotes}{%
9438   \let\ledrightnote\@ledrightnote%
9439   \let\ledleftnote\@ledleftnote%
9440   \let\ledsidenote\@ledsidenote%
9441 }%
9442 %

```

`\disable@endnotes` The same for endnotes.

`\restore@endnotes`

```

9443 \newcommand{\disable@endnotes}{%
9444   \unless\ifnoend@%
9445     \def\do##1{%
9446       \csletcs{##1@@endnote}{##1endnote}%
9447       \expandafter\renewcommand \csname ##1endnote\endcsname[1]{}%
9448     }%
9449     \dolistloop{\@series}%
9450   \fi%
9451 }%
9452 \newcommand{\restore@endnotes}{%
9453   \unless\ifnofamiliar@%
9454     \def\do##1{%
9455       \csletcs{##1endnote}{##1@@endnote}%
9456     }%
9457     \dolistloop{\@series}%
9458   \fi%
9459 }%
9460 %

```

`\disable@notes` Disable/restore side, familiar and end notes.

`\restore@notes`

```

9461 \newcommand{\disable@notes}{%
9462   \disable@sidenotes%
9463   \disable@familiarnotes%
9464   \disable@endnotes%
9465 }%
9466 \newcommand{\restore@notes}{%
9467   \restore@sidenotes%
9468   \restore@familiarnotes%
9469   \restore@endnotes%
9470 }%
9471 %

```

`\EDTEXT` We need to be able to modify the `\edtext` macros and also restore their original definitions.

`\xedtext`

```

9472 \let\EDTEXT=\edtext
9473 \newcommand{\xedtext}[2]{\EDTEXT{#1}{#2}}
9474 %

```

`\disable@edlabel` We need to be able to modify and restore the `\edlabel` macro.

`\restore@edlabel`

```

9475 \newcommand{\disable@edlabel}{%
9476   \let\old@edlabel\edlabel%
9477   \let\edlabel\@gobble%

```



```

9478 }%
9479 \newcommand{\restore@edlabel}{%
9480   \let\edlabel\old@edlabel%
9481 }%
9482 %

```

**\xedindex** Macros supporting modification and restoration of `\edindex`.

```

\nulledindex
9483 \AtBeginDocument{\let\xedindex\edindex}%
9484 \newcommand{\nulledindex}[2][\jobname]{\@bsphack\@esphack}
9485
9486 %

```

**\@line@num** Macro supporting restoration of `\linenum`.

```

9487 \let\@line@num=\linenum
9488 %

```

**\l@gobbleoptarg** `\l@gobbleoptarg[⟨arg⟩]{⟨arg⟩}` replaces these two arguments (first is optional) by `\relax`.

```

9489 \newcommand*\l@gobbleoptarg}[2][\relax]%
9490
9491 %

```

**\l@secondmandarg** `\l@secondoptarg[⟨arg⟩]{⟨arg⟩}` gobble the first (optional) argument, and expand to the second (mandatory) argument.

```

9492 \NewExpandableDocumentCommand{\l@secondmandarg}{om}{#2}%
9493 %

```

```

\Relax94 \let\Relax=\relax

```

```

\NEXT95 \let\NEXT=\next

```

```

9496
9497 %

```

**\l@dmodifyedtext** Modify and restore various macros for when `\edtext` is used.

```

\l@drestoreedtext
9498 \newcommand{\l@dmodifyedtext}{%
9499   \let\edtext\relax
9500   \def\do##1{\global\csletcs{##1footnote}{\l@gobbleoptarg}}%
9501   \dolistloop{\@series}%
9502   \let\edindex\nulledinde%
9503   \disable@edlabel%
9504   \let\linenum\@gobble}
9505 \newcommand{\l@drestoreedtext}{%
9506   \def\do##1{\global\csletcs{##1footnote}{##1@footnote}}
9507   \dolistloop{\@series}%
9508   \let\edindex\xedindex}
9509 %

```

**\l@dnullfills** Nullify and restore some column fillers, etc.

**\l@drestorefills**

```

9510 \newcommand{\l@dnnullfills}{%
9511   \disable@edlabel%
9512   \def\edrowfill##1##2##3{%
9513 }
9514 \newcommand{\l@drestorefills}{%
9515   \def\edrowfill##1##2##3{\@EDROWFILL@{##1}{##2}{##3}}%
9516 }
9517 %
9518 %

```

**\letsforverteilen** Gathers some lets and other code that is common to the *\*verteilen\** macros.

```

9519 \newcommand{\letsforverteilen}{%
9520   \let\edtext\xedtext
9521   \let\edindex\xedindex
9522   \def\do##1{\global\csletcs{##1footnote}{##1@footnote}}
9523   \dolistloop{\@series}%
9524   \let\linenum\@line@num
9525   \hilfsskip=\l@dcwidth%
9526   \advance\hilfsskip by -\wd\hilfsbox
9527   \restore@edlabel%
9528 }%
9529 %

```

**\disablel@dtabfeet** Declarations for using or using `\edtext` inside tabulars. The default at this point is for `\edtext`.

**\enablel@dtabfeet**

```

9530 \newcommand\disablel@dtabfeet{\l@dmodforedtext}%
9531 \newcommand\enablel@dtabfeet{\l@drestoreforedtext}%
9532 %

```

### XXXII.2.2 Counters, boxes and lengths

**\l@dampcount** `\l@dampcount` is a counter for the & column dividers and `\l@dcolcount` is a counter for the columns.

**\l@dcolcount**

```

9533 \newcount\l@dampcount
9534 \l@dampcount=1\relax
9535 \newcount\l@dcolcount
9536 \l@dcolcount=0\relax
9537 %
9538 %

```

**\hilfsbox** Some (temporary) helper items.

```

9539 \newbox\hilfsbox
9540 \newskip\hilfsskip
9541 \newbox\Hilfsbox
9542 \newcount\hilfscount
9543 %
9544 %

```

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).

```

9545 \newdimen\dcoli
9546 \newdimen\dcolii
9547 \newdimen\dcoliii
9548 \newdimen\dcoliv
9549 \newdimen\dcolv
9550 \newdimen\dcolvi
9551 \newdimen\dcolvii
9552 \newdimen\dcolviii
9553 \newdimen\dcolix
9554 \newdimen\dcolx
9555 \newdimen\dcolxi
9556 \newdimen\dcolxii
9557 \newdimen\dcolxiii
9558 \newdimen\dcolxiv
9559 \newdimen\dcolxv
9560 \newdimen\dcolxvi
9561 \newdimen\dcolxvii
9562 \newdimen\dcolxviii
9563 \newdimen\dcolxix
9564 \newdimen\dcolxx
9565 \newdimen\dcolxxi
9566 \newdimen\dcolxxii
9567 \newdimen\dcolxxiii
9568 \newdimen\dcolxxiv
9569 \newdimen\dcolxxv
9570 \newdimen\dcolxxvi
9571 \newdimen\dcolxxvii
9572 \newdimen\dcolxxviii
9573 \newdimen\dcolxxix
9574 \newdimen\dcolxxx
9575 \newdimen\dcolerr % added for error handling
9576
9577 %

```

`\l@dcolwidth` This is a cunning way of storing the columnwidths indexed by the column number `\l@dcolcount`, like an array. (was `\Dimenzuordnung`)

```

9578 \newcommand{\l@dcolwidth}{\ifcase \the\l@dcolcount \dcoli %???
9579 \or \dcoli \or \dcolii \or \dcoliii
9580 \or \dcoliv \or \dcolv \or \dcolvi
9581 \or \dcolvii \or \dcolviii \or \dcolix \or \dcolx
9582 \or \dcolxi \or \dcolxii \or \dcolxiii
9583 \or \dcolxiv \or \dcolxv \or \dcolxvi
9584 \or \dcolxvii \or \dcolxviii \or \dcolxix \or \dcolxx
9585 \or \dcolxxi \or \dcolxxii \or \dcolxxiii
9586 \or \dcolxxiv \or \dcolxxv \or \dcolxxvi
9587 \or \dcolxxvii \or \dcolxxviii \or \dcolxxix \or \dcolxxx
9588 \else \dcolerr \fi}
9589
9590 %

```

`\stepl@dcolcount` This increments the column counter, and issues an error message if it is too large.

```

9591 \newcommand*{\stepl@dcolcount}{\advance\l@dcolcount\@ne

```

```

9592 \ifnum\l@dcolcount>30\relax
9593 \led@err@TooManyColumns
9594 \fi}
9595
9596 %

```

**\l@dsetmaxcolwidth** Sets the column width to the maximum value seen so far.

```

9597 \newcommand{\l@dsetmaxcolwidth}{%
9598 \ifdim\l@dcolwidth < \wd\hilfsbox
9599 \l@dcolwidth = \wd\hilfsbox
9600 \else \relax \fi}
9601
9602 %

```

**\measuremcell** Measure (recursively) the width required for a math cell.

```

9603 \def\measuremcell #1&{%
9604 \ifx #1\ \ifnum\l@dcolcount=0\let\NEXT\relax%
9605 \else\l@dcheckcols%
9606 \l@dcolcount=0%
9607 \let\NEXT\measuremcell%
9608 \fi%
9609 \else\setbox\hilfsbox=\hbox{${\displaystyle{#1}}}%
9610 \step1@dcolcount%
9611 \l@dsetmaxcolwidth%
9612 \let\NEXT\measuremcell%
9613 \fi\NEXT}
9614
9615 %

```

**\measuretcell** Measure (recursively) the width required for a text cell.

```

9616 \def\measuretcell #1&{%
9617 \ifx #1\ \ifnum\l@dcolcount=0\let\NEXT\relax%
9618 \else\l@dcheckcols%
9619 \l@dcolcount=0%
9620 \let\NEXT\measuretcell%
9621 \fi%
9622 \else\setbox\hilfsbox=\hbox{#1}%
9623 \step1@dcolcount%
9624 \l@dsetmaxcolwidth%
9625 \let\NEXT\measuretcell%
9626 \fi\NEXT}
9627
9628 %

```

**\measuremrow** Measure (recursively) the width required for a math row.

```

9629 \def\measuremrow #1\\{%
9630 \ifx #1&\let\NEXT\relax%
9631 \else\measuremcell #1&\\&\\&%
9632 \let\NEXT\measuremrow%
9633 \fi\NEXT}
9634 %

```

`\measurerow` Measure (recursively) the width required for a text row.

```

9635 \def\measurerow #1\{\%
9636   \ifx #1&\let\NEXT\relax%
9637   \else\measurecell #1&\&\&\&%
9638     \let\NEXT\measurerow%
9639   \fi\NEXT}
9640
9641 %

```

`\edtabcolsep` The length `\edtabcolsep` controls the distance between columns.

```

9642 \newskip\edtabcolsep
9643 \global\edtabcolsep=10pt
9644
9645 %

```

```

\variab16 \newcommand{\variab}{\relax}
9647
9648 %

```

`\l@dcheckcols` Check that the number of columns is consistent.

```

9649 \newcommand*\l@dcheckcols{\%
9650   \ifnum\l@dcolcount=1\relax
9651   \else
9652     \ifnum\l@dampcount=1\relax
9653     \else
9654       \ifnum\l@dcolcount=\l@dampcount\relax
9655       \else
9656         \l@d@err@UnequalColumns
9657       \fi
9658     \fi
9659     \l@dampcount=\l@dcolcount
9660   \fi}
9661
9662 %

```

`\edfilldimen` A length.

```

9663 \newdimen\edfilldimen
9664 \edfilldimen=0pt
9665
9666 %

```

`\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` L<sup>A</sup>T<sub>E</sub>X command, because some packages, like `babel` can override it in some specific cases (Greek, for example).

```

9667 \newcounter{addcolcount}
9668 \renewcommand{\theadcolcount}{\romannumeral \c@addcolcount}
9669 %

```

## XXXII.2.3 Tabular typesetting

`\setmcellright` Typeset (recursively) cells of display math right justified.

```

9670 \def\setmcellright #1&{%
9671     \disable@edlabel%
9672     \let\edindex\nulledindex
9673     \ifx #1\ \ifnum\l@dcolcount=0%\removelastskip
9674         \let\Next\relax%
9675     \else\l@dcolcount=0%
9676         \let\Next=\setmcellright%
9677     \fi%
9678 \else%
9679     \disablel@dtabfeet%
9680     \stepl@dcolcount%
9681     \disable@notes%
9682     \setbox\hilfsbox=\hbox{${\displaystyle{#1}}$}%
9683     \restore@notes%
9684     \letsforverteilen%
9685     \hskip\hilfsskip${\displaystyle{#1}}$%
9686     \hskip\edtabcolsep%
9687     \let\Next=\setmcellright%
9688 \fi\Next}
9689 %
9690 %

```

`\settcclright` Typeset (recursively) cells of text right justified.

```

9691 \def\settcclright #1&{%
9692     \disable@edlabel%
9693     \let\edindex\nulledindex
9694     \ifx #1\ \ifnum\l@dcolcount=0%\removelastskip
9695         \let\Next\relax%
9696     \else\l@dcolcount=0%
9697         \let\Next=\settcclright%
9698     \fi%
9699 \else%
9700     \disablel@dtabfeet%
9701     \stepl@dcolcount%
9702     \disable@notes%
9703     \setbox\hilfsbox=\hbox{#1}%
9704     \restore@notes%
9705     \letsforverteilen%
9706     \hskip\hilfsskip#1%
9707     \hskip\edtabcolsep%
9708     \let\Next=\settcclright%
9709 \fi\Next}
9710 %

```

`\setmcellleft` Typeset (recursively) cells of display math left justified.

```

9711 \def\setmcellleft #1&{%
9712     \disable@edlabel%
9713     \let\edindex\nulledindex
9714     \ifx #1\ \ifnum\l@dcolcount=0 \let\Next\relax%

```

```

9715         \else\l@dcolcount=0%
9716             \let\Next=\setmcellleft%
9717         \fi%
9718     \else    \disablel@dtabfeet%
9719             \stepl@dcolcount%
9720             \disable@notes%
9721             \setbox\hilfsbox=\hbox{\displaystyle{#1}}%
9722             \restore@notes%
9723             \letsforverteilen%
9724             $\displaystyle{#1}$\hskip\hilfsskip\hskip\edtabcolsep%
9725             \let\Next=\setmcellleft%
9726     \fi\Next}
9727
9728 %

```

`\settcelleleft` Typeset (recursively) cells of text left justified.

```

9729 \def\settcelleleft #1&{%
9730     \disable@edlabel%
9731     \let\edindex\nulledindex
9732     \ifx #1\\ \ifnum\l@dcolcount=0 \let\Next\relax%
9733         \else\l@dcolcount=0%
9734             \let\Next=\settcelleleft%
9735         \fi%
9736     \else    \disablel@dtabfeet%
9737             \stepl@dcolcount%
9738             \disable@notes%
9739             \setbox\hilfsbox=\hbox{#1}%
9740             \restore@notes%
9741             \letsforverteilen%
9742             #1\hskip\hilfsskip\hskip\edtabcolsep%
9743             \let\Next=\settcelleleft%
9744     \fi\Next}
9745 %

```

`\setmcellcenter` Typeset (recursively) cells of display math centered.

```

9746 \def\setmcellcenter #1&{%
9747     \disable@edlabel%
9748     \let\edindex\nulledindex
9749     \ifx #1\\ \ifnum\l@dcolcount=0\let\Next\relax%
9750         \else\l@dcolcount=0%
9751             \let\Next=\setmcellcenter%
9752         \fi%
9753     \else    \disablel@dtabfeet%
9754             \stepl@dcolcount%
9755             \disable@notes%
9756             \setbox\hilfsbox=\hbox{\displaystyle{#1}}%
9757             \restore@notes%
9758             \letsforverteilen%
9759             \hskip 0.5\hilfsskip$\displaystyle{#1}$\hskip0.5\hilfsskip%
9760             \hskip\edtabcolsep%
9761             \let\Next=\setmcellcenter%
9762     \fi\Next}

```

```
9763 %
9764 %
```

**\settcclcenter** Typeset (recursively) cells of text centered.

```
9765 \def\settcclcenter #1&{%
9766     \disable@edlabel%
9767     \let\edindex\nulledindex
9768     \ifx #1\ \ifnum\l@dcclcount=0 \let\Next\relax%
9769         \else\l@dcclcount=0%
9770         \let\Next=\settcclcenter%
9771         \fi%
9772     \else \disablel@dtabfeet%
9773         \step1@dcclcount%
9774         \disable@notes%
9775         \setbox\hilfsbox=\hbox{#1}%
9776         \restore@notes%
9777         \letsforverteilen%
9778         \hskip 0.5\hilfsskip #1\hskip 0.5\hilfsskip%
9779         \hskip\edtabcolsep%
9780         \let\Next=\settcclcenter%
9781     \fi\Next}
9782 %
9783 %
```

```
\NEXT 9784 \let\NEXT=\relax
```

```
9785 %
9786 %
```

**\setmrowright** Typeset (recursively) rows of right justified math.

```
9787 \def\setmrowright #1\\{%
9788     \ifx #1& \let\NEXT\relax
9789     \else \centerline{\setmcellright #1&\\&\\&}
9790         \let\NEXT=\setmrowright
9791     \fi\NEXT}
9792 %
```

**\settroright** Typeset (recursively) rows of right justified text.

```
9793 \def\settroright #1\\{%
9794     \ifx #1& \let\NEXT\relax
9795     \else \centerline{\settcclright #1&\\&\\&}
9796         \let\NEXT=\settroright
9797     \fi\NEXT}
9798 %
9799 %
```

**\setmrowleft** Typeset (recursively) rows of left justified math.

```
9800 \def\setmrowleft #1\\{%
9801     \ifx #1&\let\NEXT\relax
9802     \else \centerline{\setmcellleft #1&\\&\\&}
9803 %
```



```

9803     \let\NEXT=\setmrowleft
9804     \fi\NEXT}
9805 %

```

**\settrorleft** Typeset (recursively) rows of left justified text.

```

9806 \def\settrorleft #1\{ \%
9807     \ifx #1& \let\NEXT\relax
9808     \else \centerline{\settcclleft #1&\&\&}
9809         \let\NEXT=\settrorleft
9810     \fi\NEXT}
9811
9812 %

```

**\setmrowcenter** Typeset (recursively) rows of centered math.

```

9813 \def\setmrowcenter #1\{ \%
9814     \ifx #1& \let\NEXT\relax%
9815     \else \centerline{\setmcellcenter #1&\&\&}
9816         \let\NEXT=\setmrowcenter
9817     \fi\NEXT}
9818 %

```

**\settrorcenter** Typeset (recursively) rows of centered text.

```

9819 \def\settrorcenter #1\{ \%
9820     \ifx #1& \let\NEXT\relax
9821     \else \centerline{\settcclcenter #1&\&\&}
9822         \let\NEXT=\settrorcenter
9823     \fi\NEXT}
9824
9825 %

```

**\nullsetzen** `\newcommand{\nullsetzen}{%`

```

9827     \step1@dcolcount%
9828     \l@dcolwidth=Opt%
9829     \ifnum\l@dcolcount=30\let\NEXT\relax%
9830         \l@dcolcount=0\relax
9831     \else\let\NEXT\nullsetzen%
9832     \fi\NEXT}
9833
9834 %

```

**\edatleft** `\edatleft[ $len$ ]{ $symbol$ }{ $len$ }`. Left  $symbol$ ,  $2len$  high with prepended  $math$  vertically centered.

```

9835 \newcommand{\edatleft}[3][\@empty]{ \%
9836     \ifx#1\@empty
9837         \vbox to 10pt{\vss\hbox{\left#2\vrule widthOpt height #3
9838             depth Opt \right. $\hss}\vfil}
9839     \else
9840         \vbox to 4pt{\vss\hbox{\left#2\vrule widthOpt height #3
9841             depth Opt \right. $\}\vfil}
9842     \fi}
9843 %

```

`\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.

```

9844 \newcommand{\edatright}[3][\@empty]{%
9845   \ifx#1\@empty
9846     \vbox to 10pt{\vss\hbox{\left.\vrule width0pt height #3
9847       depth 0pt \right#2 $\hss}\vfil}
9848   \else
9849     \vbox to 4pt{\vss\hbox{\left.\vrule width0pt height #3
9850       depth 0pt \right#2 #1 $\}\vfil}
9851   \fi}
9852
9853 %

```

`\edvertline` `\edvertline`{ $\langle len \rangle$ } vertical line  $\langle len \rangle$  high.

```

9854 \newcommand{\edvertline}[1]{\vbox to 8pt{\vss\hbox{\vrule height #1}\vfil}}
9855
9856 %

```

`\edvertdots` `\edvertdots`{ $\langle len \rangle$ } vertical dotted line  $\langle len \rangle$  high.

```

9857 \newcommand{\edvertdots}[1]{\vbox to 1pt{\vss\vbox to #1%
9858   {\cleaders\hbox{$\m@th\hbox{.}\vbox to 0.5em{ }\vfil}}}
9859
9860 %

```

`\l@dtabaddcols` `\l@dtabaddcols`{ $\langle startcol \rangle$ }{ $\langle endcol \rangle$ } adds the widths of the columns  $\langle startcol \rangle$  through  $\langle endcol \rangle$  to `\edfilldimen`. It is a  $\LaTeX$  style reimplementation of the original `\@add@`.

```

9861 \newcommand{\l@dtabaddcols}[2]{%
9862   \l@dcheckstartend{#1}{#2}%
9863   \ifl@dstartendok
9864     \setcounter{addcolcount}{#1}%
9865     \@whilenum \value{addcolcount}<#2\relax \do
9866       {\advance\edfilldimen by \the \csname dcol\theadcolcount\endcsname
9867         \advance\edfilldimen by \edtabcolsep
9868         \stepcounter{addcolcount}}%
9869     \advance\edfilldimen by \the \csname dcol\theadcolcount\endcsname
9870   \fi
9871 }
9872
9873 %

```

`\ifl@dstartendok` `\l@dcheckstartend`{ $\langle startcol \rangle$ }{ $\langle endcol \rangle$ } checks that the values of  $\langle startcol \rangle$  and  $\langle endcol \rangle$  are sensible. If they are then `\ifl@dstartendok` is set TRUE, otherwise it is set FALSE.

```

9874 \newif\ifl@dstartendok
9875 \newcommand{\l@dcheckstartend}[2]{%
9876   \l@dstartendoktrue
9877   \ifnum #1<\@ne
9878     \l@dstartendokfalse
9879     \led@err@LowStartColumn

```

```

9880 \fi
9881 \ifnum #2>30\relax
9882   \l@dstartendokfalse
9883   \led@err@HighEndColumn
9884 \fi
9885 \ifnum #1>#2\relax
9886   \l@dstartendokfalse
9887   \led@err@ReverseColumns
9888 \fi
9889 }
9890
9891 %

```

`\edrowfill` `\edrowfill{<startcol>}{<endcol>}` fill fills columns `<startcol>` to `<endcol>` inclusive with `<fill>` (e.g. `\hrulefill`, `\upbracefill`). This is a  $\TeX$  style reimplementaion and generalization of the original `\waklam`, `\Waklam`, `\waklamec`, `\wastricht` and `\wapunktetl` macros.

```

9892 \newcommand*\edrowfill}[3]{%
9893   \l@dtabaddcols{#1}{#2}%
9894   \hb@xt@ \the\l@dcolwidth{\hb@xt@ \the\edfilldimen{#3}\hss}}
9895 \let\@edrowfill@=\edrowfill
9896 \def\@EDROWFILL@#1#2#3{\@edrowfill@{#1}{#2}{#3}}
9897
9898 %

```

`\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`.

```

9899 \newcommand{\leftltab}[1]{%
9900   \hb@xt@\z@{\vbox{\edtabindent%
9901     \moveleft\Hilfsskip\hbox{\ #1}}\hss}}
9902
9903 %

```

`\leftrtab` `\leftrtab{<text>}{<math>}` for `\edbeforetab` in `\rtab`.

```

9904 \newcommand{\leftrtab}[2]{%
9905   #2\hb@xt@\z@{\vbox{\edtabindent%
9906     \advance\Hilfsskip by\dcoli%
9907     \moveleft\Hilfsskip\hbox{\ #1}}\hss}}
9908
9909 %

```

`\leftctab` `\leftctab{<text>}{<math>}` for `\edbeforetab` in `\ctab`.

```

9910 \newcommand{\leftctab}[2]{%
9911   \hb@xt@\z@{\vbox{\edtabindent\l@dcolcount=\l@dampcount%
9912     \advance\Hilfsskip by 0.5\dcoli%
9913     \setbox\hilfsbox=\hbox{#2}}

```

```

9914 \disable@edlabel%
9915 \disablel@dtabfeet$\displaystyle{#2}$}%
9916 \advance\Hilfsskip by -0.5\wd\hilfsbox%
9917 \moveleft\Hilfsskip\hbox{\ #1}\hss}%
9918 #2}
9919
9920 %

```

`\rightctab` `\rightctab{<math></math>}{<text>}` for `\edaftertab` in `\ctab`.

```

9921 \newcommand{\rightctab}[2]{%
9922 \setbox\hilfsbox=\hbox{%
9923 \disable@edlabel%
9924 \disablel@dtabfeet#2}\l@dampcount=\l@dcolcount%
9925 #1\hb@xt@z@{\vbox{\edtabindent\l@dcolcount=\l@dampcount%
9926 \advance\Hilfsskip by 0.5\l@dcolwidth%
9927 \advance\Hilfsskip by -\wd\hilfsbox%
9928 \setbox\hilfsbox=\hbox{%
9929 \disable@edlabel%
9930 \disablel@dtabfeet$\displaystyle{#1}$}%
9931 \advance\Hilfsskip by -0.5\wd\hilfsbox%
9932 \advance\Hilfsskip by \edtabcolsep%
9933 \moveright\Hilfsskip\hbox{ #2}\hss}%
9934 }
9935
9936 %

```

`\rightltab` `\rightltab{<math></math>}{<text>}` for `\edaftertab` in `\ltab`.

```

9937 \newcommand{\rightltab}[2]{%
9938 \setbox\hilfsbox=\hbox{%
9939 \disable@edlabel%
9940 \disablel@dtabfeet#2}\l@dampcount=\l@dcolcount%
9941 #1\hb@xt@z@{\vbox{\edtabindent\l@dcolcount=\l@dampcount%
9942 \advance\Hilfsskip by\l@dcolwidth%
9943 \advance\Hilfsskip by-\wd\hilfsbox%
9944 \setbox\hilfsbox=\hbox{%
9945 \disable@edlabel%
9946 \disablel@dtabfeet$\displaystyle{#1}$}%
9947 \advance\Hilfsskip by-\wd\hilfsbox%
9948 \advance\Hilfsskip by\edtabcolsep%
9949 \moveright\Hilfsskip\hbox{ #2}\hss}%
9950 }
9951
9952 %

```

`\rightrtab` `\rightrtab{<math></math>}{<text>}` for `\edaftertab` in `\rtab`.

```

9953 \newcommand{\rightrtab}[2]{%
9954 \setbox\hilfsbox=\hbox{%
9955 \disable@edlabel%
9956 \disablel@dtabfeet#2}%
9957 #1\hb@xt@z@{\vbox{\edtabindent%
9958 \advance\Hilfsskip by-\wd\hilfsbox%

```

```

9959     \advance\Hilfsskip by\edtabcolsep%
9960     \moveright\Hilfsskip\hbox{ #2}\hss}%
9961     }
9962
9963 %

```

`\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.

```

9964 \newcommand{\rtab}[1]{%
9965   \l@dnnullfills
9966   \def\edbeforetab##1##2{\lefttrtab{##1}{##2}}%
9967   \def\edaftertab##1##2{\righttrtab{##1}{##2}}%
9968   \measuretbody{#1}%
9969   \l@drestorefills
9970   \variab
9971   \setmrowright #1\&\%
9972   \enablel@dtabfeet}
9973
9974 %

```

`\measuretbody` `\measuretbody{<body>}` measures the array `<body>`.

```

9975 \newcommand{\measuretbody}[1]{%
9976   \disablel@dtabfeet%
9977   \l@dcolcount=0%
9978   \nullsetzen%
9979   \l@dcolcount=0
9980   \measuremrow #1\&\%
9981   \global\l@dampcount=1}
9982
9983 %

```

`\rtabtext` `\rtabtext{<body>}` typesets `<body>` as a tabular with the entries right justified.

```

9984 \newcommand{\rtabtext}[1]{%
9985   \l@dnnullfills
9986   \measuretbody{#1}%
9987   \l@drestorefills
9988   \variab
9989   \settroright #1\&\%
9990   \enablel@dtabfeet}
9991
9992 %

```

`\measuretbody` `\measuretbody{<body>}` measures the tabular `<body>`.

```

9993 \newcommand{\measuretbody}[1]{%
9994   \disable@notes%
9995   \disablel@dtabfeet%
9996   \l@dcolcount=0%
9997   \nullsetzen%
9998   \l@dcolcount=0

```

```

9999 \measuretrrow #1\\&\\%
10000 \restore@notes%
10001 \global\l@dampcount=1}
10002
10003 %

```

**\ltab** Array with entries left justified.

```

\edbeforetab
\edaftertab
10004 \newcommand{\ltab}[1]{%
10005 \l@dnnullfills
10006 \def\edbeforetab##1##2{\leftltab{##1}{##2}}%
10007 \def\edaftertab##1##2{\rightltab{##1}{##2}}%
10008 \measuretbody{#1}%
10009 \l@drestorefills
10010 \variab
10011 \setmrowleft #1\\&\\%
10012 \enablel@dtabfeet}
10013
10014 %

```

**\ltabtext** Tabular with entries left justified.

```

10015 \newcommand{\ltabtext}[1]{%
10016 \l@dnnullfills
10017 \measuretbody{#1}%
10018 \l@drestorefills
10019 \variab
10020 \settrrowleft #1\\&\\%
10021 \enablel@dtabfeet}
10022
10023 %

```

**\ctab** Array with centered entries.

```

\edbeforetab
\edaftertab
10024 \newcommand{\ctab}[1]{%
10025 \l@dnnullfills
10026 \def\edbeforetab##1##2{\leftctab{##1}{##2}}%
10027 \def\edaftertab##1##2{\rightctab{##1}{##2}}%
10028 \measuretbody{#1}%
10029 \l@drestorefills
10030 \variab
10031 \setmrowcenter #1\\&\\%
10032 \enablel@dtabfeet}
10033
10034 %

```

**\ctabtext** Tabular with entries centered.

```

10035 \newcommand{\ctabtext}[1]{%
10036 \l@dnnullfills
10037 \measuretbody{#1}%
10038 \l@drestorefills
10039 \variab
10040 \settrrowcenter #1\\&\\%

```

```

10041 \enablel@dtabfeet}
10042
10043 %

```

```

\spreadtext44 \newcommand{\spreadtext}[1]{%\l@dcolcount=\l@dampcount%
10045 \hb@xt@ \the\l@dcolwidth{\hbox{#1}\hss}}
10046 %

```

```

\spreadmath47 \newcommand{\spreadmath}[1]{%
10048 \hb@xt@ \the\l@dcolwidth{\hbox{$\displaystyle{#1}$}\hss}}
10049
10050 %

```

**\HILFSskip** More helpers.

```

\Hilfsskip
10051 \newskip\HILFSskip
10052 \newskip\Hilfsskip
10053
10054 %

```

```

\EDTABINDENT55 \newcommand{\EDTABINDENT}{%
10056 \ifnum\l@dcolcount=30\let\NEXT\relax\l@dcolcount=0%
10057 \else\step\l@dcolcount%
10058 \advance\Hilfsskip by\l@dcolwidth%
10059 \ifdim\l@dcolwidth=0pt\advance\hilfscout\@ne
10060 \else\advance\Hilfsskip by \the\hilfscout\edtabcolsep%
10061 \hilfscout=1\fi%
10062 \let\NEXT=\EDTABINDENT%
10063 \fi\NEXT}%
10064 %

```

**\edtabindent** (was `\tabindent`)

```

10065 \newcommand{\edtabindent}{%
10066 \l@dcolcount=0\relax
10067 \Hilfsskip=0pt%
10068 \hilfscout=1\relax
10069 \EDTABINDENT%
10070 \hilfsskip=\hsize%
10071 \advance\hilfsskip -\Hilfsskip%
10072 \Hilfsskip=0.5\hilfsskip%
10073 }%
10074
10075 %

```

**\EDTAB** (was `\TAB`)

```

10076 \def\EDTAB #1|#2|{%
10077 \setbox\tabhilfbox=\hbox{$\displaystyle{#1}$}%
10078 \setbox\tabHilfbox=\hbox{$\displaystyle{#2}$}%
10079 \advance\tabelskip -\wd\tabhilfbox%

```

```

10080 \advance\tabelskip -\wd\tabHilfbox%
10081 \unhbox\tabhilfbox\hskip\tabelskip%
10082 \unhbox\tabHilfbox}%
10083
10084 %

```

`\EDTABtext` (was `\TABtext`)

```

10085 \def\EDTABtext #1|#2|{%
10086 \setbox\tabhilfbox=\hbox{#1}%
10087 \setbox\tabHilfbox=\hbox{#2}%
10088 \advance\tabelskip -\wd\tabhilfbox%
10089 \advance\tabelskip -\wd\tabHilfbox%
10090 \unhbox\tabhilfbox\hskip\tabelskip%
10091 \unhbox\tabHilfbox}%
10092 %

```

`\tabhilfbox` Further helpers.

```

\abHilfbox
10093 \newbox\tabhilfbox
10094 \newbox\tabHilfbox
10095
10096 %

```

#### XXXII.2.4 Environments

`edarrayl edarrayc edarrayr` The environment forms for `\ltab`, `\ctab` and `\rtab`.

```

10097 \newenvironment{edarrayl}{\l@dcollect@body\ltab}{\}
10098 \newenvironment{edarrayc}{\l@dcollect@body\ctab}{\}
10099 \newenvironment{edarrayr}{\l@dcollect@body\rtab}{\}
10100
10101 %

```

`edtabularl edtabularc edtabularr` The environment forms for `\ltabtext`, `\ctabtext` and `\rtabtext`.

```

10102 \newenvironment{edtabularl}{\l@dcollect@body\ltabtext}{\}
10103 \newenvironment{edtabularc}{\l@dcollect@body\ctabtext}{\}
10104 \newenvironment{edtabularr}{\l@dcollect@body\rtabtext}{\}
10105
10106 %

```

### XXXIII Quotation's commands

`\initnumbering@quote` This macro, called at the beginning of any numbered section, locally redefines the quotation and quote environments, in order to allow their use inside of numbered sections.

```

\quotation \initnumbering@quote defines quotation environment.
\endquotation
\quote10107 \newcommand{\initnumbering@quote}{
\endquote10108 \ifnoquotation@else
\endquote10109 \renewcommand{\quotation}{\par\leavevmode%

```



```

10110         \parindent=1.5em%
10111         \skipnumbering%
10112         \ifautopar%
10113             \vskip-\parskip%
10114         \else%
10115             \vskip\topsep%
10116         \fi%
10117         \global\leftskip=\leftmargin%
10118         \global\rightskip=\leftmargin%
10119     }
10120 \renewcommand{\endquotation}{\par%
10121         \global\leftskip=0pt%
10122         \global\rightskip=0pt%
10123         \leavevmode%
10124         \skipnumbering%
10125         \ifautopar%
10126             \vskip-\parskip%
10127         \else%
10128             \vskip\topsep%
10129         \fi%
10130     }
10131 \renewcommand{\quote}{\par\leavevmode%
10132         \parindent=0pt%
10133         \skipnumbering%
10134         \ifautopar%
10135             \vskip-\parskip%
10136         \else%
10137             \vskip\topsep%
10138         \fi%
10139         \global\leftskip=\leftmargin%
10140         \global\rightskip=\leftmargin%
10141     }
10142 \renewcommand{\endquote}{\par%
10143         \global\leftskip=0pt%
10144         \global\rightskip=0pt%
10145         \leavevmode%
10146         \skipnumbering%
10147         \ifautopar%
10148             \vskip-\parskip%
10149         \else%
10150             \vskip\topsep%
10151         \fi%
10152     }
10153 \fi
10154 }
10155 %

```

## XXXIV Section's title commands

### XXXIV.1 Commands to disable some feature

`\ledsectnotoc` The `\ledsectnotoc` only disables the `\addcontentsline` macro.

```
10156 \newcommand{\ledsectnotoc}{\let\addcontentsline@gobblethree}
10157 %
```

`\ledsectnomark` The `\ledsectnomark` only disables the `\chaptermark`, `\sectionmark` and `\subsectionmark` macros.

```
10158 \newcommand{\ledsectnomark}{%
10159   \let\chaptermark@gobble%
10160   \let\sectionmark@gobble%
10161   \let\subsectionmark@gobble%
10162 }
10163 %
```

## 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).

```
10164 \catcode`\#=12
10165 \notbool{@noeled@sec}{%
10166 %
```

`\beforeeledchapter` For technical reasons, not yet solved, page-breaking before chapters can't be made automatically by `eledmac`. Users have to use `\beforeeledchapter`.

```

10167 \ifl@dmemoir
10168   \newcommand\beforeelectedchapter{%
10169     \clearforchapter%
10170   }
10171 \else
10172   \newcommand\beforeelectedchapter{%
10173     \if@openright%
10174       \cleardoublepage%
10175     \else%
10176       \clearpage%
10177     \fi%
10178   }
10179 \fi
10180 %

```

### XXXIV.4 Auxiliary commands

`\print@leftmargin@eledsection` and `\print@rightmargin@eledsection` are added by `reledmac` inside the code of sectioning command, in order to affix line numbers. They include tests for RTL languages.

```

10181 \def\print@rightmargin@eledsection{%
10182   \if@eled@sectioning%
10183     \begingroup%
10184     \if@RTL%
10185       \let\llap\rlap%
10186       \let\leftlinenum\rightlinenum%
10187       \let\leftlinenumR\rightlinenumR%
10188       \let\l@drd@ta\l@dld@ta%
10189       \let\l@drsn@te\l@dlsn@te%
10190     \fi%
10191     \hfill\l@drd@ta \csuse{LR}{\l@drsn@te}%
10192   \endgroup%
10193 \fi%
10194 }%
10195
10196 \def\print@leftmargin@eledsection{%
10197   \if@eled@sectioning%
10198     \leavevmode%
10199     \begingroup%
10200     \if@RTL%
10201       \let\rlap\llap%
10202       \let\rightlinenum\leftlinenum%
10203       \let\rightlinenumR\leftlinenumR%
10204       \let\l@dld@ta\l@drd@ta%
10205       \let\l@dlsn@te\l@drsn@te%
10206     \fi%
10207     \l@dld@ta\csuse{LR}{\l@dlsn@te}%
10208   \endgroup%
10209 \fi%
10210 }%
10211
10212 %

```

### XXXIV.5 Patching standard commands

`\M@sect`  
`\@mem@old@ssect`  
`\@makechapterhead`  
`\@makechapterhead`  
`\@makeschapterhead`  
`\@sect`  
`\@ssect`

We have to patch L<sup>A</sup>T<sub>E</sub>X, book and memoir sectioning commands in order to:

- Disable `\edtext` inside.
- Disable page breaking (for `\chapter`).
- Add line numbers and sidenotes.

Unfortunately, Maïeul Rouquette was not able to try if memoir is loaded. That is why eledmac tries to define for both standard class and memoir class.

```

10213 \AtBeginDocument{%
10214
10215
10216 \pretocmd{\M@sect}
10217   {\let\old@edtext=\edtext%
10218    \let\edtext=\dummy@edtext@showlemma%
10219   }
10220   {}
10221   {}
10222
10223 \apptocmd{\M@sect}
10224   {\let\edtext=\old@edtext}
10225   {}
10226   {}
10227
10228 \patchcmd{\M@sect}
10229   { #9}
10230   { #9%
10231    \print@rightmargin@eledsection%
10232   }
10233   {}
10234   {}
10235
10236 \patchcmd{\M@sect}
10237   {\hskip #3\relax}
10238   {\hskip #3\relax%
10239    \print@leftmargin@eledsection%
10240   }
10241   {}
10242   {}
10243
10244 \patchcmd{\@mem@old@ssect}
10245   {#5}
10246   {#5%
10247    \print@leftmargin@eledsection%
10248   }
10249   {}
10250   {}
10251
10252 \patchcmd{\@mem@old@ssect}
10253   {\hskip #1}
10254   {\hskip #1%
10255    \print@rightmargin@eledsection%

```

```

10256 }
10257 {}
10258 {}
10259
10260
10261
10262 \patchcmd{\scr@startchapter}{\if@openright\cleardoublepage\else\clearpage\
fi}{%
10263   \if@eled@sectioning\else%
10264   \ifl@dprintingpages\else%
10265   \if@openright\cleardoublepage\else\clearpage\fi}No clearpage inside a
\Pages: will keep critical notes from printing on the title page. Here for
scrbook.
10266   \fi%
10267 \fi%
10268 }
10269 {}
10270 {}
10271
10272 \patchcmd{\@makechapterhead}
10273   {#1}
10274   {\print@leftmargin@eledsection%
10275     #1%
10276     \print@rightmargin@eledsection%
10277   }
10278   {}
10279   {}
10280
10281 \patchcmd{\@makechapterhead}% For BIDI
10282   {\if@RTL\raggedleft\else\raggedright\fi}%
10283   {\if@eled@sectioning\else%
10284     \if@RTL\raggedleft\else\raggedright\fi%
10285     \fi%
10286   }%
10287   {}%
10288   {}%
10289
10290 \patchcmd{\@makeschapterhead}
10291   {#1}
10292   {\print@leftmargin@eledsection%
10293     #1%
10294     \print@rightmargin@eledsection%
10295   }
10296   {}
10297   {}
10298
10299 \pretocmd{\@sect}
10300   {\let\old@edtext=\edtext
10301   \let\edtext=\dummy@edtext@showlemma%
10302   }
10303   {}
10304   {}
10305
10306 \apptocmd{\@sect}

```

```

10307   {\let\edtext=\old@edtext}
10308   {}
10309   {}
10310
10311 \pretocmd{\@ssect}
10312   {\let\old@edtext=\edtext%
10313   \let\edtext=\dummy@edtext@showlemma%
10314   }
10315   {}
10316   {}
10317
10318 \apptocmd{\@ssect}
10319   {\let\edtext=\old@edtext}
10320   {}
10321   {}
10322
10323 %

```

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 \H@old commands.

```

10324 \ifdef{\H@old@sect}{%
10325
10326   \patchcmd{\H@old@sect}%
10327     {#8}
10328     {#8%
10329     \print@rightmargin@eledsection%
10330     }
10331     {}
10332     {}
10333
10334   \patchcmd{\H@old@sect}%
10335     {\hskip #3\relax}
10336     {\hskip #3\relax%
10337     \print@leftmargin@eledsection%
10338     }
10339     {}
10340     {}
10341
10342   \patchcmd{\H@old@@sect}%
10343     {#5}
10344     {#5%
10345     \print@rightmargin@eledsection%
10346     }
10347     {}
10348     {}
10349
10350   \patchcmd{\H@old@@@sect}%
10351     {\hskip #1}
10352     {\hskip #1%
10353     \print@leftmargin@eledsection%
10354     }
10355     {}
10356     {}

```

```

10357 }%
10358 {
10359 \patchcmd{\@sect}
10360 {#8}
10361 {#8%
10362 \print@rightmargin@eledsection%
10363 }
10364 {}
10365 {}
10366
10367 \patchcmd{\@sect}
10368 {\hskip #3\relax}
10369 {\hskip #3\relax%
10370 \print@leftmargin@eledsection%
10371 }
10372 {}
10373 {}
10374
10375 \patchcmd{\@ssect}
10376 {#5}
10377 {#5%
10378 \print@rightmargin@eledsection%
10379 }
10380 {}
10381 {}
10382
10383 \patchcmd{\@ssect}
10384 {\hskip #1}
10385 {\hskip #1%
10386 \print@leftmargin@eledsection%
10387 }
10388 {}
10389 {}
10390 }%
10391 }%
10392 %

```

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 #.

```

10393 {}}%
10394 \protect\catcode`\#=6 %Space NEEDS by \catcode
10395 %

```

**\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.

```

10396 \AtBeginDocument{%
10397 \patchcmd{\chapter}{\clearforchapter}{%
10398 \if@eled@sectioning\else%
10399 \ifl@dprintingpages\else%
10400 \clearforchapter%
10401 \fi%
10402 \fi%

```

```

10403 }%
10404 {}%
10405 {}%
10406
10407 \patchcmd{\chapter}{\if@openright\cleardoublepage\else\clearpage\fi}{%
10408 \if@eled@sectioning\else%
10409 \ifl@dprintingpages%
10410 \endgraf%
10411 \else%
10412 \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
10413 \fi%
10414 \fi%
10415 }%
10416 {}%
10417 {}%
10418 }%
10419 %

```

`\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.

```

10420 \newif\if@eled@sectioning%
10421 %

```

We reopen a new `\notbool{@noeled@sec}` statement, as we will define the `\elesection` commands.

```

10422 \notbool{@noeled@sec}{%
10423 %

```

## 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.

```

10424 \newwrite\eled@sectioning@out
10425 %

```

`\eledchapter` And now, the user sectioning commands, which write to the file, and also add content as a “normal” line.

```

\eledsubsection
\eledsubsubsection
\eledchapter*
\eledsection*
\eledsubsection*
\eledsubsubsection*
10426 \newcommand{\eledchapter}[2] [] {%
10427 \disable@familiarnotes%
10428 #2%
10429 \restore@familiarnotes%
10430 \ifl@Rcol%
10431 \immediate\write\eled@sectioningR@out{%
\string\eled@chapter{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR}{R}
10432 }%
10433 }%
10434 \else%
10435 \immediate\write\eled@sectioning@out{%

```



```

10436     \string\eled@chapter{#1}{\unexpanded{#2}}{\the\pstarts@read@L}{-}{%
10437     }%
10438     \fi%
10439 }
10440
10441 \newcommand{\eledsection}[2] []{%
10442     \disable@familiarnotes%
10443     #2%
10444     \restore@familiarnotes%
10445     \ifledRcol%
10446         \immediate\write\eled@sectioningR@out{%
10447             \string\eled@section{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR}{-}{R}
10448         }%
10449     \else%
10450         \immediate\write\eled@sectioning@out{%
10451             \string\eled@section{#1}{\unexpanded{#2}}{\the\pstarts@read@L}{-}{%
10452             }%
10453         \fi%
10454     }
10455
10456 \newcommand{\eledsubsection}[2] []{%
10457     \disable@familiarnotes%
10458     #2%
10459     \restore@familiarnotes%
10460     \ifledRcol%
10461         \immediate\write\eled@sectioningR@out{%
10462             \string\eled@subsection{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR}{-}{R}
10463         }%
10464     \else%
10465         \immediate\write\eled@sectioning@out{%
10466             \string\eled@subsection{#1}{\unexpanded{#2}}{\the\pstarts@read@L}{-}{-}
10467         }%
10468     \fi%
10469 }
10470 \newcommand{\eledsubsubsection}[2] []{%
10471     \disable@familiarnotes%
10472     #2%
10473     \restore@familiarnotes%
10474     \ifledRcol%
10475         \immediate\write\eled@sectioningR@out{%
10476             \string\eled@subsubsection{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR
10477             }{-}{R}
10478         }%
10479     \else%
10480         \immediate\write\eled@sectioning@out{%
10481             \string\eled@subsubsection{#1}{\unexpanded{#2}}{\the\pstarts@read@L
10482             }{-}{-}{%
10483             }%
10484         \fi%
10485     }

```

```

10486 \WithSuffix\newcommand\eledchapter*[2] [] {%
10487   \disable@familiarnotes%
10488   #2%
10489   \restore@familiarnotes%
10490   \ifledRcol%
10491     \immediate\write\eled@sectioningR@out{%
10492       \string\eled@chapter{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR}{*}{R}
10493     }%
10494   \else%
10495     \immediate\write\eled@sectioning@out{%
10496       \string\eled@chapter{#1}{\unexpanded{#2}}{\the\pstarts@read@L}{*}{R}
10497     }%
10498   \fi%
10499 }
10500
10501 \WithSuffix\newcommand\eledsection*[2] [] {%
10502   \disable@familiarnotes%
10503   #2%
10504   \restore@familiarnotes%
10505   \ifledRcol%
10506     \immediate\write\eled@sectioningR@out{%
10507       \string\eled@section{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR}{*}{R}
10508     }%
10509   \else%
10510     \immediate\write\eled@sectioning@out{%
10511       \string\eled@section{#1}{\unexpanded{#2}}{\the\pstarts@read@L}{*}{R}
10512     }%
10513   \fi%
10514 }
10515
10516 \WithSuffix\newcommand\eledsubsection*[2] [] {%
10517   \disable@familiarnotes%
10518   #2%
10519   \restore@familiarnotes%
10520   \ifledRcol%
10521     \immediate\write\eled@sectioningR@out{%
10522       \string\eled@subsection{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR}{*}{
R}
10523     }%
10524   \else%
10525     \immediate\write\eled@sectioning@out{%
10526       \string\eled@subsection{#1}{\unexpanded{#2}}{\the\pstarts@read@L
}{*}{R}
10527     }%
10528   \fi%
10529 }
10530
10531 \WithSuffix\newcommand\eledsubsubsection*[2] [] {%
10532   \disable@familiarnotes%
10533   #2%
10534   \restore@familiarnotes%
10535   \ifledRcol%
10536     \immediate\write\eled@sectioningR@out{%
10537       \string\eled@subsubsection{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR

```

```

}]{*}{R}
10538   }%
10539   \else%
10540     \immediate\write\eled@sectioning@out{%
10541     \string\eled@subsubsection{#1}{\unexpanded{#2}}{\the\pstarts@read@L
}]{*}{}%
10542   }%
10543   \fi%
10544 }
10545 %

```

### 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.

```

10546 \def\eled@chapter#1#2#3#4#5{%
10547   \ifstrempy{#4}%
10548   {%
10549     \ifstrempy{#1}%
10550     {%
10551       \csgdef{eled@sectioning@#3#5}{\let\edtext=\dummy@edtext@showlemma\
chapter{#2}}%
10552       \csgdef{eled@sectmark@#3#5}{\let\edtext=\dummy@edtext{}\chaptermark
{#2}}%
10553       }%Need for \pairs, because of using parbox.
10554       {%
10555         \csgdef{eled@sectioning@#3#5}{\let\edtext=\dummy@edtext@showlemma\
chapter[#1]{#2}}%
10556         \csgdef{eled@sectmark@#3#5}{\let\edtext=\dummy@edtext{}\chaptermark
{#2}}%Need for \pairs, because of using parbox.
10557         }%
10558       }%
10559       {%
10560         \ifstrempy{#1}%
10561         {\csgdef{eled@sectioning@#3#5}{\let\edtext=\dummy@edtext@showlemma\
chapter*{#2}}}%
10562         {\csgdef{eled@sectioning@#3#5}{\let\edtext=\dummy@edtext@showlemma\
chapter*[#1]{#2}}}%Bug in LaTeX!
10563         }%
10564       \listcsgadd{eled@sections#5@@}{#3}%
10565     }
10566 \def\eled@section#1#2#3#4#5{%
10567   \ifstrempy{#4}%
10568   {\ifstrempy{#1}%

```

```

10569     {%
10570     \csgdef{eled@sectioning@#3#5}{\section{#2}}%
10571     \csgdef{eled@sectmark@#3#5}{\let\edtext=\dummy@edtext{}\sectionmark
{#2}}}%Need for \pairs, because of using parbox.
10572     }%
10573     {%
10574     \csgdef{eled@sectioning@#3#5}{\section[#1]{#2}}%
10575     \csgdef{eled@sectmark@#3#5}{\let\edtext=\dummy@edtext{}\sectionmark
{#1}}}%Need for \pairs, because of using parbox.
10576     }%
10577     }%
10578     {\ifstrempy{#1}%
10579     {\csgdef{eled@sectioning@#3#5}{\section*{#2}}}%
10580     {\csgdef{eled@sectioning@#3#5}{\section*{#1}{#2}}}%Bug in LaTeX!
10581     }
10582     \listcsgadd{eled@sections#5@0}{#3}%
10583     }
10584     \def\eled@subsection#1#2#3#4#5{%
10585     \ifstrempy{#4}%
10586     {\ifstrempy{#1}%
10587     {%
10588     \csgdef{eled@sectioning@#3#5}{\subsection{#2}}%
10589     \csgdef{eled@sectmark@#3#5}{\let\edtext=\dummy@edtext{}\csuse{
subsectionmark}{#2}}}%Need for \pairs, because of using parbox. \csuse in
case of \subsectionmark is not defined (book)
10590     }%
10591     {%
10592     \csgdef{eled@sectioning@#3#5}{\subsection[#1]{#2}}%
10593     \csgdef{eled@sectmark@#3#5}{\let\edtext=\dummy@edtext{}\csuse{
subsectionmark}{#1}}}%Need for \pairs, because of using parbox. \csuse in
case of \subsectionmark is not defined (book)
10594     }%
10595     }%
10596     {\ifstrempy{#1}%
10597     {\csgdef{eled@sectioning@#3#5}{\subsection*{#2}}}%
10598     {\csgdef{eled@sectioning@#3#5}{\subsection*{#1}{#2}}}%Bug in LaTeX!
10599     }
10600     \listcsgadd{eled@sections#5@0}{#3}%
10601     }
10602     \def\eled@subsubsection#1#2#3#4#5{%
10603     \ifstrempy{#4}%
10604     {\ifstrempy{#1}%
10605     {\csgdef{eled@sectioning@#3#5}{\subsubsection{#2}}}%
10606     {\csgdef{eled@sectioning@#3#5}{\subsubsection[#1]{#2}}}%
10607     }%
10608     {\ifstrempy{#1}%
10609     {\csgdef{eled@sectioning@#3#5}{\subsubsection*{#2}}}%
10610     {\csgdef{eled@sectioning@#3#5}{\subsubsection*{#1}{#2}}}%Bug in
LaTeX!
10611     }
10612     \listcsgadd{eled@sections#5@0}{#3}%
10613     }
10614     %
10615     %

```

End of the conditional test about noledsec option.

```
10616 }{}
10617 %
```

## XXXV Page breaking or no page breaking depending on 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.

```
10618 \def\normal@page@break{}
10619 %
```

`\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.

```
10620 \def\l@prev@pb{}
10621 \def\l@prev@nopb{}
10622 %
```

`\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.

```
10623 \newcommand{\ledpb}{\write\linenum@out{\string\led@pb}}
10624 \newcommand{\ledpbnum}[1]{\write\linenum@out{\string\led@pbnum{#1}}}
10625 \newcommand{\lednopb}{\write\linenum@out{\string\led@nopb}}
10626 \newcommand{\lednopbnum}[1]{\write\linenum@out{\string\led@nopbnum{#1}}}
10627 %
```

`\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.

```
10628 \newcommand{\led@pb}{\listxadd{\l@prev@pb}{\the\absline@num}}
10629 \newcommand{\led@pbnum}[1]{\listxadd{\l@prev@pb}{#1}}
10630 \newcommand{\led@nopb}{\listxadd{\l@prev@nopb}{\the\absline@num}}
10631 \newcommand{\led@nopbnum}[1]{\listxadd{\l@prev@nopb}{#1}}
10632 %
```

`\ledpbsetting` The `\ledpbsetting` macro only changes the value of `\led@pb@macro`, for which the default value is before.

```

10633 \def\led@pb@setting{before}
10634 \newcommand{\ledpbsetting}[1]{\gdef\led@pb@setting{#1}}
10635 %

```

`\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`.

```

10636 \newcommand{\led@check@pb}{\xifinlist{\the\absline@num}{\l@prev@pb}{\
pagebreak[4]}}
10637 \newcommand{\led@check@nopb}{%
10638   \IfStrEq{\led@pb@setting}{before}{%
10639     \xifinlist{\the\absline@num}{\l@prev@nopb}%
10640     {\numdef{\abs@prevline}{\the\absline@num-1}%
10641     \xifinlist{\abs@prevline}{\normal@page@break}%
10642     {\nopagebreak[4]\enlargethispage{\baselineskip}}%
10643     {}}%
10644   }%
10645   {}%
10646 }%
10647 \IfStrEq{\led@pb@setting}{after}{%
10648   \xifinlist{\the\absline@num}{\l@prev@nopb}{%
10649     \xifinlist{\the\absline@num}{\normal@page@break}%
10650     {\nopagebreak[4]\enlargethispage{\baselineskip}}%
10651     {}}%
10652 }%
10653   {}}%
10654   {}%
10655 }%
10656 }
10657 %

```

## 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.

```

10658 \newcommand{\check@pb@in@verse}{%
10659   \ifinstanza\iflednopbinverse\ifinserthangingsymbol% Using stanzas and
enabling page breaks in verse control, while on a hanging verse.
10660   \ifnum\page@num=\last@page@num\else%If we have change page
10661     \IfStrEq{\led@pb@setting}{before}{%

```

```

10662         \numgdef{\abs@line@verse}{\the\absline@num-1}%
10663         \ledpbnum{\abs@line@verse}%
10664     }{}%
10665     \IfStrEq{\led@pb@setting}{after}{%
10666         \numgdef{\abs@line@verse}{\the\absline@num-1}%
10667         \lednopbnum{\abs@line@verse}%
10668     }{}%
10669     \fi%
10670 \fi\fi\fi%
10671 }
10672 %

```

## 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>

```

10673 \def\Hy@raisedlink@left#1{%
10674     \ifvmode
10675         #1%
10676     \else
10677         \Hy@SaveSpaceFactor
10678         \llap{\smash{%
10679             \begingroup
10680                 \let\HyperRaiseLinkLength\@tempdima
10681                 \setlength\HyperRaiseLinkLength\HyperRaiseLinkDefault
10682                 \HyperRaiseLinkHook
10683             \expandafter\endgroup
10684             \expandafter\raise\the\HyperRaiseLinkLength\hbox{%
10685                 \Hy@RestoreSpaceFactor
10686                 #1%
10687                 \Hy@SaveSpaceFactor
10688             }}%
10689         \Hy@RestoreSpaceFactor
10690         \penalty\@M\hskip\z@\relax
10691     \fi
10692 }
10693 %
10694 %

```

To deal with `footnotemarkX` and `\footnotetext` commandes in relation with hyperref, we need to generate unique identifier for footnote, called `hr@thefnmarkX` (where X stands for the footnote series letter).

But to generate this unique number, we need to know when the `footnoteX` counter is reset. So we need to patch both `\setcounter` and `\@stpelt` macros.

<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>.

```

10695
10696 \apptocmd{\setcounter}{%
10697   \ifboolexpr{%
10698     test {\ifnumequal{#2}{0}}%
10699     and%
10700     test {\ifcsdef{#1@series}}%
10701   }%
10702   {%
10703     \global\advance\expandafter\csname #1@series\endcsname by 1%
10704   }%
10705   {}%
10706 }%
10707   {}%
10708   {\led@error@fail@patch@setcounter}%
10709 \apptocmd{\@stpelt}{%
10710   \ifcsdef{#1@series}{%
10711     \global\advance\expandafter\csname #1@series\endcsname by 1%
10712   }%
10713   {}%
10714 }%
10715   {}%
10716   {\led@error@fail@patch@stpelt}%
10717 %

```

## XXXVIII Compatibility with eledmac

Here, we define some commands for the eledmac-compat option.

```

10718 \ifeledmaccompat@%
10719
10720 \newcommand{\footnormalX}[1]{\arrangementX[#1]{normal}}%
10721 \newcommand{\footparagraphX}[1]{\arrangementX[#1]{paragraph}}%
10722 \newcommand{\foottwocolX}[1]{\arrangementX[#1]{twocol}}%
10723 \newcommand{\footthreecolX}[1]{\XarrangementX[#1]{threecol}}%
10724
10725 \unless\ifnocritical@
10726   \newcommand{\footnormal}[1]{\Xarrangement[#1]{normal}}%
10727   \newcommand{\footparagraph}[1]{\Xarrangement[#1]{paragraph}}%
10728   \newcommand{\foottwocol}[1]{\Xarrangement[#1]{twocol}}%
10729   \newcommand{\footthreecol}[1]{\Xarrangement[#1]{threecol}}%
10730   \let\hsizetwocol\Xhsizetwocol
10731   \let\hsizethreecol\Xhsizethreecol
10732   \let\bhookXnote\Xbhooknote
10733   \let\boxsymlinenum\Xboxsymlinenum
10734   \let\symlinenum\Xsymlinenum
10735   \let\beforenumberinfootnote\Xbeforenumber
10736   \let\afternumberinfootnote\Xafternumber
10737   \let\beforeXsymlinenum\XbeforeXsymlinenum
10738   \let\afterXsymlinenum\XafterXsymlinenum
10739   \let\inplaceofnumber\Xinplaceofnumber
10740   \let\Xlemmaseparator\lemmaseparator
10741   \let\afterlemmaseparator\Xafterlemmaseparator
10742   \let\beforelemmaseparator\Xbeforelemmaseparator

```



```

10743 \let\inplaceoflemmaseparator\Xinplaceoflemmaseparator
10744 \let\txtbeforeXnotes\Xtxtbeforenotes
10745 \let\afterXrule\Xafterrule
10746 \let\numberonlyfirstinline\Xnumberonlyfirstinline
10747 \let\numberonlyfirstintwlines\Xnumberonlyfirstintwlines
10748 \let\nonumberinfootnote\Xnonumberinfootnote
10749 \let\pstartinfootnote\Xpstart
10750 \let\pstartinfootnoteeverytime\Xpstarteverytime
10751 \let\onlyXpstart\Xonlypstart
10752 \let\Xnonumberinfootnote\Xnonumber
10753 \let\Xnonbreakableafternumber\Xnonbreakableafternumber
10754 \let\XmaxhXnotes\Xmaxhnotes
10755 \let\XbeforeXnotes\Xbeforenotes
10756 \let\Xboxlinenum\Xboxlinenum
10757 \let\Xboxlinenumalign\Xboxlinenumalign
10758 \let\Xboxstartlinenum\Xboxstartlinenum
10759 \let\Xboxendlinenum\Xboxendlinenum
10760 \let\Xtwolines\Xtwolines
10761 \let\Xmorethantwlines\Xmorethantwlines
10762 \let\Xtwolinesbutnotmore\Xtwolinesbutnotmore
10763 \let\Xtwolinesonlyinsamepage\Xtwolinesonlyinsamepage
10764 \fi
10765
10766 \unless\ifnofamiliar@
10767 \let\XnotesXwidthliketwocolumns\XnoteswidthliketwocolumnsX
10768 \fi
10769 \newcommandx{\parafootsep}[2][1,usedefault]{%
10770 \Xparafootsep[#1]{#2}%
10771 \parafootsepX[#1]{#2}
10772 }%
10773
10774 \newcommandx{\afternote}[2][1,usedefault]{%
10775 \Xafternote[#1]{#2}%
10776 \afternoteX[#1]{#2}%
10777 }%
10778
10779 \unless\ifnoend@
10780 \let\XendXtwolines\Xendtwolines
10781 \let\XendXmorethantwlines\Xendmorethantwlines
10782 \let\XhookXendnote\Xendbhooknote
10783 \let\XboxXendlinenum\Xendboxlinenum%
10784 \let\XboxXendlinenumalign\Xendboxlinenumalign%
10785 \let\XboxXendstartlinenum\Xendboxstartlinenum%
10786 \let\XboxXendendlinenum\Xendboxendlinenum%
10787 \let\XendXlemmaseparator\Xendlemmaseparator
10788 \let\XendXbeforelemmaseparator\Xendbeforelemmaseparator
10789 \let\XendXafterlemmaseparator\Xendafterlemmaseparator
10790 \let\XendXinplaceoflemmaseparator\Xendinplaceoflemmaseparator
10791 \fi
10792
10793 \AtBeginDocument{%
10794 \ifdef\lineref{}\let\lineref\edlineref}%
10795 }%
10796

```

```
10797  
10798 \fi%  
10799 %
```

</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:

<code>I saw my friend \critext{Smith}</code>	1 I saw my friend
<code>\Afootnote{Jones C, D.}/</code>	2 <u>Smith on Tuesday.</u>
on Tuesday.	2 <u>Smith]</u> Jones C, D.

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

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

<code>\critext{I saw my friend</code>	1 I saw my friend
<code>\critext{Smith}{\Afootnote{Jones</code>	2 <u>Smith on Tuesday.</u>
<code>C, D.}/ on Tuesday.}</code>	2 <u>Smith]</u> Jones C, D.
<code>\Bfootnote{The date was</code>	1–2 I saw my friend
<code>July 16, 1954.}</code>	Smith on Tuesday.] The
<code>/</code>	date was July 16, 1954.

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

The second argument of the `\critext` macro, `⟨commands⟩`, is the same as the second argument to the `\edtext` macro.

It is possible to define aliases for `\critext`, which can be easier to type. You can make a single character substitute for `\critext` by saying this:

```
\catcode'\<=\active
\let<=\critext
```

<sup>40</sup>A name like `\text` is likely to be defined by other  $\text{\LaTeX}$  packages (it certainly is by the AMS packages) and it seems sensible to try and avoid clashes with other definitions.

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. 130 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. 39), and remove any `\Xfootnote` command.

If you have customized the `\XXXXXXfmt` command, please check whether you can achieve the same by the commands documented for display options (7 p. 40) or `\Xfootnote` options (6.2.2 p. 28). 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 `\XXXXXXfmt` command, but you must replace:

```
\renewcommand*{XXXXfmt}[3]
```

with

```
\renewcommandx*{XXXXfmt}[4][4=Z]
```

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.

<sup>41</sup><https://github.com/maieul/ledmac/issues>

### A.3 Migration to *eledmac* 1.5.1

The version 1.5.1 corrects a bug in `stanzaindentsrepetition` (cf. 9.3 p. 55). 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 introduces a better way to handle sectional divisions inside numbered text. Please read 16.2 p. 71.

### A.5 Migration to *eledmac* 17.1

This version changes the default setting of `\Xpstart`. Henceforth, `pstart` numbers will be printed in footnotes within the section of text where you have called `\numberpstarttrue`.

We do not see any reason to print them in the other sections. However, if you want to print the `\pstart` numbers in every footnote, whatever the section, without having to use `\numberpstarttrue`, you can use `\Xpstarteverytime`.

## A.6 Migration to eledmac 1.21.0

### A.6.1 `\Xledsetnormalparstuff` and `\ledsetnormalparstuffX`

The `\ledsetnormalparstuff` has been split into two different commands:

- `\Xledsetnormalparstuff` for critical notes;
- `\ledsetnormalparstuffX` for familiar notes.

Both commands can take an optional argument which is the series letter. If you have redefined `\ledsetnormalparstuff` or any of the commands which call them, you must change them accordingly.

### A.6.2 Endnotes

In any case, delete the `.end` file before the next run.

The previous version of Eledmac had a bug: there were two spaces between the starting page number and the starting line number, but only one space between the ending page number and the ending line number.

As a matter of fact, a spurious space was added after the first `\printnpnum`. This spurious space has been deleted. However, if you want to keep the previous spurious space, you may load the package with the `oldprintnpnumspace` option.

If you have redefined `\endprint`, you must:

- Contact us and ask for the feature that required your hack, in order to avoid such a hack in the future.
- Use the new fifth argument.
- Add `\xdef\@currentseries{#4}` at the beginning of your own command.

## A.7 Migration to eledmac 1.22.0

The `\ledinnote` command now takes a first optional argument, which is the label for the hyperreference. If you have redefined it, change your redefinition, and check whether you can avoid this redefinition by only redefining `\ledinnotemark`.

## A.8 Migration to eledmac 1.23.0

You must delete the numbered auxiliary files before compiling with the new version of eledmac.

## A.9 Migration from eledmac to reledmac

There are many changes in reledmac which require the user to make modifications.

### A.9.1 Risk of ‘no room for a new’

The risk to obtain a ‘no room for a new something’ error is greater in reledmac than it is in eledmac. See [19.1.3 p. 74](#) in order to know how to limit it.

### A.9.2 Multiple indices with memoir

`Eledmac` and `ledmac` used the specific indexing tools of the `memoir` class designed to produce multiple indices. However, `eledmac` could also use `imakeidx` or `indextools` tools independently of the `memoir` class. This system forced to maintain redundant code. Since `reledmac`, we use only the `imakeidx` or `indextools` tools.

Consequently: Users of `memoir` are invited to use `indextool` or `imakeidx` to produce multiple indices.

### A.9.3 Deprecated commands and options

The table of deprecated commands and their alternatives follows. Note that the way some commands must be used may have changed. Please read the handbook.

<i>Deprecated command</i>	<i>Replaced with</i>
<code>\addfootins</code>	<code>\newseries</code>
<code>\addfootinsX</code>	<code>\newseries</code>
<code>\critext</code>	<code>\edtext</code>
<code>\falseverse</code>	<code>\newverse</code>
<code>\interparanoteglua</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 `ledsecnolinenum` option has been removed, because it was related to deprecated commands.

The `oldprintnnumspace` 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>

<i>Deprecated \renewcommand</i>	<i>Replaced with</i>	
<code>\notefontsetup</code>	<code>\Xnotefontsize,</code> <code>\Xendnotefontsize</code>	and
	<code>\notefontsizeX</code>	
<code>\sidenotesep</code>	<code>\setsidenotesep</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>
<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>\boxsymplinenum</code>	<code>\Xboxsymplinenum</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>



<i>Old command</i>	<i>New command</i>
<code>\morethantwolines</code>	<code>\Xmorethantwolines</code>
<code>\nonumberinfootnote</code>	<code>\Xnonumber</code>
<code>\notesXwidthliketwocolumns</code>	<code>\noteswidthliketwocolumnsX</code>
<code>\noXlemmaseparator</code>	<code>\Xnolemmaseparator</code>
<code>\numberonlyfirstinline</code>	<code>\Xnumberonlyfirstinline</code>
<code>\numberonlyfirstintwolines</code>	<code>\Xnumberonlyfirstintwolines</code>
<code>\nonbreakableafternumber</code>	<code>\Xnonbreakableafternumber</code>
<code>\onlyXpstart</code>	<code>\Xonlypstart</code>
<code>\parafootsep</code>	<code>\Xparafootsep</code> and <code>\parafootsepX</code>
<code>\pstartinfootnote</code>	<code>\Xpstart</code>
<code>\pstartinfootnoteeverytime</code>	<code>\Xpstarteverytime</code>
<code>\symlinenum</code>	<code>\Xsymlinenum</code>
<code>\twolines</code>	<code>\Xtwolines</code>
<code>\twolinesbutnotmore</code>	<code>\Xtwolinesbutnotmore</code>
<code>\twolinesonlyinsamepage</code>	<code>\Xtwolinesonlyinsamepage</code>
<code>\txtbeforeXnotes</code>	<code>\Xtxtbeforenotes</code>
<code>\XendXafterlemmaseparator</code>	<code>\Xendafterlemmaseparator</code>
<code>\XendXbeforelemmaseparator</code>	<code>\Xendbeforelemmaseparator</code>
<code>\XendXinplaceoflemmaseparator</code>	<code>\Xendinplaceoflemmaseparator</code>
<code>\XendXlemmaseparator</code>	<code>\Xendlemmaseparator</code>
<code>\XendXmorethantwolines</code>	<code>\Xendmorethantwolines</code>
<code>\XendXtwolines</code>	<code>\Xendtwolines</code>
<code>\Xnonumberinfootnote</code>	<code>\Xnonumber</code>
<code>\lineref</code>	<code>\edlineref</code>

### A.9.6 Endnotes

With reledmac, there is now one auxiliary file for each endnotes set (.Aend, .Bend, .Cend etc.). If you have overridden `\doendnotes` (which you should not have done) you must adapt your code.

### A.9.7 Z Series

The ‘Z’ series of notes has been removed. Only five series are provided now by default: A, B, C, D, E.

### A.9.8 Internal commands

Users who have overridden internal commands, which is wrong, must adapt according to the following. Or better, they should not override any of such commands and use reledmac options instead.

- If you have modified `\Xfootfmt`, note that the fourth argument is now mandatory.
- `\unvxh` has been replaced with `\Xunvxh` and `\unvxhX` with two mandatory arguments.

## A.10 Migration to reledmac 2.1.0

Reledmac 2.1.0 fix some bugs when using `\Xbhooknote` and `\bhooknoteX` not in order to execute code at the beginning of each notes, but to insert content of at the beginning

of each notes.

People who use these commands to do it, which is not the original idea, must change the following:

1. Horizontal space is no longer automatically added after the content of the `\Xbhooknote/\bhooknoteX` argument. You must include it manually. So instead of `\Xbhooknote{content}`, use `\Xbhooknote{content }.`
2. Indent is no longer automatically added before the content of the `\Xbhooknote/\bhooknoteX` argument. If you want to keep it, add `\indent` to the argument of `\Xbhooknote/\bhooknoteX`.

### A.11 Migration to reledmac 2.1.3

Reledmac 2.1.3 fix an historical bug, (style in ledmac 0.7!) which doubled the space before the rules of paragraphed familiar footnotes. Consequently, if you use paragraphed familiar footnotes, you should maybe adapt it, playing with `\beforenotesX`.

### A.12 Migration to reledmac 2.3.0

Before reledmac 2.3.0, for typesetting verse, any empty line was considered a paragraph inside verses. Counting empty lines this created breaking verse, hanging verses, and also added spurious vertical spaces. Version 2.3.0 disables paragraph in stanza. If you want vertical space, use the optional argument of `\stanza` or `\endverse`.

### A.13 Migration to reledmac 2.4.0

It is not mandatory, but strongly recommended, to change any `\renewcommand{\endashchar}{⟨...⟩}` to the use of `\Xlinerrangeseparator` or `/` and `\Xendlinerrangeseparator` (7.4.4 p. 43).

### A.14 Migration to reledmac 2.5.0

It is strongly recommended to stop redefining `\printnpnum` and to use the hooks documented in 7.4.3 p. 42.

`\xlineref` does not print anymore the side flag (R for right side), because it is incompatible with numerical test. Use `\xflagref` to obtain it.

The `\printlines` and `\printendlines` commands take now an eighth argument, which is the side flag. It is strongly recommended to NEVER redefine these two commands and to use the setting commands instead (or to ask for new setting commands if the actual does not answer to your needs). However, if you have done it, just change your redefinition to have a new argument.

It is strongly recommended to stop redefining `\fullstop` and to use `\Xsublinesep` instead.

### 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 `\Serefwitpage`. That has been deleted. And so, the `\XendafterpagenumberSerefonlypage` to set it was also deleted.

`\rigidbalance` is split to two new commands: `\Xrigidbalance` for critical footnotes and `\rigidbalanceX` for familiar footnotes. If you have redefined it – but why should you have? –, you should split your single redefinition in two redefinitions.

### A.16 Migration to reledmac 2.7.2

`\Xhsize` is already defined in the floatrow package. It becomes `\Xwidth`, and, consequently, `\hsizeX` becomes `\widthX`.

The ancient names are temporarily maintained as aliases.

### A.17 Migration to reledmac 2.8.0

Reledmac 2.8.0 fix spurious indents for paragraphed critical and familiar footnotes in `ledgroup` and `minipage`. You can re-establish the indent with `\Xparindent` and `\parindentX`.

### A.18 Migration to reledmac 2.13.1

Reledmac 2.5.0 added a bug, which makes the right flag to be printed on the right side of critical footnotes, even if not explicitly requested by using `\Xlineflag`.

Version 2.13.1 solves this issue. Please use `\Xlineflag` if you want to add the right flag.

### A.19 Migration to reledmac 2.18.0

After updating reledmac, and before any new compilation, you need to clean your `.aux` files, if you use `\edlabel` or related.

### A.20 Migration to reledmac 2.21.0

Previously, there was a bug, which meant that the description in the handbook was incorrect. If you wrote

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

“elephant” was indexed in the main text and in the critical footnotes. With the new version of reledmac, it is indexed only in main text. If you also want to index it in critical footnotes, do

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

### A.21 Migration to reledmac 2.24.0

When using `\labelpstarttrue`, a spurious space<sup>a</sup> was introduced after the `pstart` number (only for normal typesetting, not while typesetting in parallel). The new version of the package has deleted this spurious space. If you consider that it was NOT a spurious space, you should add it manually in your definition of `\thepstart`.

**A.22 Migration to reledmac 2.26.0**

You must delete your `.aux` file after having upgraded to this new version of `reledmac`.

**A.23 Migration to reledmac 2.27.1**

This release fixes spurious space in `\hidenumbering`. If you considered this spurious space as normal, you must insert it manually using.

```
xx\hidenumbering\ xx
```

To get the space after `xx`.

**A.24 Migration to reledmac 2.30.0**

If you have multiple annotations for the same line, these are now separated with a comma in the margin. You can use `\setlinenumannotationsep` to change the separator.

**A.25 Migration to reledmac 2.31.1**

If you use `\Xgroupbyline`, the indentation of the notes has been deleted, in order to have the same behavior as for normal critical notes.

Use `\Xparindent` to restore indentation.

**A.26 Migration to reledmac 2.32.7**

For paragraphed familiar footnotes, the separator between notes was `\Xparafootsep` instead of `\parafootsepX`. This release fixes this bug. You may adapt your settings.

**A.27 Migration to reledmac 2.33.0**

You should not redefine anymore `\ledinnotemark`, but use instead `\innotemarkX` and `\Xinnotemark`.

If you have personal script which parses for `\ledinnote` or if you redefine `\ledinnote` (but normally, you should not), this macro have new arguments. Look at the code for more details.

**Appendix B Auxiliary softwares**

This appendix lists some software and tools related to `reledmac` that may be useful.

**B.1 *samewords***

The *samewords* program (Michael Stenskjær Christensen) automatically adds `\sameword` commands (6.3 p. 31) into a `.tex` file.

<https://samewords.readthedocs.io/en/latest/>

## B.2 **critical-keys** for *Emacs*

The *Emacs* editor has a package to help inserting `reledmac`'s commands (Juan Manuel Macías Chaín).

<https://gitlab.com/maciaschain/critical-keys>

## B.3 **critical-marks** for *Emacs*

A set of functions for *Emacs* that run in the export process from Org Mode to LaTeX and replace a series of simple textual marks with the basic `reledmac`'s commands (Juan Manuel Macías Chaín) <https://gitlab.com/maciaschain/critical-marks>

## B.4 **Import from TEI**

## B.5 **Import from TEI**

There are multiple tools to convert from XML-TEI to *reledmac*. Here is a non exhaustive-list:

- <http://ciham-digital.huma-num.fr/teicat/>
- [https://github.com/fizzbucket/tei\\_transformer](https://github.com/fizzbucket/tei_transformer)
- <https://github.com/TEIC/Stylesheets>
- <http://lombardpress.org/print/>
- <https://github.com/Jean-Baptiste-Camps/TEItoLaTeX>

Please, tell us if you know of other tools.

## References

- [Bre96] Herbert Breger. `tabmac`. October 1996. (Available from CTAN in `macros/plain/contrib/tabmac`)
- [Bur01] John Burt. “Typesetting critical editions of poetry”. *TUGboat*, **22**, 4, pp 353–361, December 2001. (Code available from CTAN in `macros/latex/contrib/poemscol`)
- [Eck03] Matthias Eckermann. *The Parallel-Package*. April 2003. (Available from CTAN in `macros/latex/contrib/parallel`)
- [Fai03] Robin Fairbairns. *footmisc—a portmanteau package for customising footnotes in L<sup>A</sup>T<sub>E</sub>X*. February 2003. (Available from CTAN in `macros/latex/contrib/footmisc`)
- [LW90] John Lavagnino and Dominik Wujastyk. “An overview of EDMAC: a PLAIN TeX format for critical editions”. *TUGboat*, **11**, 4, pp. 623–643, November 1990. (Code available from CTAN in `macros/plain/contrib/edmac`)
- [Lüc03] Uwe Lück. “ednotes—critical edition typesetting with LaTeX”. *TUGboat*, **24**, 2, pp. 224–236, 2003. (Code available from CTAN in `macros/latex/contrib/ednotes`)
- [Sul92] Wayne G. Sullivan. *The file edstanza.doc*. June 1992. (Available from CTAN in `macros/plain/contrib/edmac`)
- [Wil02] Peter Wilson. *The memoir class for configurable typesetting*. November 2002. (Available from CTAN in `macros/latex/contrib/memoir`)
- [Wil04] Peter Wilson and Maïeul Rouquette. *Parallel typesetting for critical editions: the eledpar package*. December 2004. (Available from CTAN in `macros/latex/contrib/ledmmac`)

## Index

### Symbols

<code>\&amp;</code> .....	54
<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, 1
<code>\@getfirstseries</code>	1
<code>\@gobblefour</code>	1
<code>\@gobbleseven</code>	1
<code>\@gobblethree</code>	1
<code>\@h</code>	1
<code>\@changingsymbol</code>	1
<code>\@iiminipage</code>	1
<code>\@insertstanzanumber</code>	1
<code>\@insidethisline</code>	1
<code>\@k</code>	1
<code>\@l@dttempcnta</code>	1
<code>\@l@dttempcntb</code>	1
<code>\@lab</code>	1
<code>\@led@testifnofoot</code>	1
<code>\@lemma</code>	1
<code>\@line@num</code>	1
<code>\@linenumannotationsep</code>	1
<code>\@lock</code>	1
<code>\@lopL</code>	1
<code>\@lopR</code>	1
<code>\@makechapterhead</code>	1, 1
<code>\@makeschapterhead</code>	1
<code>\@mem@extranofeet</code>	1
<code>\@mem@old@ssect</code>	1
<code>\@mpfnpos</code>	1
<code>\@msd</code>	1
<code>\@msd@c</code>	1
<code>\@msd@options@iffullpage</code>	1
<code>\@msdata@list</code>	1
<code>\@nl</code>	1
<code>\@nl@reg</code>	1
<code>\@opXfeet</code>	1
<code>\@outputpage</code>	1
<code>\@page</code>	1
<code>\@pend</code>	1
<code>\@pendR</code>	1
<code>\@ref</code>	1
<code>\@ref@reg</code>	1
<code>\@ref@reg@parse</code>	1
<code>\@resetannot</code>	1
<code>\@sect</code>	1
<code>\@series</code>	1
<code>\@set</code>	1
<code>\@sidenotesep</code>	1
<code>\@ssect</code>	1
<code>\@startstanza</code>	1
<code>\@stopmsd</code>	1
<code>\@stopstanza</code>	1
<code>\@sw</code>	1
<code>\@tag</code>	1, 1
<code>\@wredindex</code>	1

\@xloop	1
\@xypar	1
CLASSarticle	74
CLASSbook	74, 364
CLASSmemoir	214, 215, 277–280, 314, 364, 383, 444, 447
CLASSscrbook	448
COMMAND\*footnote	75
COMMAND\...@footnotemark...	217
COMMAND\...d@ta	158
COMMAND\<hook	
@<series	264
COMMAND\<hookname	
<pseudoserries	266, 267
COMMAND\<type	
footfmt	201
COMMAND\@@line	191
COMMAND\@MM	179, 445
COMMAND\@Rlineflag	316, 445
COMMAND\@SErefprefix	294
COMMAND\@SErefprefixmore	294
COMMAND\@add@	354
COMMAND\@adv	114
COMMAND\@annot	164
COMMAND\@annot@end@print	204
COMMAND\@annot@start@print	204, 209, 246
COMMAND\@apprefprefixmore	294
COMMAND\@apprefprefixsingle	294
COMMAND\@beforeinsertofthisedtext	137, 138
COMMAND\@bsphack	282
COMMAND\@doclearpage	278, 279, 438, 447
COMMAND\@doreinfeetX	447
COMMAND\@dprintingcolumns	444
COMMAND\@edindex@hyperref	316, 317
COMMAND\@edtext@level	134
COMMAND\@esphack	282
COMMAND\@firstofone	35
COMMAND\@fnpos	235, 275
COMMAND\@footnotemark	215, 438, 447
COMMAND\@footnotetext	215, 216, 438
COMMAND\@gobble	26, 35, 132, 164, 259
COMMAND\@gobblefive	446
COMMAND\@gobblefour	444
COMMAND\@gobbleseven	261
COMMAND\@gobblethree	437
COMMAND\@h	194
COMMAND\@hangingsymbol	322
COMMAND\@iiiminipage	306, 308, 437, 447
COMMAND\@iiiminpage	306
COMMAND\@l	442
COMMAND\@l@dtempcnta	161, 166, 173
COMMAND\@l@dtempcntb	166
COMMAND\@l@reg	442
COMMAND\@lab	110, 282, 285, 289, 437
COMMAND\@ldunboxmpfoot	308



COMMAND\@led@extranofeet	383
COMMAND\@ledinnote@command	312, 313
COMMAND\@lemma	137, 140
COMMAND\@lock	105, 322
COMMAND\@lopL	438
COMMAND\@lopR	438
COMMAND\@makecol	274, 275, 277, 278, 447
COMMAND\@mempnum	280
COMMAND\@mpfnpos	235
COMMAND\@msd	332
COMMAND\@msd@c	332
COMMAND\@msd@options@iffullpage	338
COMMAND\@msdata@list	332, 333
COMMAND\@nl	110–112, 114, 116, 125, 165, 285, 437, 438
COMMAND\@nl@reg	111, 381, 438, 442
COMMAND\@opXfeet	438
COMMAND\@opfeetX	447
COMMAND\@opxtrafeeti	447
COMMAND\@outputpage	280, 281
COMMAND\@page	111, 112, 285
COMMAND\@pend	438
COMMAND\@pendR	438
COMMAND\@ref	110, 120–122, 126, 132
COMMAND\@ref@later	122, 127
COMMAND\@ref@reg	121, 164, 438
COMMAND\@ref@reg@parsearg	122
COMMAND\@reinserts	274, 276, 277, 447
COMMAND\@resetannot	165
COMMAND\@secondoftwo	76
COMMAND\@sect	366
COMMAND\@series	263
COMMAND\@set	115
COMMAND\@sidenotesep	304
COMMAND\@stopmsd	333
COMMAND\@stpelt	375
COMMAND\@sw	122, 141, 145
COMMAND\@tag	133, 135, 139
COMMAND\@tempcnta	90
COMMAND\@tempcntb	90
COMMAND\@this@crossref@start	214
COMMAND\@toksa	96
COMMAND\@toksb	96
COMMAND\@xloop	174
COMMAND\@xympar	298, 447
COMMAND\Aendnote	16, 29
COMMAND\Afootfmt	178
COMMAND\Afootgroup	178
COMMAND\Afootnote	8, 16, 27, 28, 31, 135, 186, 214, 236, 254, 447
COMMAND\Afootstart	178
COMMAND\AtBeginDocument	278
COMMAND\AtEndEveryPend	18, 453
COMMAND\AtEveryPend	18, 57, 152, 444, 445, 447, 453
COMMAND\AtEveryPend*	18
COMMAND\AtEveryPstart	18, 57, 444, 445, 447, 450, 453

COMMAND\AtEveryPstart*	18
COMMAND\AtEveryStanza	57, 450, 453
COMMAND\AtEveryStopStanza	57, 450, 453
COMMAND\AtStartEveryPstart	18, 453
COMMAND\AtStartEveryStanza	57, 453
COMMAND\BeforeEveryStopStanza	57, 453
COMMAND\Bendnote	16, 27
COMMAND\Bfootnote	8, 16, 214, 236, 254
COMMAND\Centering	50
COMMAND\Cfootnote	214
COMMAND\Columns	90, 148, 183
COMMAND\Dfootnote	214
COMMAND\Efootnote	214
COMMAND\Gls	68
COMMAND\H@old	366
COMMAND\Hy@raisedlink	375
COMMAND\Hy@raisedlink@left	375
COMMAND\LTR	50
COMMAND\Pages	90, 148, 275, 276, 279
COMMAND\ProcessOptionsX	79
COMMAND\RL	48
COMMAND\RaggedLeft	50
COMMAND\RaggedRight	50
COMMAND\RenewExpandableDocumentCommand	36, 144
COMMAND\SEonlypage	291, 449
COMMAND\SEref	61–64, 291, 295, 450, 452, 455
COMMAND\SErefonlypage	61–63, 386, 449
COMMAND\SErefwithpage	61, 63, 291, 295, 386, 449, 451, 456
COMMAND\Stanza	443
COMMAND\Waklam	355
COMMAND\X@doreinfeet	276, 447
COMMAND\XR@prefix	298
COMMAND\XR@test	297, 298
COMMAND\XR@test@mac	298
COMMAND\XR@test@mac@test	298
COMMAND\XXXXXXfmt	380
COMMAND\XXXXXXfmt	380
COMMAND\Xafterrenumber	45
COMMAND\Xafterlemmaseparator	47, 384
COMMAND\Xafternote	51, 383, 384
COMMAND\Xafternumber	45, 384
COMMAND\Xafterrule	52, 236, 384, 442, 445
COMMAND\Xaftersymmlinenumber	45, 384
COMMAND\Xarrangement	40, 51, 75, 179, 180, 265, 384
COMMAND\Xarrangement@footparagraph	184
COMMAND\Xarrangement@normal	180
COMMAND\Xarrangement@paragraph	184
COMMAND\Xbeforeinserting	50
COMMAND\Xbeforelemmaseparator	47, 384
COMMAND\Xbeforenotes	52, 235, 384, 443, 445
COMMAND\Xbeforenumber	45, 384
COMMAND\Xbeforepagenumber	42, 457
COMMAND\Xbeforesymmlinenumber	45, 384
COMMAND\Xbhookendnote	384

COMMAND\Xbhookgroup	52, 449, 450
COMMAND\Xbhooknote	50, 384–386, 448
COMMAND\Xboxendlinenum	46, 47, 384, 446
COMMAND\Xboxlinenum	46, 47, 384
COMMAND\Xboxlinenumalign	46, 47, 384, 446
COMMAND\Xboxstartlinenum	46, 47, 384, 446
COMMAND\Xboxsymlinenum	46, 384
COMMAND\Xcolalign	50, 445
COMMAND\Xdo@feet	447, 452
COMMAND\Xend	261
COMMAND\XendXafterlemmaseparator	385
COMMAND\XendXbeforelemmaseparator	385
COMMAND\XendXinplaceoflemmaseparator	385
COMMAND\XendXlemmaseparator	385
COMMAND\XendXmorethantwolines	385
COMMAND\XendXtwolines	385
COMMAND\Xendafterenumber	45, 448
COMMAND\Xendafterlemmaseparator	47, 385
COMMAND\Xendafternote	53, 450
COMMAND\Xendafternumber	49
COMMAND\Xendafterpagenumber	451
COMMAND\Xendafterpagenumber	42, 63
COMMAND\XendafterpagenumberSerefonlypage	386
COMMAND\Xendaftersymlinenum	45, 49, 448
COMMAND\Xendahookinplaceofnumber	49, 448
COMMAND\Xendahooklinenum	49, 448
COMMAND\Xendbeforelemmaseparator	47, 385
COMMAND\Xendbeforelinenum	49
COMMAND\Xendbeforenumber	45, 448
COMMAND\Xendbeforepagenumber	42, 62, 63
COMMAND\XendbeforepagenumberSerefonlypage	62
COMMAND\Xendbeforesymlinenum	45, 49, 448
COMMAND\Xendbhookinplaceofnumber	49, 448
COMMAND\Xendbhooklinenum	49, 448
COMMAND\Xendbhooknote	50
COMMAND\Xendboxendlinenum	47, 446
COMMAND\Xendboxlinenum	47, 384, 445
COMMAND\Xendboxlinenumalign	47, 384, 446
COMMAND\Xendboxstartlinenum	47, 446
COMMAND\Xendboxsymlinenum	46, 448
COMMAND\Xendfootnote	133
COMMAND\Xendhangindent	49, 448, 450
COMMAND\Xendinplaceoflemmaseparator	29, 47, 385
COMMAND\Xendinplaceofnumber	46, 447
COMMAND\Xendinplaceofpagenumber	42, 452
COMMAND\Xendinsertsep@	242
COMMAND\Xendlemmadisablefontselection	48
COMMAND\Xendlemmafont	48, 449
COMMAND\Xendlemmaseparator	29, 47, 385
COMMAND\Xendletboxendlinenum	384
COMMAND\Xendlineflag	63
COMMAND\Xendlinenumannotationonlyfirst	45, 246, 455
COMMAND\Xendlinenumannotationonlyfirstintwo	45, 246, 455
COMMAND\Xendlinenumannotationposition	25, 63

COMMAND\Xendlineprefixmore	42, 63
COMMAND\Xendlineprefixsingle	42, 63
COMMAND\Xendlinerangeseparator	43, 63, 199, 386, 449
COMMAND\Xendmorethantwolines	29, 43, 63, 385, 445, 447
COMMAND\Xendnoidenticallinenumannotation	26, 63, 246, 455
COMMAND\Xendnolinenum	457
COMMAND\Xendnolinenumberifannotation	45, 457
COMMAND\Xendnonumber	43, 447
COMMAND\Xendnotboxingsubline	47, 457
COMMAND\Xendnote	238, 260, 261, 445
COMMAND\Xendnotefontsize	48, 384
COMMAND\Xendnotenumfont	48, 49, 383
COMMAND\Xendnotes	241
COMMAND\Xendnumberonlyfirstinline	41, 448
COMMAND\Xendnumberonlyfirstintwolines	41, 448
COMMAND\Xendpagenumberonlyfirst	42, 452
COMMAND\Xendpagenumberonlyfirstifsingle	42, 452
COMMAND\Xendpagenumberonlyfirstintwo	42, 452
COMMAND\Xendparagraph	53, 442
COMMAND\Xendsep	53, 455
COMMAND\Xendstorelineinfo	240
COMMAND\Xendsublinesep	45, 63, 199
COMMAND\Xendsymmlinum	41, 448, 455
COMMAND\Xendsymmlinumannotation	45, 455
COMMAND\Xendsympagenum	42, 452
COMMAND\Xendtwolines	29, 43, 63, 385, 445, 447
COMMAND\Xendtwolinesbutnotmore	43, 63, 445, 447
COMMAND\Xendtwolinesonlyinsamepage	43, 63, 445, 447
COMMAND\Xendtxtbeforenotes	51, 454
COMMAND\Xendwrapcontent	49, 450
COMMAND\Xendwraplemma	48, 450
COMMAND\Xendwraplinenumannotation	26, 63
COMMAND\Xfootfmt	385
COMMAND\Xfootgroup	184
COMMAND\Xfootins	182
COMMAND\Xfootnote	60, 66, 133, 380, 440, 443, 445, 449, 451
COMMAND\Xfootstarts	184
COMMAND\Xgroupbyline	51, 172, 213, 272, 388, 454–457
COMMAND\Xgroupbylines	452
COMMAND\Xgroupbylineseparetwolines	51
COMMAND\Xhangindent	49, 448, 455
COMMAND\Xhsize	387, 449, 450
COMMAND\Xhsizethreecol	50, 53, 384
COMMAND\Xhsizetwocol	50, 53, 266, 384
COMMAND\Xinnotemark	66, 388, 456
COMMAND\Xinplaceoflemmaseparator	28, 47, 384
COMMAND\Xinplaceofnumber	28, 46, 384, 445, 446
COMMAND\Xinsertparafootsep	189, 190
COMMAND\Xledsetnormalparstuff	382, 383, 446
COMMAND\Xlemmadisablefontselection	48
COMMAND\Xlemmafnt	48, 449
COMMAND\Xlemmaseparator	47, 199, 268, 270, 274, 384
COMMAND\Xlineflag	62, 387, 451
COMMAND\Xlinenumannotationonlyfirst	44, 45, 209, 455

COMMAND\Xlinenumannotationonlyfirstintwo	44, 45, 209, 455
COMMAND\Xlinenumannotationposition	25, 63, 164
COMMAND\Xlinenumannotationposition@side	164
COMMAND\Xlinenumannotationpositionside	25
COMMAND\Xlinerangeseparator	43, 62, 199, 386, 449
COMMAND\Xmaxhnotes	52, 74, 75, 383, 384, 443, 445
COMMAND\Xmorethantwolines	28, 43, 62, 385, 445
COMMAND\Xnoidenticallinenumannotation	26, 63, 208, 455
COMMAND\Xnoindent	448
COMMAND\Xnolemmaseparator	47, 274, 385
COMMAND\Xnolinenum	204, 457
COMMAND\Xnolinenumifannotation	45, 204, 457
COMMAND\Xnonbreakableaftersymbol	45, 385, 441
COMMAND\Xnonote	41, 457
COMMAND\Xnonumber	43, 385
COMMAND\Xnonumberinfootnote	385
COMMAND\Xnopagenumberifcurrent	43, 211, 457
COMMAND\Xnotboxingsubline	47, 457
COMMAND\Xnotefontsize	48, 384
COMMAND\Xnotefontsize@(<s>)	189, 193, 194
COMMAND\Xnotenumfont	48, 383
COMMAND\Xnoteswidthliketwocolumns	53, 443
COMMAND\Xnumberonlyfirstinlineline	41, 42, 51, 107, 200, 240, 267, 268, 270, 385, 440, 444, 453
COMMAND\Xnumberonlyfirstintwolines	41, 51, 385, 440, 453
COMMAND\Xonlypstart	44, 385, 440, 444
COMMAND\Xpagelinesep	45, 452
COMMAND\Xparafootsep	51, 108, 383, 385, 388, 451, 453, 454
COMMAND\Xparafootsep@series	189
COMMAND\Xparindent	49, 387, 388, 445, 448, 450, 455
COMMAND\Xprenotes	52, 236, 451
COMMAND\Xprenotes@	183, 236, 440
COMMAND\Xprintlines	107
COMMAND\Xpstart	44, 381, 385, 440, 444
COMMAND\Xpstarteverytime	44, 381, 385, 444
COMMAND\Xpstartonlyfirst	44, 456
COMMAND\Xpstartseparator	44, 456
COMMAND\Xragged	51
COMMAND\Xrigidbalance	191, 387, 450
COMMAND\Xsethangindent	198
COMMAND\Xstanza	44, 57
COMMAND\Xstanzaonlyfirst	44, 456, 457
COMMAND\Xstanzaseparator	44
COMMAND\Xstorelineinfo	200
COMMAND\Xsublinesep	22, 45, 63, 199, 386
COMMAND\Xsublinesepside	22, 45
COMMAND\Xsymlinenum	41, 51, 384, 385, 446, 453
COMMAND\Xsymlinenumannotation	44, 45, 455
COMMAND\Xtextbeforenotes	174
COMMAND\Xtoendnotes	30, 261
COMMAND\Xtwolines	28, 43, 63, 208, 209, 266, 385, 445
COMMAND\Xtwolinesappref	266
COMMAND\Xtwolinesbutnotmore	43, 63, 385, 445
COMMAND\Xtwolinesbutnotmoreappref	267
COMMAND\Xtwolinesonlyinsamepage	43, 63, 385, 445

COMMAND\Xtxtbeforenotes	51, 52, 385, 451–453
COMMAND\Xtxtbeforenotesonlyonce	51, 453
COMMAND\Xtxtbeforenumber	42
COMMAND\Xunvxh	187, 385
COMMAND\Xwidth	53, 387, 450
COMMAND\Xwrapcontent	49, 450
COMMAND\Xwraplemma	48, 50, 450
COMMAND\Xwraplinenumannotation	25, 26, 63, 164
COMMAND\Xwraplinenumannotation@ref	164
COMMAND\Xwraplinenumannotation@side	164
COMMAND\Xwraplinenumannotationref	26
COMMAND\Xwraplinenumannotationside	25
COMMAND\&	383
COMMAND\absline@num	104, 105, 160
COMMAND\accent	132
COMMAND\actionlines@list	105, 161
COMMAND\actions@list	105
COMMAND\add@Xgroupbyline	172
COMMAND\add@hooktoggle@specific@to@cs	272
COMMAND\add@inserts	105, 171, 172
COMMAND\add@inserts@next	171, 172
COMMAND\add@msd@	332
COMMAND\add@msdata	332, 333
COMMAND\add@msdata@firstlineofpage	336
COMMAND\add@msddata	332
COMMAND\add@penalties	160, 173, 443
COMMAND\addcontentsline	361
COMMAND\addfootins	380, 383
COMMAND\addfootinsX	380, 383
COMMAND\addtoendnotes	261
COMMAND\advancelabel@refs	284
COMMAND\advanceline	22, 106, 114, 128, 448
COMMAND\affixline@num	165, 168, 170, 304, 438
COMMAND\affixpstart@num	170
COMMAND\afterXrule	384
COMMAND\afterXsymlinenum	384
COMMAND\afterantilabe	58
COMMAND\aftergroup	131, 136
COMMAND\afterlemmaseparator	384
COMMAND\afternote	384
COMMAND\afternoteX	51, 383, 384
COMMAND\afternumberinfootnote	384
COMMAND\afterruleX	52, 442, 445
COMMAND\annot@end@print	209, 246
COMMAND\annot@list	107, 165
COMMAND\annotationref	25, 26, 59
COMMAND\antilabe	15, 58, 456
COMMAND\applabel	62, 137, 285, 287, 445, 451, 457
COMMAND\appref	60, 62–64, 291, 295, 449, 450, 455
COMMAND\apprefprefixmore	62, 383
COMMAND\apprefprefixsingle	62, 383
COMMAND\apprefwithpage	62, 63, 291, 295, 447, 449, 456
COMMAND\arrangementX	40, 75, 218, 265, 384
COMMAND\arrangementX@normal	223

COMMAND\article	15
COMMAND\at@every@pend	152
COMMAND\autopar	17, 18, 148, 153, 154, 233, 439, 441, 442, 445, 454
COMMAND\ballast	75
COMMAND\ballast@count	160, 173
COMMAND\baselineskip	41, 185, 189
COMMAND\beforeXnotes	384
COMMAND\beforeXsymlinenum	384
COMMAND\beforeantilabe	58
COMMAND\beforeedchapter	10, 72, 362
COMMAND\beforeinsertingX	50
COMMAND\beforelemmaseparator	384
COMMAND\beforenotesX	52, 386, 441, 442, 445
COMMAND\beforenumberinfootnote	384
COMMAND\begin	340
COMMAND\beginnumbering	16, 17, 19, 90–92, 95, 96, 104, 108, 123, 124, 153, 238, 331, 439, 442, 446, 447, 452
COMMAND\beginnumberingR	90
COMMAND\bf	439
COMMAND\bfseries	48, 439
COMMAND\bhookXnote	384
COMMAND\bhookgroupX	52, 449
COMMAND\bhooknoteX	50, 385, 386, 448
COMMAND\body	323
COMMAND\bodyfootmarkA	38
COMMAND\book	15
COMMAND\boxXendlinenum	384
COMMAND\boxXendlinenumalign	384
COMMAND\boxXendstartlinenum	384
COMMAND\boxendlinenum	384
COMMAND\boxlinefootnote	202
COMMAND\boxlinenum	384
COMMAND\boxlinenumalign	384
COMMAND\boxstartlinenum	384
COMMAND\boxsymlinenum	384
COMMAND\break	41, 187
COMMAND\brokenpenalty	173
COMMAND\c@page	125
COMMAND\c@page@page	125
COMMAND\centering	50
COMMAND\ch@ck@l@ck	438
COMMAND\ch@cksub@l@ck	168, 438
COMMAND\chapter	71, 364, 367, 443, 445, 448, 451
COMMAND\chaptermark	362
COMMAND\check@pb@in@verse	374
COMMAND\colalignX	50, 445
COMMAND\collect@body	341
COMMAND\color	450
COMMAND\colorbox	75
COMMAND\columns	53
COMMAND\columnwidth	185, 443
COMMAND\command names	266, 267
COMMAND\continuousnumberingwithcolumns@sync@linenum@singletext	162
COMMAND\copyright	132

COMMAND\correct@Xfootins@box	445
COMMAND\correct@footinsX@box	445
COMMAND\count	192
COMMAND\critex	439
COMMAND\critext	140, 379, 380, 383
COMMAND\csname	80
COMMAND\csquotes	257
COMMAND\ctab	355, 356, 360
COMMAND\ctabtext	360
COMMAND\current@annot	165
COMMAND\dcoll	349
COMMAND\def	77, 93
COMMAND\dimen	192
COMMAND\dimexpr	53
COMMAND\discretionary	186
COMMAND\displaywidowpenalty	173
COMMAND\do@Xfeet	275, 438, 447, 452
COMMAND\do@actions	159, 160, 162, 438
COMMAND\do@actions@fixedcode	438
COMMAND\do@actions@next	160, 161
COMMAND\do@ballast	159, 160, 173
COMMAND\do@feet@custom@order	275
COMMAND\do@insidelinehook	441
COMMAND\do@line	105, 131, 150, 154, 157, 171, 173, 322, 438, 439, 441, 442
COMMAND\do@linehook	438
COMMAND\do@lockoff	106
COMMAND\do@lockon	106
COMMAND\dodoreintrafeet	437
COMMAND\doendnotes	29, 242, 385, 446, 453
COMMAND\doendnotesbysection	29, 242, 261, 446, 453
COMMAND\doennotes	452
COMMAND\doinsidelinehook	26, 443
COMMAND\doinsidethislinehook	27, 158, 455
COMMAND\dolinehook	26, 443
COMMAND\doreintrafeeti	447
COMMAND\doreintrafeetii	447
COMMAND\doxtrafeet	437
COMMAND\doxtrafeeti	447
COMMAND\doxtrafeetii	447
COMMAND\dummy@ref	132
COMMAND\edaftertab	70, 355, 356
COMMAND\edannotationref	289
COMMAND\edatleft	70, 353
COMMAND\edatright	70, 71, 354
COMMAND\edbforetab	70, 355
COMMAND\edfilldimen	354
COMMAND\edfont@info	139
COMMAND\edgls	68, 311, 457
COMMAND\edgls...	450
COMMAND\edglsadd	452
COMMAND\edindex	36, 66–68, 311, 312, 315, 317, 345, 441, 443, 444, 447, 448, 451–453, 456
COMMAND\edindexlab	67
COMMAND\edlabel	59–62, 64, 132, 282, 284, 285, 288, 289, 297, 311, 344, 375, 387, 437, 439, 440, 442, 444, 449, 454, 457



COMMAND\edlabelE	61, 287
COMMAND\edlabelS	61, 287
COMMAND\edlabelSE	61
COMMAND\edlineref	59, 282, 385, 444, 446, 449, 452
COMMAND\edmakelabel	61, 297
COMMAND\edpageref	59, 282, 288, 297
COMMAND\edrowfill	355
COMMAND\edsublineref	59
COMMAND\edtabcolsep	349
COMMAND\edtext	7, 24, 27, 28, 30–33, 35, 54, 59, 60, 62, 65, 66, 68, 75, 94, 105, 107, 120, 122, 126, 127, 130–140, 142, 143, 145, 146, 163–165, 285–287, 290, 344, 346, 364, 375, 379, 380, 383, 437, 439, 440, 443–446, 451–454, 456, 457
COMMAND\edtext@level	446
COMMAND\edtext@notes	133, 134
COMMAND\edtext@notes@start	133, 134
COMMAND\edtextlater	122
COMMAND\edvertdots	71, 354
COMMAND\edvertline	71, 354
COMMAND\elechapter	72
COMMAND\eled@sectioning@out	368
COMMAND\eledchapter	71, 383, 443, 448
COMMAND\eledchapter*	72
COMMAND\eledmac@error	437
COMMAND\eledsection	7, 16, 72, 132, 156, 362, 383, 445, 453, 454, 456, 457
COMMAND\eledsection*	72
COMMAND\eledsubsection	72, 383
COMMAND\eledsubsection*	72
COMMAND\eledsubsubsection	72, 383
COMMAND\eledsubsubsection*	72
COMMAND\eledxxx	10, 72, 368, 442, 443
COMMAND\eledxxxx	362
COMMAND\esection	368
COMMAND\else	310, 362
COMMAND\emph	35
COMMAND\empty	89, 166, 167, 282
COMMAND\end	340
COMMAND\end@lemmas	131
COMMAND\endashchar	199
COMMAND\endgraf	150, 189, 233
COMMAND\endlock	22, 106, 129, 327
COMMAND\endminipage	306, 308, 437, 447
COMMAND\endmsdata	37
COMMAND\endnotes	446, 449, 450
COMMAND\endnumbering	16, 17, 19, 91, 93, 94, 124, 438, 446, 452
COMMAND\endnumberingR	90
COMMAND\endprint	238, 241, 261, 382
COMMAND\endstanzaextra	383
COMMAND\endsub	21, 106, 128
COMMAND\endverse	386
COMMAND\everypar	153
COMMAND\extensionchars	73, 91
COMMAND\externaldocument	64, 297
COMMAND\f@x@l@cks	438
COMMAND>falseverse	383, 441, 443

COMMAND\fi	362
COMMAND\firstlinenum	20, 166, 439
COMMAND\firstsublinenum	20, 439
COMMAND\flag@end	126, 127, 138, 443
COMMAND\flag@end@RTL	127
COMMAND\flag@end@later	127
COMMAND\flag@start	126, 127, 138, 443
COMMAND\flag@start@RTL	127
COMMAND\flag@start@later	127
COMMAND\flagstanza	58
COMMAND\floatingpenalty	179, 445
COMMAND\flush@notes	174
COMMAND\fnpos	39, 235, 442, 452
COMMAND\footfmt	178, 180
COMMAND\footfmt...	218
COMMAND\footfootmarkA	38
COMMAND\footfudgefactor	187
COMMAND\footfudgefiddle	75, 185, 437
COMMAND\footgroup	178
COMMAND\footins	183
COMMAND\footnormal	266, 384, 438
COMMAND\footnormalX	384
COMMAND\footnote	37, 74, 214–216, 380, 438
COMMAND\footnote@lang	199
COMMAND\footnoteA	16, 37
COMMAND\footnoteB	16
COMMAND\footnoteC	27
COMMAND\footnoteE	37
COMMAND\footnoteX	8, 38, 257, 259, 451
COMMAND\footnoteX@reading	259
COMMAND\footnoteX@series	259
COMMAND\footnoteXmark	259, 453, 456, 457
COMMAND\footnoteXmk	274
COMMAND\footnoteXtext	259, 453, 456, 457
COMMAND\footnote(X)	132
COMMAND\footnote(X)mark	38
COMMAND\footnote(X)mk	38
COMMAND\footnote(X)nomk	38
COMMAND\footnote(X)text	38
COMMAND\footnotelang@lua	177
COMMAND\footnotelang@poly	177
COMMAND\footnotemark	38, 259
COMMAND\footnoteoption@	176, 449
COMMAND\footnoterule	192
COMMAND\footnotesize	48
COMMAND\footnotetext	38, 259
COMMAND\footparagraph	185, 266, 384, 443
COMMAND\footparagraphX	228, 384, 442
COMMAND\footsplitskips	438, 445
COMMAND\footstart	178, 182, 192
COMMAND\footstrut	189
COMMAND\footthreecol	384
COMMAND\footthreecolX	384, 445
COMMAND\foottwocol	384

COMMAND\footwocolX	384, 445
COMMAND\foreignlanguage	49
COMMAND\fullstop	386
COMMAND\get@edindex@hyperref	316
COMMAND\get@edindex@ledinnote@command	312
COMMAND\get@fnmark	216
COMMAND\get@index@command	441
COMMAND\get@linelistfile	438
COMMAND\get@thisfootnote	222
COMMAND\getline@num	159, 161
COMMAND\gl@p	97
COMMAND\global	110
COMMAND\globaldefs	110
COMMAND\gls	68, 319
COMMAND\hangindentX	49, 445, 448, 455
COMMAND\hangingsymbol	383, 439
COMMAND\hbox	186, 187
COMMAND\hfill	441
COMMAND\hidenumering	23, 119, 388, 445, 455, 456
COMMAND\hidenumeringonleftpage	23, 119, 451
COMMAND\hidenumeringonrightpage	23, 119, 451
COMMAND\hline	68
COMMAND\hooktoggle@specific	272
COMMAND\hrulefill	355
COMMAND\hsizex	41, 183, 185–187, 193, 195, 234, 438, 443
COMMAND\hsizexX	387, 449, 450
COMMAND\hsizethreecol	384
COMMAND\hsizethreecolX	50, 53
COMMAND\hsizetwocol	384
COMMAND\hsizetwocolX	50, 53
COMMAND\hyperlinkR	316
COMMAND\hyperlinkformat	316
COMMAND\hyperlinkformatR	316
COMMAND\if@RTL	81, 136
COMMAND\if@edtext@	444, 446
COMMAND\if@eled@sectioning	368
COMMAND\if@firstlineofpage	81
COMMAND\if@firstlineofpageR	81
COMMAND\if@msd@options@fullpage	338
COMMAND\if@msdata@insertedfrompreviouspage	336
COMMAND\if@nobreak	151
COMMAND\if@noneed@notes	134
COMMAND\ifXnote@	90
COMMAND\ifafterendnumberingR	90
COMMAND\ifautopar	153
COMMAND\ifby@autopar	153
COMMAND\ifbypage@	97
COMMAND\ifbypage@R	97
COMMAND\ifbypstart@	97
COMMAND\ifbypstart@R	97
COMMAND\iffirst@linenum@out@	123, 124
COMMAND\ifindtl@innote	91
COMMAND\ifindtl@notenumber	91
COMMAND\ifinserthangingsymbol	322

COMMAND\ifinstanza	322
COMMAND\ifistwofollowinglines	209
COMMAND\ifl@d@Xlinenumber	204
COMMAND\ifl@d@Xmorethantwolines	204, 445
COMMAND\ifl@d@Xtwolines	204
COMMAND\ifl@d@dash	204
COMMAND\ifl@d@elin	204
COMMAND\ifl@d@esl	204
COMMAND\ifl@d@pnum	204
COMMAND\ifl@d@ssub	204
COMMAND\ifl@dend@X	260
COMMAND\ifl@dmemoir	437
COMMAND\ifl@dpaging	443
COMMAND\ifl@dpairing	90, 439
COMMAND\ifl@dprintingpages	444
COMMAND\ifl@dskipnumber	166
COMMAND\ifl@dstartendok	354
COMMAND\ifl@imakeidx	81
COMMAND\ifl@noexpands@edgl	321
COMMAND\ifledRcol	90, 439
COMMAND\ifledRcol@	90, 442
COMMAND\iflemmacommand@	444
COMMAND\iflinenumannotationotherside	164
COMMAND\ifnoend@	243
COMMAND\ifnoledgroup@	310
COMMAND\ifnoteschanged@	107
COMMAND\ifnumberedpar@	148
COMMAND\ifnumbering	91, 94
COMMAND\ifnumberingR	90, 439
COMMAND\ifnumberline	138, 166
COMMAND\ifpst@rted	439
COMMAND\ifpst@rtedL	92
COMMAND\ifresumenummering@start	95
COMMAND\ifseriesbefore	264
COMMAND\ifstopmsdata@inserted@	331
COMMAND\ifsublines@	104, 117
COMMAND\ifsublines@R	104
COMMAND\iftrue	446
COMMAND\ifvmode	284
COMMAND\ifxxx	362
COMMAND\ignorespaces	135
COMMAND\imki@wrindexentry	81
COMMAND\immediate	123, 124, 238
COMMAND\indent	18, 153, 386
COMMAND\index	318, 452
COMMAND\indtl@wrindexentry	81
COMMAND\initnumbering@quote	360, 447
COMMAND\initnumbering@reg	438
COMMAND\initnumbering@sectcmd	447
COMMAND\innotemarkX	66, 388, 456
COMMAND\inplaceoflemmaseparator	384
COMMAND\inplaceofnumber	384
COMMAND\insert	138, 171, 178, 180, 181, 194, 213, 218
COMMAND\insert@Xtxtbeforenotes	175, 194

COMMAND\insert@count	120, 126, 135
COMMAND\insert@countR	135
COMMAND\insert@msdata	332, 338
COMMAND\insert@txtbeforenotesX	175
COMMAND\inserthangingsymbol	441
COMMAND\insertlines@list	105, 120, 121
COMMAND\insertparafootsepX	231
COMMAND\inserts@list	131, 148, 171, 186
COMMAND\interAfootnotelinepenalty	438
COMMAND\interfootnotelinepenalty	438
COMMAND\interlinepenalty	178
COMMAND\interparanoteglue	383
COMMAND\justifying	50
COMMAND\keep@this@crossref@forinserting	214
COMMAND\l@advance@parledgroup@beforenormalnotes	447
COMMAND\l@d@@wrindexhyp	444
COMMAND\l@d@add	141
COMMAND\l@d@end	238, 260
COMMAND\l@d@nums	135, 138, 140, 141, 204, 205
COMMAND\l@d@section	238
COMMAND\l@d@set	116, 129
COMMAND\l@dampcount	346
COMMAND\l@dbfnote	216, 438
COMMAND\l@dcheckstartend	354
COMMAND\l@dchset@num	116
COMMAND\l@dcolcount	346, 347
COMMAND\l@dcollect@@body	340
COMMAND\l@dcollect@body	340
COMMAND\l@dcsnote	442
COMMAND\l@dcsnotetext	158, 302
COMMAND\l@dcsnotetext@l	158, 302
COMMAND\l@dcsnotetext@r	158, 302
COMMAND\l@ddodorextrafeet	276, 437
COMMAND\l@ddoxtrafeet	274, 276, 437
COMMAND\l@emptyd@ta	438
COMMAND\l@dend@close	238
COMMAND\l@dend@open	238
COMMAND\l@dend@stuff	238
COMMAND\l@denvbody	340
COMMAND\l@dfeetbeginmini	437
COMMAND\l@dfeetendmini	437
COMMAND\l@dgetline@margin	439
COMMAND\l@dgetlock@disp	439
COMMAND\l@dgetref@num	289, 290
COMMAND\l@dgetsidenote@margin	298, 439
COMMAND\l@dgobbleoptarg	443
COMMAND\l@dgobblearg	443
COMMAND\l@dgobbleoptarg	345
COMMAND\l@dlabel@parse	289, 290
COMMAND\l@dld@ta	165, 167
COMMAND\l@dlp@rbox	304
COMMAND\l@dlsn@te	439
COMMAND\l@dlsnote	442
COMMAND\l@dmake@labels	284, 298

COMMAND\l@dmake@labelsR	298
COMMAND\l@dnumpstartsL	92, 439
COMMAND\l@dp@rsefootspec	205
COMMAND\l@dparsefootspec	205
COMMAND\l@dpush@begins	340
COMMAND\l@drd@ta	165, 167
COMMAND\l@dref@undefined	289
COMMAND\l@drsn@te	439
COMMAND\l@drsnote	442
COMMAND\l@dtabaddcols	354
COMMAND\l@dtabnoexpands	437
COMMAND\l@dumboxmpfoot	447
COMMAND\l@dunboxmpfoot	438
COMMAND\l@dunhbox@line	154
COMMAND\l@dzeropenalties	439, 443
COMMAND\l@noexpands@edgl	321
COMMAND\l@pb	374
COMMAND\l@prev@nopb	373
COMMAND\l@prev@pb	373
COMMAND\l@reg	381
COMMAND\l@secondoptarg	345
COMMAND\l@wrap@ifnotemptybox	89
COMMAND\label	19, 61, 64, 67, 282, 283, 290
COMMAND\label@refs	282
COMMAND\labelstarttrue	19, 387, 440, 454
COMMAND\labelref@list	282, 285
COMMAND\language	186
COMMAND\lastbox	153
COMMAND\lastskip	128
COMMAND\latex@makecol	277
COMMAND\leavevmode	18, 153
COMMAND\led@check@nopb	374
COMMAND\led@check@pb	374
COMMAND\led@nopb	373, 374
COMMAND\led@nopbnum	373
COMMAND\led@pb	373, 374
COMMAND\led@pb@macro	373
COMMAND\led@pbnum	373
COMMAND\led@reinit@index@fornote	318
COMMAND\led@set@index@fornote	318
COMMAND\ledRflag	316
COMMAND\ledchapter	383, 440
COMMAND\ledfootinsdim	383
COMMAND\ledinnernote	64, 300, 442, 443, 451
COMMAND\ledinnote	313, 382, 388, 446
COMMAND\ledinnotemark	382, 388, 446, 456
COMMAND\ledleftnote	64, 300
COMMAND\ledlinenum	102, 439
COMMAND\ledllfill	159
COMMAND\ledlsnotefontsetup	450
COMMAND\ledlsnotesep	65
COMMAND\ledlsnotewidth	65
COMMAND\lednopb	73, 373
COMMAND\lednopbinverse	374

COMMAND\lednopbinversettrue	56, 73
COMMAND\lednopbnum	373
COMMAND\ledouternote	64, 300, 442, 443, 451
COMMAND\ledpb	73, 373, 383
COMMAND\ledpbnum	373
COMMAND\ledpbsetting	73, 373, 448
COMMAND\ledrightnote	64, 300
COMMAND\ledrsnotefontsetup	450
COMMAND\ledrsnotesep	65
COMMAND\ledrsnotewidth	65
COMMAND\ledsection	383
COMMAND\ledsectnomark	362
COMMAND\ledsectnotoc	361
COMMAND\ledsetnormalparstuff	382, 383, 446
COMMAND\ledsetnormalparstuff@common	233
COMMAND\ledsetnormalparstuffX	382, 383, 446
COMMAND\ledsidenote	64, 300, 302
COMMAND\ledsidenote@	300
COMMAND\ledsubsection	383
COMMAND\ledsubsubsection	383
COMMAND\ledxxx	442
COMMAND\left	70
COMMAND\leftctab	355
COMMAND\leftheadline	102
COMMAND\leftlinenum	21, 102, 437, 439
COMMAND\leftlinenumannotation	103
COMMAND\lefttab	355
COMMAND\leftnoteupfalse	65
COMMAND\leftpstartnum	170
COMMAND\leftftab	355
COMMAND\leftsidenote	302
COMMAND\leftskip	183, 186, 187
COMMAND\lemma	3, 28, 30–33, 35, 131, 134–136, 139, 140, 142, 379, 439, 440, 447, 449, 457
COMMAND\lemmaseparator	384
COMMAND\let	31, 54, 132, 138, 259, 277, 327, 437
COMMAND\letboxXendendlinenum	384
COMMAND\line	191, 194
COMMAND\line@list	105, 107, 121, 138
COMMAND\line@list@stuff	92, 108, 123, 437, 439
COMMAND\line@list@version	110
COMMAND\line@margin	99, 167, 298
COMMAND\line@num	104–106, 166, 437
COMMAND\line@set	140, 141
COMMAND\lineannot	25, 60, 141
COMMAND\lineannot@set	141
COMMAND\lineation	21, 98, 454, 456
COMMAND\linebreak	41
COMMAND\lineneation	450
COMMAND\linenum	25, 28, 30, 31, 60, 62, 131, 140, 141, 288, 290, 297, 379
COMMAND\linenum@out	123, 282, 285
COMMAND\linenumannotation	24, 163, 454, 455
COMMAND\linenumannotationothersidefalse	25
COMMAND\linenumannotationothersidetrue	25, 457
COMMAND\linenumberlist	20, 21, 89, 166, 167, 437

COMMAND\linenumberstyle	22, 102, 437, 452
COMMAND\linenumincrement	20, 439
COMMAND\linenummargin	21, 99, 298
COMMAND\linenumr@p	102, 437, 439
COMMAND\linenumrep	102, 439
COMMAND\linenumsep	21, 65, 102, 299
COMMAND\linerangesep@	274
COMMAND\lineref	282, 288, 297, 385, 444
COMMAND\list@clear	96
COMMAND\list@clearing@reg	439
COMMAND\list@create	96
COMMAND\lock@disp	101
COMMAND\lock@off	118
COMMAND\lock@on	117
COMMAND\lockdisp	22, 101
COMMAND\loop	174, 323
COMMAND\ltab	355, 356, 360
COMMAND\ltabtext	360
COMMAND\m@mmf@prepare	215
COMMAND\mainmatter	454, 455
COMMAND\makeatletter	157
COMMAND\makehboxofhboxes	188, 189
COMMAND\makeindex	66, 315
COMMAND\makelabel	297
COMMAND\manage stanza@modulo	324
COMMAND\marginpar	64, 74, 298, 438
COMMAND\marginparwidth	65, 299
COMMAND\markboth	26, 27, 157
COMMAND\mathchardef	323
COMMAND\maxhXnotes	384
COMMAND\maxhnotesX	52, 75, 383, 441, 442, 445
COMMAND\maxlinesinpar@list	108
COMMAND\measurebody	357
COMMAND\measuredbody	357
COMMAND\memorybreak	20
COMMAND\metaXXXvfootnote	213
COMMAND\microtype	154
COMMAND\morenoexpands	75, 76, 131, 132
COMMAND\morethantwolines	385
COMMAND\mpfnpos	39, 235, 442, 452
COMMAND\mpnormalfootgroup	437
COMMAND\mpnormalvfootnote	437
COMMAND\msdata	36, 37, 330–332, 452, 453, 455
COMMAND\msdataposition	37
COMMAND\multfootsep	38, 214
COMMAND\multiplefootnotemarker	214
COMMAND\musixtex	442
COMMAND\n@num	439, 446
COMMAND\n@num@ref	446
COMMAND\new@line	125, 438
COMMAND\new@series	132
COMMAND\newcommand	31, 77, 94, 214, 284
COMMAND\newcommandx	31
COMMAND\newhookarg@specific	272



COMMAND\newhookcommand@series	266, 267, 446, 456
COMMAND\newhookcommand@series@reload	267
COMMAND\newhookcommand@toggle@reload	267, 443
COMMAND\newhooktoggle@series	267, 446
COMMAND\newhooktoggle@specific	272
COMMAND\newif	445
COMMAND\newline	41
COMMAND\newlinechar	260
COMMAND\newseries	39, 380, 383
COMMAND\newseries@	251, 264
COMMAND\newverse	57, 58, 383, 443, 453
COMMAND\next	323
COMMAND\next@action	109
COMMAND\next@actionline	109
COMMAND\next@insert	171
COMMAND\next@line@list@stuff	124, 454
COMMAND\nl@regR	111
COMMAND\no@expands	75, 132, 139, 259, 321, 437
COMMAND\noXlemmaseparator	385
COMMAND\nobreak	204
COMMAND\nocritical	251
COMMAND\noeledsec	383
COMMAND\noendnotes	383
COMMAND\noexpand	380
COMMAND\nofamiliar	270
COMMAND\noindent	18, 56, 57, 153, 448
COMMAND\noindentX	448
COMMAND\nomk@	274
COMMAND\nonbreakableafternumber	385
COMMAND\nonoteX	41, 457
COMMAND\nonumberinfootnote	385
COMMAND\norelax	54
COMMAND\normal@footnotemarkX	218
COMMAND\normal@page@break	373
COMMAND\normal@pars	233
COMMAND\normalbfnoteX	439
COMMAND\normalbodyfootmarkX	218
COMMAND\normalfont	452
COMMAND\normalfootfmt	54, 182, 189, 199, 238, 451
COMMAND\normalfootfmtX	219
COMMAND\normalfootfootmarkX	219
COMMAND\normalfootgroup	183
COMMAND\normalfootgroupX	220
COMMAND\normalfootnoterule	179
COMMAND\normalfootstart	182, 186
COMMAND\normalfootstartX	220
COMMAND\normalvfootnote	180, 182
COMMAND\normalvfootnote@inserted	181, 182
COMMAND\normalvfootnoteX	218
COMMAND\notbool	362
COMMAND\notfontsetup	384
COMMAND\notfontsizeX	48, 384
COMMAND\notenumfont	383
COMMAND\notenumfontX	48, 383

COMMAND\notesXwidthliketwocolumns	385
COMMAND\noteswidthliketwocolumnsX	53, 385, 443, 445
COMMAND\num@lines	148, 173
COMMAND\numberlinefalse	20, 455
COMMAND\numberlinetrue	20, 455
COMMAND\numberonlyfirstinline	264, 385
COMMAND\numberonlyfirstintwolines	385
COMMAND\numberpstartfalse	19
COMMAND\numberpstarttrue	19, 44, 381, 439, 447, 452, 454
COMMAND\numberstanza	44
COMMAND\numberstanzafalse	57
COMMAND\numberstanzatru	57
COMMAND\numlabfont	21, 53, 102
COMMAND\one@line	148
COMMAND\onehalfspacing	448
COMMAND\onlyXpstart	385
COMMAND\onlysideX	258
COMMAND\page@action	106, 116
COMMAND\page@start	106, 439
COMMAND\pagecontents	106
COMMAND\pagenumbering	280, 458
COMMAND\pageparbreak	383
COMMAND\pageref	61, 288
COMMAND\par	30, 41, 153, 233
COMMAND\par@line	148, 173
COMMAND\par@patch@pagenumbering	280
COMMAND\par@patch@thepage	125, 280
COMMAND\para@footgroup	186
COMMAND\para@footgroupX	230
COMMAND\para@footsetup	185, 437
COMMAND\para@footsetupX	228, 437, 443
COMMAND\para@vfootnoteX	229
COMMAND\parafootfmt	188, 189, 451
COMMAND\parafootfmtX	230
COMMAND\parafootftm	190
COMMAND\parafootftmX	231
COMMAND\parafootftmsep	383
COMMAND\parafootsep	385, 441, 446
COMMAND\parafootsepX	51, 108, 383, 385, 388, 451, 456, 457
COMMAND\parafootstart	186
COMMAND\parafootstartX	229
COMMAND\paravfootnote	186, 189
COMMAND\parfillskip	188
COMMAND\parindent	448
COMMAND\parindentX	49, 387, 448, 450, 455
COMMAND\parse@annot	165
COMMAND\parshape	75
COMMAND\parskip	18, 57, 153
COMMAND\pause@numbering	94
COMMAND\pausenumbering	19, 94, 95, 109, 110, 154, 442, 443, 450
COMMAND\pausenumbering@page@num	94
COMMAND\penalty	188
COMMAND\pend	2,
	7, 17, 18, 20–22, 72, 129, 131, 134, 141, 148, 150–154, 170, 171, 329, 381, 441–443, 451–454

- COMMAND

```
Xnotes
```

 . . . . . 445, 451  
 COMMAND

```
notesX
```

 . . . . . 52, 237, 445  
 COMMAND

```
prepare@Xgroupbyline
```

 . . . . . 213, 214  
 COMMAND

```
prepare@Xprenotes
```

 . . . . . 235  
 COMMAND

```
prev@nopb
```

 . . . . . 373  
 COMMAND

```
prev@pb
```

 . . . . . 373  
 COMMAND

```
prevlineX
```

 . . . . . 107  
 COMMAND

```
prevpageX@num
```

 . . . . . 108  
 COMMAND

```
print@Xfootnoterule
```

 . . . . . 446  
 COMMAND

```
print@Xnotes
```

 . . . . . 275, 276  
 COMMAND

```
print@Xnotes@forpages
```

 . . . . . 445  
 COMMAND

```
print@eledsection
```

 . . . . . 156  
 COMMAND

```
print@footnoteXrule
```

 . . . . . 446  
 COMMAND

```
print@leftmargin@eledsection
```

 . . . . . 363  
 COMMAND

```
print@lemma
```

 . . . . . 200  
 COMMAND

```
print@line
```

 . . . . . 155  
 COMMAND

```
print@notesX@forpages
```

 . . . . . 445  
 COMMAND

```
print@rightmargin@eledsection
```

 . . . . . 363  
 COMMAND

```
printendlines
```

 . . . . . 243, 250, 295, 386, 437, 438  
 COMMAND

```
printlinefootnote
```

 . . . . . 201, 202, 444  
 COMMAND

```
printlinefootnotearea
```

 . . . . . 202, 204, 444  
 COMMAND

```
printlinefootnotenumbers
```

 . . . . . 201  
 COMMAND

```
printlines
```

 . . . . . 182, 199, 204, 205, 243, 295, 386, 437, 439, 445, 449  
 COMMAND

```
printnpnum
```

 . . . . . 382, 386  
 COMMAND

```
printpstart
```

 . . . . . 199  
 COMMAND

```
protect
```

 . . . . . 132, 380  
 COMMAND

```
providecommand
```

 . . . . . 214, 437  
 COMMAND

```
pstarL
```

 . . . . . 148  
 COMMAND

```
pstart
```

 . . . . . 2, 7, 17–22, 71, 72, 116,  
     129, 134, 138, 141, 148, 151–153, 156, 171, 329, 367, 381, 439–443, 445–448, 450–454, 456  
 COMMAND

```
pstartinfootnote
```

 . . . . . 385  
 COMMAND

```
pstartinfootnoteeverytime
```

 . . . . . 385  
 COMMAND

```
pstartnum
```

 . . . . . 170  
 COMMAND

```
pstartref
```

 . . . . . 59, 282, 289, 442, 453  
 COMMAND

```
pstarts
```

 . . . . . 439  
 COMMAND

```
pstarts@readL
```

 . . . . . 148  
 COMMAND

```
pstarts@typeset@L
```

 . . . . . 148  
 COMMAND

```
raggedX
```

 . . . . . 51  
 COMMAND

```
raggedleft
```

 . . . . . 50  
 COMMAND

```
raggedright
```

 . . . . . 50  
 COMMAND

```
raw@text
```

 . . . . . 148  
 COMMAND

```
rbracket
```

 . . . . . 47  
 COMMAND

```
read@linelist
```

 . . . . . 108–110  
 COMMAND

```
ref
```

 . . . . . 61, 64, 67  
 COMMAND

```
refformated@
```

 . . . . . 295  
 COMMAND

```
refformatedwithpage
```

 . . . . . 295  
 COMMAND

```
relax
```

 . . . . . 18, 116, 138, 160, 171, 327, 345, 381  
 COMMAND

```
renewcommand
```

 . . . . . 75, 383, 384, 386  
 COMMAND

```
reset@current@annot
```

 . . . . . 165  
 COMMAND

```
reset@msd@options@iffullpage
```

 . . . . . 338  
 COMMAND

```
resetlinenumannotation
```

 . . . . . 24, 165  
 COMMAND

```
resetprevline@
```

 . . . . . 107  
 COMMAND

```
resetprevpage@
```

 . . . . . 108  
 COMMAND

```
resumenumbering
```

 . . . . . 19, 90, 91, 95, 96, 109, 110, 154, 439, 442, 443, 450

COMMAND\right	70
COMMAND\rightctab	356
COMMAND\rightlinenum	21, 102, 437, 439
COMMAND\rightlinenumannotation	103
COMMAND\rightltab	356
COMMAND\rightnoteupfalse	65
COMMAND\rightrtab	356
COMMAND\rightsidenote	302
COMMAND\rightskip	183, 186–188
COMMAND\rightstartnum	170
COMMAND\rigidbalance	191, 192, 194, 387, 450
COMMAND\rigidbalanceX	191, 387, 450
COMMAND\robustify	41, 45
COMMAND\roman	349, 449
COMMAND\rtab	355–357, 360
COMMAND\rtabtext	357, 360
COMMAND\sameparallelpagenum	125
COMMAND\sameword	15, 31–33, 35, 122, 141–144, 146, 388, 444, 446–448, 451, 453, 454, 456
COMMAND\sameword@inedtext	143
COMMAND\saweword	142
COMMAND\scriptsize	102
COMMAND\section	71, 439, 452
COMMAND\section@num	91
COMMAND\sectionmark	362
COMMAND\select@lemmfont	53, 54, 176
COMMAND\series	251
COMMAND\series@	251
COMMAND\seriesatbegin	39, 263, 445
COMMAND\seriesatend	39, 264, 446
COMMAND\set@Xtxtbeforenotes	174
COMMAND\set@continuousnumberingforl	96
COMMAND\set@line	138
COMMAND\set@line@action	106, 116
COMMAND\set@txtbeforenotesX	175
COMMAND\setSErefonlypageprefixmore	62, 294, 449, 453
COMMAND\setSErefonlypageprefixsingle	62, 294, 449, 453
COMMAND\setSErefprefixmore	62
COMMAND\setSErefprefixsingle	62
COMMAND\setapprefprefixmore	62, 383
COMMAND\setapprefprefixsingle	62, 64, 383, 449
COMMAND\setcommand@series	265
COMMAND\setcounter	375
COMMAND\sethangindentX	198
COMMAND\sethangingsymbol	56, 322, 383, 448
COMMAND\sethangingsymbol	54
COMMAND\setistwofollowinglines	209
COMMAND\setl@dlprbox	304
COMMAND\setline	22, 106, 111, 115, 129, 132, 448
COMMAND\setlinenum	22, 111, 116, 129, 151, 437, 454
COMMAND\setlinenumannotationsep	26, 388, 455
COMMAND\setmsdatalabel	37
COMMAND\setmsdataposition	453
COMMAND\setmsdataseries	37
COMMAND\setprintendlines	243, 247, 438

COMMAND\setprintlines	205, 209, 243, 439
COMMAND\setsidenotesepp	65
COMMAND\setsidenotsep	384
COMMAND\setstanzaindent	324
COMMAND\setstanzaindents	55, 323, 381, 456
COMMAND\setstanzapenalties	323
COMMAND\setstanzavalues	323
COMMAND\settoggle@series	265, 440, 443
COMMAND\showlemma	132, 437, 438
COMMAND\showwordrank	35, 143
COMMAND\sidenote@margin	438
COMMAND\sidenotemargin	64, 438, 442
COMMAND\sidenotesepp	384
COMMAND\sidepstartnumtrue	19
COMMAND\skip	182, 183
COMMAND\skipnumbering	23, 118, 119, 130, 439, 446
COMMAND\skipnumbering@reg	446
COMMAND\small	48
COMMAND\somemacro	35
COMMAND\special	13
COMMAND\splitmaxdepth	178, 193
COMMAND\splitoff	191
COMMAND\splittopskip	178, 193, 194
COMMAND\stanza	22, 23, 56–58, 91, 327, 384, 386, 448, 453
COMMAND\stanza@hang	326
COMMAND\stanza@line	326
COMMAND\stanzaindent	55, 324, 444
COMMAND\stanzaindent*	55
COMMAND\stanzaindentbase	323
COMMAND\stanzanumwrapper	57
COMMAND\startlock	20, 22, 106, 129, 327
COMMAND\startstanzahook	384
COMMAND\startsub	21, 106, 128
COMMAND\stopmsd	333
COMMAND\stopmsdata	36, 37, 331
COMMAND\store@annot@to@absline	164
COMMAND\strip@pt	186
COMMAND\strutbox	193
COMMAND\sub@action	106, 117
COMMAND\sub@lock	105
COMMAND\sub@off	114, 285
COMMAND\sub@on	114, 285
COMMAND\subline@num	104–106
COMMAND\sublinenum@rep	437
COMMAND\sublinenumberstyle	23, 102, 437
COMMAND\sublinenumincrement	20
COMMAND\sublinenumr@p	102, 437, 439
COMMAND\sublinenumrep	102, 439
COMMAND\sublineref	59, 282, 288
COMMAND\subsectionmark	362
COMMAND\sw@inthisedtext	134
COMMAND\sw@list@inedtext	139, 146
COMMAND\sw@txt	143
COMMAND\swnoexpands	35, 454

COMMAND\symlinenum	385
COMMAND\symplinum	384
COMMAND\sza@penalty	326
COMMAND>tag	444
COMMAND\text	379
COMMAND\text<language>	49
COMMAND\textcolor	76
COMMAND\textheight	75
COMMAND\textsuperscript	25, 89, 164
COMMAND\the	125, 437
COMMAND\thefootnoteA	38
COMMAND\thefootnoteX	441
COMMAND\thelabidx	317
COMMAND\thepage	111, 125, 280
COMMAND\thepstart	19, 387
COMMAND\thepstartL	439
COMMAND\thepstartR	439
COMMAND\thestanza	57
COMMAND\this@crossref@end	214
COMMAND\this@line@list@version	123
COMMAND\this@section@next@page@num	94
COMMAND\this@section@page@num	107
COMMAND\thisfootnote	222
COMMAND\threecol@begin@insert	194
COMMAND\threecolfootfmt	193, 194, 451
COMMAND\threecolfootfmtX	227
COMMAND\threecolfootgroup	192
COMMAND\threecolfootgroupX	227
COMMAND\threecolfootsetup	192
COMMAND\threecolfootsetupX	226
COMMAND\threecolvfootnote	193
COMMAND\threecolvfootnote@inserted	193
COMMAND\threecolvfootnoteX	227
COMMAND\toendnotes	29, 241, 451
COMMAND\twocolfootfmt	451
COMMAND\twocolfootfmtX	225
COMMAND\twocolfootgroupX	225
COMMAND\twocolfootsetupX	224
COMMAND\twocolvfootnoteX	224
COMMAND\twolines	264, 385
COMMAND\twolines@A	264
COMMAND\twolines@B	264
COMMAND\twolines@C	264
COMMAND\twolinesbutnotmore	385
COMMAND\twolinesonlyinsamepage	385
COMMAND\txbeforeXnotes	385
COMMAND\txbeforenotesX	51, 175, 453
COMMAND\txbeforenotesonlyonceX	51, 453
COMMAND\uline	38
COMMAND\unhbox	154, 186
COMMAND\unpenalty	188, 189
COMMAND\unskip	188
COMMAND\unvxh	188, 385
COMMAND\unvxhX	385

COMMAND\upbracefill	355
COMMAND\usingcritext	380, 383
COMMAND\usingedtext	380, 383
COMMAND\vAfootnote	178
COMMAND\vadjust	126
COMMAND\variant	31
COMMAND\vbox	150, 153, 187, 191, 235
COMMAND\vfootnote	178, 183, 186, 193
COMMAND\vl@dbfnote	216, 438
COMMAND\vl@disnote	302
COMMAND\vl@dlsnote	302
COMMAND\vl@dosnote	302
COMMAND\vl@drsnote	302
COMMAND\vnumfootnoteX	439
COMMAND\vsize	52, 75
COMMAND\vsplit	173
COMMAND\waklam	355
COMMAND\waklamec	355
COMMAND\wapunktel	355
COMMAND\wastricht	355
COMMAND\widthX	53, 387, 450
COMMAND\wrap@edcrossref	287, 443
COMMAND\wrapcontentX	49, 450
COMMAND\wrapped@bodyfootmarkX	232
COMMAND\wrapped@footfootmarkX	232
COMMAND\x...	60
COMMAND\xabslineref	288
COMMAND\xannotationref	60
COMMAND\xdef	96, 327
COMMAND\xflagref	60, 289, 386, 449
COMMAND\xleft@appenditem	97, 131
COMMAND\xlineref	60, 386, 449
COMMAND\xpageref	60
COMMAND\xparse	36
COMMAND\xpstartref	60, 442, 454
COMMAND\xr	64
COMMAND\xright@appenditem	96, 97
COMMAND\xsublineref	60
COMMAND\xxref	60, 290, 297, 442, 445, 451
COMMAND\xxxmatter	280
ENVIRONMENTastanza	91, 449
ENVIRONMENTedarrayc	360
ENVIRONMENTedarrayl	360
ENVIRONMENTedarrayr	360
ENVIRONMENTedtabular	456
ENVIRONMENTedtabularc	360
ENVIRONMENTedtabularl	360
ENVIRONMENTedtabularr	360
ENVIRONMENTledgroup	81, 309, 387, 449
ENVIRONMENTledgroupsize	309
PACKAGE(r(e)ledmac	39
PACKAGEEDMAC	1, 6, 10, 12, 13, 77, 204, 214, 282, 323, 379, 390, 437
PACKAGEEledmac	106, 314, 382, 383, 444, 446
PACKAGEEledpar	446

PACKAGEetoolbox	80
PACKAGEParallel	390
PACKAGEReledmac	58, 385–387
PACKAGEamsgen	339
PACKAGEamsmath	339
PACKAGEbabel	49, 76, 349, 449
PACKAGEbiblatex	74
PACKAGEbidi	48, 50, 81, 448, 456
PACKAGEcalc	457
PACKAGEcaption	90
PACKAGEcolor	75
PACKAGEcsquotes	453
PACKAGEedstanza	1, 13, 322
PACKAGEeledmac	1, 10, 14, 15, 67, 214, 310, 314, 342, 364, 376, 380, 382–384, 440, 442, 444
PACKAGEeledpar	179, 390, 439, 442–445
PACKAGEetex	448
PACKAGEetoolbox	96, 141, 251, 264, 274, 302, 362, 373
PACKAGEfancyhdr	277, 278, 451
PACKAGEfloatrow	74, 387
PACKAGEfootmisc	38, 76, 81, 214, 390, 452
PACKAGEgeometry	15
PACKAGEglossaries	68, 319, 449, 457
PACKAGEglossaries-extra	68, 457
PACKAGEhandout	444
PACKAGEhyperref	60, 134, 137, 232, 283, 287, 316, 366, 375, 441–444, 450, 453, 455, 457
PACKAGEifluatex	80
PACKAGEifxetex	80
PACKAGEimakeidx	66, 74, 81, 310, 314, 383, 441, 442, 444
PACKAGEindextols	318
PACKAGEindextool	383
PACKAGEindextools	66, 74, 81, 91, 310, 314, 318, 383, 444, 449, 452
PACKAGEledarab	76
PACKAGEledmac	1, 10, 13, 14, 76, 96, 314, 379, 380, 383, 386
PACKAGEledpar	76
PACKAGEMemoir	81, 314, 383, 390, 443, 447
PACKAGEMorewrites	74
PACKAGEmusixtex	442
PACKAGEperpage	449
PACKAGEpolyglossia	47, 76, 177, 199, 449, 456, 457
PACKAGERagged2e	50, 80
PACKAGEreledmac	1, 6, 10–12, 14–17, 19, 20, 23, 26, 27, 29–33, 36, 38–40, 42–44, 46, 49, 50, 52, 54, 56, 58, 60–62, 64–68, 72–78, 97, 99, 105, 106, 110, 114, 123–125, 132, 136, 138, 141, 171, 179, 183, 187, 199, 214, 238, 251, 256, 264, 274, 279, 287, 288, 291, 297, 314, 336, 339, 342, 362, 363, 374, 375, 382–389, 447, 450, 452
PACKAGEreledpar	1, 4, 6, 8, 16, 18, 20, 38, 53, 60, 62, 63, 73, 74, 76, 78, 90, 97, 108, 114, 124, 125, 135, 138, 148, 180, 183, 233, 234, 251, 258, 274–276, 279, 310, 322, 362, 448, 449, 451–454, 457
PACKAGESuffix	80
PACKAGETabmac	1, 13, 390
PACKAGEthalié	58
PACKAGEulem	38
PACKAGEuninormalize	32
PACKAGEXargs	31, 80
PACKAGEXkeyval	78, 274
PACKAGEXparse	80, 454



PACKAGEXr	5, 64, 297, 298, 450
PACKAGEXref	297
PACKAGEXstring	80, 316

## A

<code>\absline@num</code>	1
Abu Kamil Shuja' b. Aslam	13
<code>\actionlines@list</code>	1
<code>\actions@list</code>	1
<code>\add@hookarg@specific@to@cs</code>	1
<code>\add@hooktoggle@specific@to@cs</code>	1
<code>\add@inserts</code>	1
<code>\add@inserts@next</code>	1
<code>\add@msd@</code>	1
<code>\add@msdata</code>	1
<code>\add@msdata@firstlineofpage</code>	1
<code>\add@penalties</code>	1
<code>\add@Xgroupbyline</code>	1
<code>\addtol@denbody</code>	1
Adelard II	13
<code>\advancelabel@refs</code>	1
<code>\advanceline</code>	1, 22
<code>\Aendnote</code>	28
<code>\affixline@num</code>	1
<code>\affixpstart@num</code>	1
<code>\affixside@note</code>	1
<code>\Afootnote</code>	28
<code>\afterantilabe</code>	58
<code>\afternoteX</code>	51
<code>\afterruleX</code>	52
<code>\ampersand</code>	1, 58
<code>\annot@list</code>	1
<code>\annotationref</code>	1, 59
<code>\antilabe</code>	1, 58
<code>\append@notesX</code>	1
<code>\append@Xnotes</code>	1
<code>\applabel</code>	1, 62
<code>\appref</code>	1, 62
<code>\apprefwithpage</code>	1, 62
<code>\arrangementX</code>	1, 40
<code>\arrangementX@normal</code>	1
<code>\arrangementX@paragraph</code>	1
<code>\arrangementX@threecol</code>	1
<code>\arrangementX@twocol</code>	1
<code>\at@every@pend</code>	1
<code>\AtEndEveryPend</code>	1, 18
<code>\AtEveryPend</code>	1, 18
<code>\AtEveryPstart</code>	1, 18
<code>\AtEveryStanza</code>	1, 57
<code>\AtEveryStopStanza</code>	1, 57
<code>\AtStartEveryPstart</code>	1, 18
<code>\AtStartEveryStanza</code>	1, 57
<code>\autopar</code>	1, 17

**B**

<code>\ballast</code> .....	75
<code>\ballast@count</code> .....	<u>1</u>
Beeton, Barbara Ann Neuhaus Friend .....	18
<code>\beforeantilabe</code> .....	58
<code>\beforeeledchapter</code> .....	<u>1</u>
<code>\BeforeEveryStopStanza</code> .....	<u>1</u> , 57
<code>\beforeinsertingX</code> .....	50
<code>\beforeinsertion@X</code> .....	<u>1</u>
<code>\beforenotesX</code> .....	52
<code>\beginnumbering</code> .....	<u>1</u> , 16
<code>\Bendnote</code> .....	28
<code>\Bfootnote</code> .....	28
<code>\bhookgroupX</code> .....	52
<code>\bhooknoteX</code> .....	50
<code>\bodyfootmarkA</code> .....	38
<code>\boxfootnotenumbers</code> .....	<u>1</u>
Bredon, Simon .....	13
Breger, Herbert .....	13, 342
Brey, Gerhard .....	13
Busard, Hubert L. L. ....	13
<code>\bypage@false</code> .....	<u>1</u>
<code>\bypage@true</code> .....	<u>1</u>
<code>\bypstart@false</code> .....	<u>1</u>
<code>\bypstart@true</code> .....	<u>1</u>

**C**

<code>\c@addcolcount</code> .....	<u>1</u>
<code>\c@ballast</code> .....	<u>1</u>
<code>\c@firstlinenum</code> .....	<u>1</u>
<code>\c@firstsublinenum</code> .....	<u>1</u>
<code>\c@labidx</code> .....	<u>1</u>
<code>\c@linenumincrement</code> .....	<u>1</u>
<code>\c@sublinenumincrement</code> .....	<u>1</u>
<code>\Cendnote</code> .....	28
<code>\Cfootnote</code> .....	28
<code>\ch@ck@l@ck</code> .....	<u>1</u>
<code>\ch@cksub@l@ck</code> .....	<u>1</u>
<code>\chapter</code> .....	<u>1</u>
<code>\check@pb@in@verse</code> .....	<u>1</u>
Chester, Robert of .....	13
Claassens, Geert H. M. ....	13
<code>\colalignX</code> .....	50
<code>\continuousnumberingwithcolumns@sync@linenumber@singletext</code> .....	<u>1</u>
Copernicus, Nicolaus .....	13
<code>\critext</code> .....	379
<code>\ctab</code> .....	<u>1</u>
<code>\ctabtext</code> .....	<u>1</u>
<code>\current@annot</code> .....	<u>1</u>

**D**

Dekker, Dirk-Jan .....	76
<code>\Dendnote</code> .....	28
<code>\Dfootnote</code> .....	28

<code>\disable@edlabel</code> .....	<u>1</u>
<code>\disable@endnotes</code> .....	<u>1</u>
<code>\disable@familiarnotes</code> .....	<u>1</u>
<code>\disable@notes</code> .....	<u>1</u>
<code>\disable@sidenotes</code> .....	<u>1</u>
<code>\disable@dtabfeet</code> .....	<u>1</u>
<code>\do@actions</code> .....	<u>1</u>
<code>\do@actions@fixedcode</code> .....	<u>1</u>
<code>\do@actions@next</code> .....	<u>1</u>
<code>\do@ballast</code> .....	<u>1</u>
<code>\do@feet@custom@order</code> .....	<u>1</u>
<code>\do@feetX</code> .....	<u>1</u>
<code>\do@insidelinehook</code> .....	<u>1</u>
<code>\do@line</code> .....	<u>1</u>
<code>\do@linehook</code> .....	<u>1</u>
<code>\do@lockoff</code> .....	<u>1</u>
<code>\do@lockoffL</code> .....	<u>1</u>
<code>\do@lockon</code> .....	<u>1</u>
<code>\do@lockonL</code> .....	<u>1</u>
<code>\do@Xfeet</code> .....	<u>1</u>
<code>\doedindexlabel</code> .....	<u>1</u>
<code>\doendnotes</code> .....	<u>1</u> , <u>29</u>
<code>\doendnotesbysection</code> .....	<u>1</u> , <u>29</u>
<code>\doinsidelinehook</code> .....	<u>1</u> , <u>26</u>
<code>\doinsidethislinehook</code> .....	<u>1</u> , <u>27</u>
<code>\dolinehook</code> .....	<u>1</u> , <u>26</u>
<code>\dosplits</code> .....	<u>1</u>
Downes, Michael .....	75, 186, 188
<code>\dummy@edindex</code> .....	<u>1</u>
<code>\dummy@edtext</code> .....	<u>1</u>
<code>\dummy@edtext@showlemma</code> .....	<u>1</u>
<code>\dummy@ref</code> .....	<u>1</u>

## E

<code>\edaftertab</code> .....	<u>1</u> , <u>1</u> , <u>1</u> , <u>70</u> , <u>355</u>
<code>edarrayc</code> (env.) .....	68
<code>edarrayl</code> (env.) .....	68
<code>edarrayr</code> (env.) .....	68
<code>\edatleft</code> .....	<u>1</u> , <u>70</u>
<code>\edatright</code> .....	<u>1</u> , <u>70</u>
<code>\edbforetab</code> .....	<u>1</u> , <u>1</u> , <u>1</u> , <u>70</u> , <u>355</u>
<code>\edfilldimen</code> .....	<u>1</u>
<code>\edfont@info</code> .....	<u>1</u>
<code>\edindex</code> .....	<u>1</u> , <u>65</u>
<code>\edindexlab</code> .....	<u>1</u> , <u>67</u>
<code>\edlabel</code> .....	<u>1</u> , <u>59</u>
<code>\edlabelE</code> .....	<u>1</u> , <u>61</u>
<code>\edlabelS</code> .....	<u>1</u> , <u>61</u>
<code>\edlabelSE</code> .....	<u>1</u> , <u>61</u>
<code>\edlineref</code> .....	<u>1</u> , <u>59</u>
<code>\edmakelabel</code> .....	<u>1</u> , <u>61</u>
<code>\edpageref</code> .....	<u>1</u> , <u>59</u>
<code>\edrowfill</code> .....	<u>1</u> , <u>69</u>
<code>\EDTAB</code> .....	<u>1</u>

<code>\edtabcolsep</code> .....	<u>1</u> , 69
<code>\EDTABINDENT</code> .....	<u>1</u>
<code>\edtabindent</code> .....	<u>1</u>
<code>\EDTABtext</code> .....	<u>1</u>
<code>edtabularc</code> (env.) .....	68
<code>edtabularl</code> (env.) .....	68
<code>edtabularr</code> (env.) .....	68
<code>\EDTEXT</code> .....	<u>1</u>
<code>\edtext</code> .....	<u>1</u> , 27
<code>\edtext@notes</code> .....	<u>1</u>
<code>\edtext@notes@start</code> .....	<u>1</u>
<code>\advertdots</code> .....	<u>1</u> , 71
<code>\advertline</code> .....	<u>1</u> , 71
<code>\Eendnote</code> .....	28
<code>\Efootnote</code> .....	28
<code>\eled@chapter</code> .....	<u>1</u>
<code>\eled@section</code> .....	<u>1</u>
<code>\eled@sectioning@out</code> .....	<u>1</u>
<code>\eled@subsection</code> .....	<u>1</u>
<code>\eled@subsubsection</code> .....	<u>1</u>
<code>\eledchapter</code> .....	<u>1</u>
<code>\eledchapter*</code> .....	<u>1</u>
<code>\eledsection</code> .....	<u>1</u>
<code>\eledsection*</code> .....	<u>1</u>
<code>\eledsubsection</code> .....	<u>1</u>
<code>\eledsubsection*</code> .....	<u>1</u>
<code>\eledsubsubsection</code> .....	<u>1</u>
<code>\eledsubsubsection*</code> .....	<u>1</u>
<code>\enablel@dtabfeet</code> .....	<u>1</u>
<code>\end@lemmas</code> .....	<u>1</u>
<code>\endashchar</code> .....	<u>1</u>
<code>\endline@num</code> .....	<u>1</u>
<code>\endlock</code> .....	<u>1</u> , 22
<code>\endminipage</code> .....	<u>1</u>
<code>\endnumbering</code> .....	<u>1</u> , 16
<code>\endpage@num</code> .....	<u>1</u>
<code>\endprint</code> .....	<u>1</u>
<code>\endquotation</code> .....	<u>1</u>
<code>\endquote</code> .....	<u>1</u>
<code>\endsub</code> .....	<u>1</u> , 21
<code>\endsubline@num</code> .....	<u>1</u>
environments:	
<code>edarrayc</code> .....	68
<code>edarrayl</code> .....	68
<code>edarrayr</code> .....	68
<code>edtabularc</code> .....	68
<code>edtabularl</code> .....	68
<code>edtabularr</code> .....	68
<code>ledgroup</code> .....	59
<code>ledgroupsize</code> .....	59
<code>minipage</code> .....	59
Euclid .....	13
<code>\extensionchars</code> .....	<u>1</u> , 73

## F

<code>\f@x@l@cks</code> .....	<u>1</u>
Fairbairns, Robin .....	38
<code>\first@linenum@out@false</code> .....	<u>1</u>
<code>\first@linenum@out@true</code> .....	<u>1</u>
<code>\firsthalf@antilabe@wd</code> .....	<u>1</u>
<code>\firstlinenum</code> .....	<u>1</u> , 20
<code>\firstseriesX@</code> .....	<u>1</u>
<code>\firstsublinenum</code> .....	<u>1</u> , 20
<code>\firstXseries@</code> .....	<u>1</u>
<code>\flag@end</code> .....	<u>1</u>
<code>\flag@end@later</code> .....	<u>1</u>
<code>\flag@end@RTL</code> .....	<u>1</u>
<code>\flag@start</code> .....	<u>1</u>
<code>\flag@start@later</code> .....	<u>1</u>
<code>\flag@start@RTL</code> .....	<u>1</u>
<code>\flagstanza</code> .....	<u>1</u> , 58
<code>\flush@notes</code> .....	<u>1</u>
<code>\fnpos</code> .....	<u>1</u> , 39
Folkerts, Menso .....	13
<code>\footfootmarkA</code> .....	38
<code>\footfudgefiddle</code> .....	<u>1</u> , 75
<code>\footnote</code> .....	<u>1</u>
<code>\footnoteA</code> .....	37
<code>\footnoteB</code> .....	37
<code>\footnoteC</code> .....	37
<code>\footnoteD</code> .....	37
<code>\footnoteE</code> .....	37
<code>\footnotelang@lua</code> .....	<u>1</u>
<code>\footnotelang@poly</code> .....	<u>1</u>
<code>\footnoteoptions@</code> .....	<u>1</u>
<code>\footnoteXmark</code> .....	38
<code>\footnoteXtext</code> .....	38
<code>\footsplitskips</code> .....	<u>1</u>
<code>\fullstop</code> .....	<u>1</u>

## G

Gädeke, Nora .....	13
<code>\get@edindex@hyperref</code> .....	<u>1</u>
<code>\get@edindex@ledinnote@command</code> .....	<u>1</u>
<code>\get@firsthalf@antilabe@wd</code> .....	<u>1</u>
<code>\get@fnmark</code> .....	<u>1</u>
<code>\get@fnmarkX</code> .....	<u>1</u>
<code>\get@index@command</code> .....	<u>1</u>
<code>\get@linelistfile</code> .....	<u>1</u>
<code>\get@pstartcounter</code> .....	<u>1</u>
<code>\get@stanzacounter</code> .....	<u>1</u>
<code>\get@sw@txt</code> .....	<u>1</u>
<code>\get@thisfootnote</code> .....	<u>1</u>
<code>\get@thisfootnoteX</code> .....	<u>1</u>
<code>\getline@num</code> .....	<u>1</u>
<code>\gl@p</code> .....	<u>1</u>

## H

<code>\h@num</code> .....	<u>1</u>
<code>\hangindentX</code> .....	<u>49</u>
<code>\hidenumbering</code> .....	<u>1</u> , <u>23</u>
<code>\hidenumberingonleftpage</code> .....	<u>1</u> , <u>23</u>
<code>\hidenumberingonrightpage</code> .....	<u>1</u>
<code>\Hilfsbox</code> .....	<u>1</u>
<code>\hilfsbox</code> .....	<u>1</u>
<code>\hilfscount</code> .....	<u>1</u>
<code>\HILFSskip</code> .....	<u>1</u>
<code>\Hilfsskip</code> .....	<u>1</u>
<code>\hilfsskip</code> .....	<u>1</u>
<code>\hookarg@specific</code> .....	<u>1</u>
<code>\hooktoggle@specific</code> .....	<u>1</u>
<code>\hsizethreecolX</code> .....	<u>50</u>
<code>\hsizetwocolX</code> .....	<u>50</u>
<code>\Hy@raisedlink@left</code> .....	<u>1</u>
<code>\hyperlinkformat</code> .....	<u>1</u>
<code>\hyperlinkformatR</code> .....	<u>1</u>
<code>\hyperlinkR</code> .....	<u>1</u>

## I

<code>\if@addsw</code> .....	<u>1</u>
<code>\if@edtabular</code> .....	<u>1</u>
<code>\if@edtext@secondarg@</code> .....	<u>1</u>
<code>\if@eled@sectioning</code> .....	<u>1</u>
<code>\if@firstlineofpage</code> .....	<u>1</u>
<code>\if@led@nofoot</code> .....	<u>1</u>
<code>\if@ledgroup</code> .....	<u>1</u>
<code>\if@lemmacommand@</code> .....	<u>1</u>
<code>\if@msdata@insertedfrompreviouspage</code> .....	<u>1</u>
<code>\if@noeled@sec</code> .....	<u>1</u>
<code>\if@noneed@notes</code> .....	<u>1</u>
<code>\if@RTL</code> .....	<u>1</u>
<code>\ifafterendnumberingR</code> .....	<u>1</u>
<code>\ifantilabe@</code> .....	<u>1</u>
<code>\ifat@every@pend@star@</code> .....	<u>1</u>
<code>\ifat@every@pstart@star@</code> .....	<u>1</u>
<code>\ifautopar@pause</code> .....	<u>1</u>
<code>\ifbypage@</code> .....	<u>1</u>
<code>\ifbypage@R</code> .....	<u>1</u>
<code>\ifbypstart@</code> .....	<u>1</u>
<code>\ifbypstart@R</code> .....	<u>1</u>
<code>\ifeledmaccompat@</code> .....	<u>1</u>
<code>\iffirst@linenum@out@</code> .....	<u>1</u>
<code>\ifinastanzaL</code> .....	<u>1</u>
<code>\ifinastanzaR</code> .....	<u>1</u>
<code>\ifindtl@innote</code> .....	<u>1</u>
<code>\ifindtl@notenumber</code> .....	<u>1</u>
<code>\ifinserthangingsymbol</code> .....	<u>1</u>
<code>\ifinstanza</code> .....	<u>1</u>
<code>\ifinstanzaL</code> .....	<u>1</u>
<code>\ifinstanzaR</code> .....	<u>1</u>
<code>\ifl@d@dash</code> .....	<u>1</u>

<code>\ifl@d@elin</code>	<u>1</u>
<code>\ifl@d@esl</code>	<u>1</u>
<code>\ifl@d@pnum</code>	<u>1</u>
<code>\ifl@d@ssub</code>	<u>1</u>
<code>\ifl@d@Xendlinenum</code>	<u>1</u>
<code>\ifl@d@Xlinenum</code>	<u>1</u>
<code>\ifl@d@Xmorethantwolines</code>	<u>1</u>
<code>\ifl@d@Xtwolines</code>	<u>1</u>
<code>\ifl@dend@X</code>	<u>1</u>
<code>\ifl@dhiddenumber</code>	<u>1</u>
<code>\ifl@dmemoir</code>	<u>1</u>
<code>\ifl@dpaging</code>	<u>1</u>
<code>\ifl@dpairing</code>	<u>1</u>
<code>\ifl@dprintingcolumns</code>	<u>1</u>
<code>\ifl@dprintingpages</code>	<u>1</u>
<code>\ifl@dskipnumber</code>	<u>1</u>
<code>\ifl@dskipversenumber</code>	<u>1</u>
<code>\ifl@dstartendok</code>	<u>1</u>
<code>\ifl@footmisc</code>	<u>1</u>
<code>\ifl@imakeidx</code>	<u>1</u>
<code>\ifl@indextools</code>	<u>1</u>
<code>\ifledfinal</code>	<u>1, 73</u>
<code>\ifledgroupnotesL@</code>	<u>1</u>
<code>\ifledgroupnotesR@</code>	<u>1</u>
<code>\iflednopbinverse</code>	<u>1, 1</u>
<code>\ifledRcol</code>	<u>1</u>
<code>\ifledRcol@</code>	<u>1</u>
<code>\iflinenumannotationotherside</code>	<u>1</u>
<code>\ifnocritical@</code>	<u>1</u>
<code>\ifnoend@</code>	<u>1</u>
<code>\ifnofamiliar@</code>	<u>1</u>
<code>\ifnoledgroup@</code>	<u>1</u>
<code>\ifnoquotation@</code>	<u>1</u>
<code>\ifnoresetlinenumannotation@</code>	<u>1</u>
<code>\ifnoteschanged@</code>	<u>1</u>
<code>\ifnumberedpar@</code>	<u>1</u>
<code>\ifnumbering</code>	<u>1</u>
<code>\ifnumberingR</code>	<u>1</u>
<code>\ifnumberline</code>	<u>1</u>
<code>\ifnumberstanza</code>	<u>1</u>
<code>\ifparapparus@</code>	<u>1</u>
<code>\ifparledgroup</code>	<u>1</u>
<code>\ifprevpgnotnumbered</code>	<u>1</u>
<code>\ifpst@rtedL</code>	<u>1</u>
<code>\ifresumenummering@start</code>	<u>1</u>
<code>\ifsameparallelpagenumber</code>	<u>1</u>
<code>\ifseriesbefore</code>	<u>1</u>
<code>\ifsidepstartnum</code>	<u>1</u>
<code>\ifstopmsdata@inserted@</code>	<u>1</u>
<code>\ifsublines@</code>	<u>1</u>
<code>\ifsublines@R</code>	<u>1</u>
<code>\ifwidthliketwocolumns</code>	<u>1</u>
<code>\ifXendinsertsep@</code>	<u>1</u>
<code>\ifxindy@</code>	<u>1</u>

<code>\ifxindyhyperref@</code> .....	<u>1</u>
<code>\initnumbering@quote</code> .....	<u>1</u>
<code>\initnumbering@reg</code> .....	<u>1</u>
<code>\innotemarkX</code> .....	<u>66</u>
<code>\insert@count</code> .....	<u>0, 1</u>
<code>\insert@msdata</code> .....	<u>1</u>
<code>\insert@txtbeforenotesX</code> .....	<u>1</u>
<code>\insert@Xtxtbeforenotes</code> .....	<u>1</u>
<code>\inserthangingymbol</code> .....	<u>1</u>
<code>\insertlines@list</code> .....	<u>1</u>
<code>\insertparafootsepX</code> .....	<u>1</u>
<code>\inserts@list</code> .....	<u>1</u>

**J**

Jayaditya .....	13
-----------------	----

**K**

Kabelschacht, Alois .....	174
<code>\keep@this@crossref@forinserting</code> .....	<u>1</u>

**L**

<code>\l@advance@parledgroup@beforenormalnotes</code> .....	<u>1</u>
<code>\l@d@add</code> .....	<u>1</u>
<code>\l@d@nums</code> .....	<u>1</u>
<code>\l@d@section</code> .....	<u>1</u>
<code>\l@d@set</code> .....	<u>1</u>
<code>\l@d@Xend</code> .....	<u>1</u>
<code>\l@dampcount</code> .....	<u>1</u>
<code>\l@dbfnote</code> .....	<u>1</u>
<code>\l@dcheckcols</code> .....	<u>1</u>
<code>\l@dcheckstartend</code> .....	<u>1</u>
<code>\l@dchset@num</code> .....	<u>1</u>
<code>\l@dcolcount</code> .....	<u>1</u>
<code>\l@dcollect@body</code> .....	<u>1</u>
<code>\l@dcollect@body</code> .....	<u>1</u>
<code>\l@dcolwidth</code> .....	<u>1</u>
<code>\l@dcsnote</code> .....	<u>1</u>
<code>\l@dcsnotetext</code> .....	<u>1</u>
<code>\l@dcsnotetext@l</code> .....	<u>1</u>
<code>\l@dcsnotetext@r</code> .....	<u>1</u>
<code>\l@ddodoreintrafeet</code> .....	<u>1</u>
<code>\l@ddoxtrafeet</code> .....	<u>1</u>
<code>\l@dedbeginmini</code> .....	<u>1</u>
<code>\l@dedendmini</code> .....	<u>1</u>
<code>\l@emptyd@ta</code> .....	<u>1</u>
<code>\l@dend@close</code> .....	<u>1</u>
<code>\l@dend@open</code> .....	<u>1</u>
<code>\l@dend@stuff</code> .....	<u>1</u>
<code>\l@dend@Xfalse</code> .....	<u>1</u>
<code>\l@dend@Xtrue</code> .....	<u>1</u>
<code>\l@denbody</code> .....	<u>1</u>
<code>\l@desnote</code> .....	<u>1</u>
<code>\l@dfambeginmini</code> .....	<u>1</u>
<code>\l@dfamendmini</code> .....	<u>1</u>



<code>\l@dfheetbeginmini</code>	<u>1</u>
<code>\l@dfheetendmini</code>	<u>1</u>
<code>\l@dgetline@margin</code>	<u>1</u>
<code>\l@dgetlock@disp</code>	<u>1</u>
<code>\l@dgetref@num</code>	<u>1</u>
<code>\l@dgetsidenote@margin</code>	<u>1</u>
<code>\l@dobbleoptarg</code>	<u>1</u>
<code>\l@disnote</code>	<u>1</u>
<code>\l@dlabel@parse</code>	<u>1</u>
<code>\l@dld@ta</code>	<u>1, 1</u>
<code>\l@dlp@rbox</code>	<u>1</u>
<code>\l@dlsn@te</code>	<u>1</u>
<code>\l@dlsnote</code>	<u>1</u>
<code>\l@dmake@labels</code>	<u>1</u>
<code>\l@dmodforedtext</code>	<u>1</u>
<code>\l@dnullfills</code>	<u>1</u>
<code>\l@dnumpstartsL</code>	<u>1</u>
<code>\l@doldold@footnotetext</code>	<u>1</u>
<code>\l@dp@rsefootspec</code>	<u>1</u>
<code>\l@dparsedendline</code>	<u>1</u>
<code>\l@dparsedendpage</code>	<u>1</u>
<code>\l@dparsedendsub</code>	<u>1</u>
<code>\l@dparsedstartline</code>	<u>1</u>
<code>\l@dparsedstartpage</code>	<u>1</u>
<code>\l@dparsedstartsub</code>	<u>1</u>
<code>\l@dparsefootspec</code>	<u>1</u>
<code>\l@dpush@begins</code>	<u>1</u>
<code>\l@drd@ta</code>	<u>1, 1</u>
<code>\l@dref@undefined</code>	<u>1</u>
<code>\l@drestorefills</code>	<u>1</u>
<code>\l@drestoreforedtext</code>	<u>1</u>
<code>\l@drp@rbox</code>	<u>1</u>
<code>\l@drsn@te</code>	<u>1</u>
<code>\l@drsnote</code>	<u>1</u>
<code>\l@dsetmaxcolwidth</code>	<u>1</u>
<code>\l@dskipnumberfalse</code>	<u>1</u>
<code>\l@dskipnumbertrue</code>	<u>1</u>
<code>\l@dtabaddcols</code>	<u>1</u>
<code>\l@dtabnoexpands</code>	<u>1</u>
<code>\l@dunboxmpfoot</code>	<u>1</u>
<code>\l@dunhbox@line</code>	<u>1</u>
<code>\l@dzeropenalties</code>	<u>1</u>
<code>\l@noexpands@edgl</code>	<u>1</u>
<code>\l@secondmandarg</code>	<u>1</u>
<code>\l@wrap@ifnotemptybox</code>	<u>1</u>
<code>\l@wrapcs@ifnotemptybox</code>	<u>1</u>
<code>\label</code>	<u>61</u>
<code>\labelstartfalse</code>	<u>1</u>
<code>\labelstarttrue</code>	<u>1, 19</u>
<code>\labelref@list</code>	<u>1</u>
<code>\labelrefsparseline</code>	<u>1</u>
<code>\labelrefsparsesubline</code>	<u>1</u>
<code>\last@page@num</code>	<u>1</u>
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@edtextoutsidestart</code>	1
<code>\led@err@EdtextWithoutNote</code>	1
<code>\led@err@FootnoteNotInSecondArgEdtext</code>	1
<code>\led@err@HighEndColumn</code>	1
<code>\led@err@LineationInNumbered</code>	1
<code>\led@err@LowStartColumn</code>	1
<code>\led@err@ManyLeftnotes</code>	1
<code>\led@err@ManyRightnotes</code>	1
<code>\led@err@ManySidenotes</code>	1
<code>\led@err@NumberingNotStarted</code>	1
<code>\led@err@NumberingShouldHaveStarted</code>	1
<code>\led@err@NumberingStarted</code>	1
<code>\led@err@NumberingWithoutPstart</code>	1
<code>\led@err@PendInEdtext</code>	1
<code>\led@err@PendNoPstart</code>	1
<code>\led@err@PendNotNumbered</code>	1
<code>\led@err@PstartInEdtext</code>	1
<code>\led@err@PstartInPstart</code>	1
<code>\led@err@PstartNotNumbered</code>	1
<code>\led@err@ReverseColumns</code>	1
<code>\led@err@samewordRTL</code>	1
<code>\led@err@toendnotes@outsidenumbering</code>	1
<code>\led@err@TooManyColumns</code>	1
<code>\led@err@UnequalColumns</code>	1
<code>\led@error@fail@patch@@docclearpage</code>	1
<code>\led@error@fail@patch@@iiiminipage</code>	1
<code>\led@error@fail@patch@@makecol</code>	1
<code>\led@error@fail@patch@@reinserts</code>	1
<code>\led@error@fail@patch@addtocounter</code>	1
<code>\led@error@fail@patch@endminipage</code>	1, 1
<code>\led@error@fail@patch@mempnum</code>	1
<code>\led@error@fail@patch@setcounter</code>	1
<code>\led@error@fail@patch@stpelt</code>	1
<code>\led@error@PackageAfterEledmac</code>	1
<code>\led@error@require@glossariesextra</code>	1
<code>\led@mess@NotesChanged</code>	1
<code>\led@mess@SectionContinued</code>	1
<code>\led@nopb</code>	1
<code>\led@nopbnum</code>	1
<code>\led@pb</code>	1
<code>\led@pb@setting</code>	1
<code>\led@pbnum</code>	1
<code>\led@reinit@index@fornote</code>	1
<code>\led@set@index@fornote</code>	1
<code>\led@toksa</code>	1
<code>\led@toksb</code>	1
<code>\led@warn@AppLabelOutSecondArgEdtext</code>	1
<code>\led@warn@BadAction</code>	1
<code>\led@warn@BadAdvancelineLine</code>	1
<code>\led@warn@BadAdvancelineSubline</code>	1

<code>\led@warn@BadHookSeries</code>	<a href="#">1</a>
<code>\led@warn@BadLineation</code>	<a href="#">1</a>
<code>\led@warn@BadLinenummargin</code>	<a href="#">1</a>
<code>\led@warn@BadLockdisp</code>	<a href="#">1</a>
<code>\led@warn@BadSetline</code>	<a href="#">1</a>
<code>\led@warn@BadSetlinenum</code>	<a href="#">1</a>
<code>\led@warn@BadSidenotemargin</code>	<a href="#">1</a>
<code>\led@warn@BadSublockdisp</code>	<a href="#">1</a>
<code>\led@warn@DuplicateLabel</code>	<a href="#">1</a>
<code>\led@warn@edinde@outsidenumbering</code>	<a href="#">1</a>
<code>\led@warn@LineFileObsolete</code>	<a href="#">1</a>
<code>\led@warn@NoFile</code>	<a href="#">1</a>
<code>\led@warn@NoIndexFile</code>	<a href="#">1</a>
<code>\led@warn@NoMarginpars</code>	<a href="#">1</a>
<code>\led@warn@RefUndefined</code>	<a href="#">1, 1</a>
<code>\led@warn@SeriesStillExist</code>	<a href="#">1</a>
<code>\led@warning@hsizeX@deprecated</code>	<a href="#">1</a>
<code>\led@warning@msdatawithoutstop</code>	<a href="#">1</a>
<code>\led@warning@preXnotes@deprecated</code>	<a href="#">1</a>
<code>\led@warning@Xhsize@deprecated</code>	<a href="#">1</a>
<code>ledgroup (env.)</code>	<a href="#">59</a>
<code>ledgroupsized (env.)</code>	<a href="#">59</a>
<code>\ledinnernote</code>	<a href="#">1, 64</a>
<code>\ledinnote</code>	<a href="#">1</a>
<code>\ledinnotemark</code>	<a href="#">1</a>
<code>\ledleftnote</code>	<a href="#">1, 64</a>
<code>\ledlinenum</code>	<a href="#">1</a>
<code>\ledllfill</code>	<a href="#">1</a>
<code>\ledlsnotefontsetup</code>	<a href="#">1, 65</a>
<code>\ledlsnotesep</code>	<a href="#">1, 65</a>
<code>\ledlsnotewidth</code>	<a href="#">1, 65</a>
<code>\lednopb</code>	<a href="#">1, 73</a>
<code>\lednopbinversettrue</code>	<a href="#">73</a>
<code>\lednopbnum</code>	<a href="#">1</a>
<code>\ledouternote</code>	<a href="#">1, 64</a>
<code>\ledpb</code>	<a href="#">1, 73</a>
<code>\ledpbnum</code>	<a href="#">1</a>
<code>\ledpbsetting</code>	<a href="#">1, 73</a>
<code>\ledrightnote</code>	<a href="#">1, 64</a>
<code>\ledrlfill</code>	<a href="#">1</a>
<code>\ledrsnotefontsetup</code>	<a href="#">1, 65</a>
<code>\ledrsnotesep</code>	<a href="#">1, 65</a>
<code>\ledrsnotewidth</code>	<a href="#">1, 65</a>
<code>\ledsectnomark</code>	<a href="#">1</a>
<code>\ledsectnotoc</code>	<a href="#">1</a>
<code>\ledsetnormalparstuff@common</code>	<a href="#">1</a>
<code>\ledsetnormalparstuffX</code>	<a href="#">1</a>
<code>\ledsidenote</code>	<a href="#">1, 64</a>
<code>\ledsidenote@</code>	<a href="#">1</a>
<code>\leftctab</code>	<a href="#">1</a>
<code>\leftlinenum</code>	<a href="#">1, 21</a>
<code>\leftlinenumannotation</code>	<a href="#">1</a>
<code>\leftltab</code>	<a href="#">1</a>
<code>\leftnoteupfalse</code>	<a href="#">65</a>

<code>\leftpstartnum</code>	1
<code>\leftrtab</code>	1
Leibniz	13
<code>\lemma</code>	1, 30
<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, 25
<code>\lineannot@set</code>	1
<code>\lineation</code>	1, 21
<code>\linenum</code>	1, 30
<code>\linenum@out</code>	1
<code>\linenumannotation</code>	1, 23
<code>\linenumannotationothersidettrue</code>	25
<code>\linenumberlist</code>	1, 20
<code>\linenumberstyle</code>	1, 22
<code>\linenumincrement</code>	1, 20
<code>\linenummargin</code>	1, 21
<code>\linenumr@p</code>	1
<code>\linenumrep</code>	1
<code>\linenumsep</code>	1, 21
<code>\linerangesep@</code>	1
<code>\list@clear</code>	1
<code>\list@clearing@reg</code>	1
<code>\list@create</code>	1
<code>\lock@disp</code>	1
<code>\lock@off</code>	1
<code>\lock@on</code>	1
<code>\lockdisp</code>	1, 22
Lorch, Richard	13
<code>\ltab</code>	1
<code>\ltabtext</code>	1
Luecking, Dan	79

## 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>	52
Mayer, Gyula	13
<code>\measurembody</code>	1
<code>\measuremcell</code>	1
<code>\measuremrow</code>	1
<code>\measuretbody</code>	1
<code>\measuretcell</code>	1
<code>\measuretrow</code>	1
Middleton, Thomas	13, 104
<code>minipage (env.)</code>	59

Mittelbach, Frank	12, 13
\morenoexpands	<u>1</u> , <u>75</u>
\mp@append@notesX	<u>1</u>
\mp@append@Xnotes	<u>1</u>
\mpfnpos	<u>1</u> , <u>39</u>
\mpnormalfootgroup	<u>1</u>
\mpnormalfootgroupX	<u>1</u>
\mpnormalvfootnote	<u>1</u>
\mpnormalvfootnote@inserted	<u>1</u>
\mpnormalvfootnoteX	<u>1</u>
\mppara@footgroupX	<u>1</u>
\mppara@vfootnoteX	<u>1</u>
\mpparafootgroup	<u>1</u>
\mpparavfootnote	<u>1</u>
\mpthreecolfootgroup	<u>1</u>
\mpthreecolfootgroupX	<u>1</u>
\mpthreecolfootsetup	<u>1</u>
\mpthreecolfootsetupX	<u>1</u>
\mptwocolfootgroup	<u>1</u>
\mptwocolfootgroupX	<u>1</u>
\mptwocolfootsetup	<u>1</u>
\mptwocolfootsetupX	<u>1</u>
\msdata	<u>1</u> , <u>36</u>
\msdata@c	<u>1</u>
\msdata@cR	<u>1</u>
\multfootsep	<u>1</u> , <u>38</u>
\multiplefootnotemarker	<u>1</u>

## N

\n@num	<u>1</u>
\n@num@stanza	<u>1</u>
\new@line	<u>1</u>
\newhookarg@specific	<u>1</u>
\newhookcommand@series	<u>1</u>
\newhookcommand@series@reload	<u>1</u>
\newhooktoggle@series	<u>1</u>
\newhooktoggle@series@reload	<u>1</u>
\newhooktoggle@specific	<u>1</u>
\newseries@	<u>1</u>
\newverse	<u>1</u>
\NEXT	<u>1</u> , <u>1</u>
\next@line@list@stuff	<u>1</u>
\no@expands	<u>1</u>
\nomk@	<u>1</u>
\nonoteX	<u>41</u>
\normal@footnotemarkX	<u>1</u>
\normal@page@break	<u>1</u>
\normal@pars	<u>1</u>
\normalbfnoteX	<u>1</u>
\normalbodyfootmarkX	<u>1</u>
\normalfootfmt	<u>1</u>
\normalfootfmtX	<u>1</u>
\normalfootfootmarkX	<u>1</u>
\normalfootgroup	<u>1</u>

<code>\normalfootgroupX</code> .....	<u>1</u>
<code>\normalfootnoterule</code> .....	<u>1</u>
<code>\normalfootnoteruleX</code> .....	<u>1</u>
<code>\normalfootstart</code> .....	<u>1</u>
<code>\normalfootstartX</code> .....	<u>1</u>
<code>\normalvfootnote</code> .....	<u>1</u>
<code>\normalvfootnote@inserted</code> .....	<u>1</u>
<code>\normalvfootnoteX</code> .....	<u>1</u>
<code>\notefontsizeX</code> .....	48
<code>\notenumfontX</code> .....	48
<code>\noteschanged@false</code> .....	<u>1</u>
<code>\noteschanged@true</code> .....	<u>1</u>
<code>\noteswidthliketwocolumnsX</code> .....	53
<code>\nulledindex</code> .....	<u>1</u>
<code>\nullsetzen</code> .....	<u>1</u>
<code>\num@lines</code> .....	<u>1</u>
<code>\numberedpar@false</code> .....	<u>1</u>
<code>\numberedpar@true</code> .....	<u>1</u>
<code>\numberingfalse</code> .....	<u>1</u>
<code>\numberingtrue</code> .....	<u>1</u>
<code>\numberlinefalse</code> .....	20
<code>\numberlinetrue</code> .....	20
<code>\numberpstartfalse</code> .....	<u>1</u> , 19
<code>\numberpstarttrue</code> .....	<u>1</u> , 19
<code>\numberstanzafalse</code> .....	57
<code>\numberstanzatrue</code> .....	57
<code>\numlabfont</code> .....	<u>1</u> , 53

## O

<code>\old@hsize</code> .....	<u>1</u>
<code>\one@line</code> .....	<u>1</u>
<code>optionafter</code> .....	25
<code>optionantilabe</code> .....	58, 329
<code>optionauxdir</code> .....	15, 451
<code>optionbefore</code> .....	25
<code>optioncontinuousnumberingwithcolumns</code> .....	90, 156, 162, 450, 454, 457
<code>optiondraft</code> .....	73
<code>optionfinal</code> .....	73
<code>optioninnnote</code> .....	449
<code>optioninnote</code> .....	449
<code>optionlinrangesep</code> .....	274
<code>optionminus</code> .....	450
<code>optionnocritical</code> .....	447, 449
<code>optionnoeledsec</code> .....	72, 367, 451
<code>optionnoend</code> .....	449
<code>optionnofamiliar</code> .....	447, 455
<code>optionnopenalties</code> .....	75, 451
<code>optionnoprefix</code> .....	64, 455
<code>optionnoresetlinenumannotation</code> .....	23, 44, 455
<code>optionnotenumber</code> .....	449, 452
<code>optionparapparatus</code> .....	457
<code>optionplus</code> .....	450
<code>optionprefixmore</code> .....	64, 455
<code>optionsameparallelpagnumber</code> .....	454

optionswcaseinsensitive . . . . . 32, 453

## P

<code>\page@action</code> . . . . .	<u>1</u>
<code>\page@num</code> . . . . .	<u>1</u>
<code>\pagelinesep</code> . . . . .	<u>1</u> , 66
<code>\pageref</code> . . . . .	61
<code>\par@line</code> . . . . .	<u>1</u>
<code>\par@patch@pagenumbering</code> . . . . .	<u>1</u>
<code>\par@patch@thepage</code> . . . . .	<u>1</u>
<code>\para@footgroupX</code> . . . . .	<u>1</u>
<code>\para@footsetup</code> . . . . .	<u>1</u>
<code>\para@footsetupX</code> . . . . .	<u>1</u>
<code>\para@vfootnoteX</code> . . . . .	<u>1</u>
<code>\parafootfmt</code> . . . . .	<u>1</u>
<code>\parafootfmtX</code> . . . . .	<u>1</u>
<code>\parafootgroup</code> . . . . .	<u>1</u>
<code>\parafootsepX</code> . . . . .	51
<code>\parafootstart</code> . . . . .	<u>1</u>
<code>\parafootstartX</code> . . . . .	<u>1</u>
<code>\paravfootnote</code> . . . . .	<u>1</u>
<code>\parindentX</code> . . . . .	49
<code>\parse@annot</code> . . . . .	<u>1</u>
<code>\pausenumbering</code> . . . . .	<u>1</u> , 19
<code>\pausenumbering@page@num</code> . . . . .	<u>1</u>
<code>\pend</code> . . . . .	<u>1</u> , 17
Plato of Tivoli . . . . .	13
<code>\postbodyfootmark</code> . . . . .	<u>1</u>
<code>\prebodyfootmark</code> . . . . .	<u>1</u>
<code>\prenotesX</code> . . . . .	52
<code>\prepare@edindex@fornote</code> . . . . .	<u>1</u>
<code>\prepare@prenotesX</code> . . . . .	<u>1</u>
<code>\prepare@Xgroupbyline</code> . . . . .	<u>1</u>
<code>\prepare@Xprenotes</code> . . . . .	<u>1</u>
<code>\prev@nopb</code> . . . . .	<u>1</u>
<code>\prev@pb</code> . . . . .	<u>1</u>
<code>\prevpage@num</code> . . . . .	<u>1</u>
<code>\print@eledsection</code> . . . . .	<u>1</u>
<code>\print@footnoterule</code> . . . . .	<u>1</u>
<code>\print@leftmargin@eledsection</code> . . . . .	<u>1</u>
<code>\print@lemma</code> . . . . .	<u>1</u>
<code>\print@line</code> . . . . .	<u>1</u>
<code>\print@notesX</code> . . . . .	<u>1</u>
<code>\print@rightmargin@eledsection</code> . . . . .	<u>1</u>
<code>\print@Xfootnoterule</code> . . . . .	<u>1</u>
<code>\print@Xnotes</code> . . . . .	<u>1</u>
<code>\printendlines</code> . . . . .	<u>1</u>
<code>\printlineendnote</code> . . . . .	<u>1</u>
<code>\printlineendnotearea</code> . . . . .	<u>1</u>
<code>\printlinefootnote</code> . . . . .	<u>1</u>
<code>\printlinefootnotearea</code> . . . . .	<u>1</u>
<code>\printlinefootnotenumbers</code> . . . . .	<u>1</u>
<code>\printlines</code> . . . . .	<u>1</u>
<code>\printlines@count</code> . . . . .	<u>1</u>

<code>\printnpnum</code> .....	<u>1</u>
<code>\printpstart</code> .....	<u>1</u>
<code>\printsymlineendnotearea</code> .....	<u>1</u>
<code>\printsymlinefootnotearea</code> .....	<u>1</u>
<code>\printXafternumber</code> .....	<u>1</u>
<code>\printXbeforenumber</code> .....	<u>1</u>
<code>\pstart</code> .....	<u>1</u> , <u>17</u>
<code>\pstarteref</code> .....	<u>1</u>
<code>\pstartnum</code> .....	<u>1</u>
<code>\pstartref</code> .....	<u>59</u>
<code>\pstarts@read@L</code> .....	<u>1</u>
<code>\pstarts@typeset@L</code> .....	<u>1</u>

**Q**

<code>\quotation</code> .....	<u>1</u>
<code>\quote</code> .....	<u>1</u>

**R**

<code>\raggedX</code> .....	<u>51</u>
<code>\raw@text</code> .....	<u>1</u>
<code>\rbracket</code> .....	<u>1</u>
<code>\read@linelist</code> .....	<u>1</u>
<code>\ref</code> .....	<u>61</u>
<code>\ref@reg@later</code> .....	<u>1</u>
<code>\Relax</code> .....	<u>1</u>
<code>\reledmac@error</code> .....	<u>1</u>
<code>\reledmac@warning</code> .....	<u>1</u>
<code>\removehboxes</code> .....	<u>1</u>
<code>\reset@current@cannot</code> .....	<u>1</u>
<code>\reset@msd@options@iffullpage</code> .....	<u>1</u>
<code>\resetlinenumannotation</code> .....	<u>1</u>
<code>\resetprevline@</code> .....	<u>1</u> , <u>107</u>
<code>\resetprevpage@</code> .....	<u>1</u>
<code>\resetprevpage@num</code> .....	<u>108</u>
<code>\restore@edlabel</code> .....	<u>1</u>
<code>\restore@endnotes</code> .....	<u>1</u>
<code>\restore@familiarnotes</code> .....	<u>1</u>
<code>\restore@notes</code> .....	<u>1</u>
<code>\restore@sidenotes</code> .....	<u>1</u>
<code>\resumenumbering</code> .....	<u>1</u> , <u>19</u>
<code>\rightctab</code> .....	<u>1</u>
<code>\rightlinenum</code> .....	<u>1</u> , <u>21</u>
<code>\rightlinenumannotation</code> .....	<u>1</u>
<code>\rightlftab</code> .....	<u>1</u>
<code>\rightnoteupfalse</code> .....	<u>65</u>
<code>\rightrtab</code> .....	<u>1</u>
<code>\rightstartnum</code> .....	<u>1</u>
<code>\rigidbalance</code> .....	<u>1</u>
<code>\rigidbalanceX</code> .....	<u>1</u>
<code>\rtab</code> .....	<u>1</u>
<code>\rtabtext</code> .....	<u>1</u>

**S**

Sacrobosco .....	<u>13</u>
------------------	-----------



<code>\sameword</code> .....	<a href="#">1</a> , <a href="#">31</a>
<code>\sameword@inedtext</code> .....	<a href="#">1</a>
Schöpf, Rainer .....	13
<code>\section@num</code> .....	<a href="#">1</a>
<code>\select@lemmafонт</code> .....	<a href="#">1</a>
<code>\select@lemmafont</code> .....	<a href="#">1</a> , <a href="#">53</a>
<code>\SEref</code> .....	<a href="#">1</a> , <a href="#">61</a>
<code>\SErefonlypage</code> .....	<a href="#">61</a>
<code>\SErefwithpage</code> .....	<a href="#">1</a> , <a href="#">1</a> , <a href="#">61</a>
<code>\series</code> .....	<a href="#">1</a>
<code>\seriesatbegin</code> .....	<a href="#">1</a> , <a href="#">39</a>
<code>\seriesatend</code> .....	<a href="#">1</a> , <a href="#">39</a>
<code>\set@continuousnumberingforL</code> .....	<a href="#">1</a>
<code>\set@line</code> .....	<a href="#">1</a>
<code>\set@line@action</code> .....	<a href="#">1</a>
<code>\set@this@c@page</code> .....	<a href="#">1</a>
<code>\set@txtbeforenotesX</code> .....	<a href="#">1</a>
<code>\set@Xtxtbeforenotes</code> .....	<a href="#">1</a>
<code>\setapprefprefixmore</code> .....	<a href="#">62</a>
<code>\setapprefprefixsingle</code> .....	<a href="#">62</a>
<code>\setcommand@series</code> .....	<a href="#">1</a>
<code>\sethangindentX</code> .....	<a href="#">1</a>
<code>\sethangingsymbol</code> .....	<a href="#">1</a> , <a href="#">56</a>
<code>\setistwofollowinglines</code> .....	<a href="#">1</a>
<code>\setl@dlp@rbox</code> .....	<a href="#">1</a>
<code>\setl@drpr@box</code> .....	<a href="#">1</a>
<code>\setline</code> .....	<a href="#">1</a> , <a href="#">22</a>
<code>\setlinenum</code> .....	<a href="#">1</a> , <a href="#">22</a>
<code>\setlinenumannotationsep</code> .....	<a href="#">1</a> , <a href="#">26</a>
<code>\setmcellcenter</code> .....	<a href="#">1</a>
<code>\setmcellleft</code> .....	<a href="#">1</a>
<code>\setmcellright</code> .....	<a href="#">1</a>
<code>\setmrowcenter</code> .....	<a href="#">1</a>
<code>\setmrowleft</code> .....	<a href="#">1</a>
<code>\setmrowright</code> .....	<a href="#">1</a>
<code>\setmsdatalabel</code> .....	<a href="#">1</a> , <a href="#">37</a>
<code>\setmsdataposition</code> .....	<a href="#">1</a> , <a href="#">37</a>
<code>\setmsdataseries</code> .....	<a href="#">1</a> , <a href="#">37</a>
<code>\setnoteswidthliketwocolumnsX@</code> .....	<a href="#">1</a>
<code>\setnotesXpositionliketwocolumns@</code> .....	<a href="#">1</a>
<code>\setparindentX</code> .....	<a href="#">1</a>
<code>\setprintendlines</code> .....	<a href="#">1</a>
<code>\setprintlines</code> .....	<a href="#">1</a>
<code>\setSErefonlypageprefixmore</code> .....	<a href="#">62</a>
<code>\setSErefonlypageprefixsingle</code> .....	<a href="#">62</a>
<code>\setSErefprefixmore</code> .....	<a href="#">62</a>
<code>\setSErefprefixsingle</code> .....	<a href="#">62</a>
<code>\setsidenotesep</code> .....	<a href="#">65</a>
<code>\setstanzaindents</code> .....	<a href="#">1</a> , <a href="#">54</a>
<code>\setstanzapenalties</code> .....	<a href="#">1</a> , <a href="#">56</a>
<code>\setstanzavalues</code> .....	<a href="#">1</a>
<code>\settccllcenter</code> .....	<a href="#">1</a>
<code>\settccllleft</code> .....	<a href="#">1</a>
<code>\settccllright</code> .....	<a href="#">1</a>

<code>\settoggle@series</code> .....	<u>1</u>
<code>\setthrowcenter</code> .....	<u>1</u>
<code>\setthrowleft</code> .....	<u>1</u>
<code>\setthrowright</code> .....	<u>1</u>
<code>\setXnotespositionliketwocolumns@</code> .....	<u>1</u>
<code>\setXnoteswidthliketwocolumns@</code> .....	<u>1</u>
<code>\showlemma</code> .....	<u>1</u> , 73
<code>\showwordrank</code> .....	<u>1</u> , 35
<code>\sidenote@margin</code> .....	<u>1</u>
<code>\sidenotemargin</code> .....	<u>1</u> , 64
<code>\sidestartnumtrue</code> .....	19
<code>\skip@lockoff</code> .....	<u>1</u>
<code>\skipnumbering</code> .....	<u>1</u> , 23
<code>\splitoff</code> .....	<u>1</u>
<code>\spreadmath</code> .....	<u>1</u> , 69
<code>\spreadtext</code> .....	<u>1</u> , 69
<code>\stanza</code> .....	<u>1</u> , 54
<code>\stanza@count</code> .....	<u>1</u>
<code>\stanza@hang</code> .....	<u>1</u>
<code>\stanza@indent</code> .....	<u>1</u>
<code>\stanza@line</code> .....	<u>1</u>
<code>\stanzaindent</code> .....	<u>1</u> , 55
<code>\stanzaindent*</code> .....	<u>1</u> , 55
<code>\stanzaindentbase</code> .....	<u>1</u> , 54
<code>\stanz anumwrapper</code> .....	<u>1</u> , 57
<code>\startlock</code> .....	<u>1</u> , 22
<code>\startsub</code> .....	<u>1</u> , 21
<code>\stepl@dc colcount</code> .....	<u>1</u>
<code>\stopmsdata</code> .....	<u>1</u> , 36
<code>\store@annot@to@absline</code> .....	<u>1</u>
<code>\strip@szacnt</code> .....	<u>1</u>
<code>\sub@action</code> .....	<u>1</u>
<code>\sub@lock</code> .....	<u>1</u>
<code>\sub@off</code> .....	<u>1</u>
<code>\sub@on</code> .....	<u>1</u>
<code>\subline@num</code> .....	<u>1</u>
<code>\sublinenumberstyle</code> .....	<u>1</u> , 22
<code>\sublinenumincrement</code> .....	<u>1</u> , 20
<code>\sublinenumr@p</code> .....	<u>1</u>
<code>\sublinenumrep</code> .....	<u>1</u>
<code>\sublineref</code> .....	<u>1</u> , 59
<code>\sublines@false</code> .....	<u>1</u>
<code>\sublines@true</code> .....	<u>1</u>
<code>\sublock@disp</code> .....	<u>1</u>
<code>\sublockdisp</code> .....	<u>1</u>
Sullivan, Wayne .....	13, 75, 89, 94, 186, 188, 282, 322
<code>\sw@noexpand</code> .....	<u>1</u>
<code>\swnoexpands</code> .....	35
<code>\sza@penalty</code> .....	<u>1</u>
<b>T</b>	
<code>\tabHilfbox</code> .....	<u>1</u>
<code>\tabhilfbox</code> .....	<u>1</u>
<code>\theaddcolcount</code> .....	<u>1</u>

<code>\theedtext</code> .....	<u>1</u>
<code>\theendpageline</code> .....	<u>1</u>
<code>\thefootnoteA</code> .....	38
Theodosius .....	13
<code>\thepageline</code> .....	<u>1</u>
<code>\thepar@page</code> .....	<u>1</u>
<code>\thepstart</code> .....	<u>1</u> , 19
<code>\thestanza</code> .....	<u>1</u> , 57
<code>\thestartpageline</code> .....	<u>1</u>
<code>\this@line@list@version</code> .....	<u>1</u>
<code>\this@section@last@page@num</code> .....	<u>1</u>
<code>\this@section@page@num</code> .....	<u>1</u>
<code>\threecol@begin@insert</code> .....	<u>1</u>
<code>\threecolfootfmt</code> .....	<u>1</u>
<code>\threecolfootfmtX</code> .....	<u>1</u>
<code>\threecolfootgroup</code> .....	<u>1</u>
<code>\threecolfootgroupX</code> .....	<u>1</u>
<code>\threecolfootsetup</code> .....	<u>1</u>
<code>\threecolfootsetupX</code> .....	<u>1</u>
<code>\threecolvfootnote</code> .....	<u>1</u>
<code>\threecolvfootnote@inserted</code> .....	<u>1</u>
<code>\threecolvfootnoteX</code> .....	<u>1</u>
<code>\toendnotes</code> .....	<u>1</u> , 29
<code>\toendnotes*</code> .....	<u>1</u>
<code>\twocolfootfmt</code> .....	<u>1</u>
<code>\twocolfootfmtX</code> .....	<u>1</u>
<code>\twocolfootgroup</code> .....	<u>1</u>
<code>\twocolfootgroupX</code> .....	<u>1</u>
<code>\twocolfootsetup</code> .....	<u>1</u>
<code>\twocolfootsetupX</code> .....	<u>1</u>
<code>\twocolvfootnote</code> .....	<u>1</u>
<code>\twocolvfootnote@inserted</code> .....	<u>1</u>
<code>\twocolvfootnoteX</code> .....	<u>1</u>
<code>\txtbeforenotesonlyonceX</code> .....	51
<code>\txtbeforenotesX</code> .....	51

## U

<code>\unvxhX</code> .....	<u>1</u>
----------------------------	----------

## V

Vamana .....	13
<code>\variab</code> .....	<u>1</u>
<code>\vbfnoteX</code> .....	<u>1</u>
<code>\vl@dbfnote</code> .....	<u>1</u> , <u>1</u>
<code>\vl@dcsnote</code> .....	<u>1</u>
<code>\vl@disnote</code> .....	<u>1</u>
<code>\vl@dlsnote</code> .....	<u>1</u>
<code>\vl@dosnote</code> .....	<u>1</u>
<code>\vl@drsnote</code> .....	<u>1</u>
<code>\vnumfootnoteX</code> .....	<u>1</u>

## W

Whitney, Ron .....	13
<code>\widthX</code> .....	53

<code>\wrap@edcrossref</code> .....	<u>1</u>
<code>\wrapcontentX</code> .....	49
<code>\wrapped@bodyfootmarkX</code> .....	<u>1</u>
<code>\wrapped@footfootmarkX</code> .....	<u>1</u>
<code>\write@lemma</code> .....	<u>1</u>
Wujastyk, Dominik .....	12

## X

<code>\X@atbegininsertion</code> .....	<u>1</u>
<code>\X@beforeinsertion</code> .....	<u>1</u>
<code>\X@doreinfeet</code> .....	<u>1</u>
<code>\xabslineref</code> .....	<u>1</u>
<code>\Xafterlemmaseparator</code> .....	47
<code>\Xafternote</code> .....	51
<code>\Xafternumber</code> .....	45
<code>\Xafterrule</code> .....	52
<code>\Xaftersymlinenum</code> .....	45
<code>\xannotationref</code> .....	<u>1</u> , 60
<code>\Xarrangement</code> .....	<u>1</u> , 40
<code>\Xarrangement@normal</code> .....	<u>1</u>
<code>\Xarrangement@paragraph</code> .....	<u>1</u>
<code>\Xarrangement@threecol</code> .....	<u>1</u>
<code>\Xarrangement@twocol</code> .....	<u>1</u>
<code>\Xbeforeinserting</code> .....	50
<code>\Xbeforelemmaseparator</code> .....	47
<code>\Xbeforenotes</code> .....	52
<code>\Xbeforenumber</code> .....	45
<code>\Xbeforepagenumber</code> .....	42
<code>\Xbeforesymlinenum</code> .....	45
<code>\Xbhookgroup</code> .....	52
<code>\Xbhooknote</code> .....	50
<code>\Xboxlinenum</code> .....	46
<code>\Xboxlinenumalign</code> .....	46
<code>\Xboxsymlinenum</code> .....	46
<code>\Xcolalign</code> .....	50
<code>\xedindex</code> .....	<u>1</u>
<code>\xedtext</code> .....	<u>1</u>
<code>\Xendafterlemmaseparator</code> .....	47
<code>\Xendafternote</code> .....	53
<code>\Xendafterpagenumber</code> .....	42
<code>\Xendaftersymlinenum</code> .....	45
<code>\Xendaftrenumber</code> .....	45
<code>\Xendahookinplaceofnumber</code> .....	49
<code>\Xendahooklinenumber</code> .....	49
<code>\Xendbeforelemmaseparator</code> .....	47
<code>\Xendbeforenumber</code> .....	45
<code>\Xendbeforepagenumber</code> .....	42
<code>\Xendbeforesymlinenum</code> .....	45
<code>\Xendbhookinplaceofnumber</code> .....	49
<code>\Xendbhooklinenumber</code> .....	49
<code>\Xendbhooknote</code> .....	50
<code>\Xendboxendlinenum</code> .....	47
<code>\Xendboxlinenum</code> .....	47
<code>\Xendboxlinenumalign</code> .....	47

<code>\Xendboxstartline</code>	47
<code>\Xendboxsymlinum</code>	46
<code>\Xendhangindent</code>	49
<code>\Xendinplaceoflemmaseparator</code>	47
<code>\Xendinplaceofnumber</code>	46
<code>\Xendinplaceofpagenumber</code>	42
<code>\Xendlemmadisablefontselection</code>	48
<code>\Xendlemmafont</code>	48
<code>\Xendlemmaseparator</code>	47
<code>\Xendlinenumannotationonlyfirst</code>	45
<code>\Xendlinenumannotationonlyfirstintwo</code>	45
<code>\Xendlinenumannotationposition</code>	25
<code>\Xendlineprefixmore</code>	42
<code>\Xendlineprefixsingle</code>	42
<code>\Xendlinerangeseparator</code>	43
<code>\Xendmorethantwolines</code>	43
<code>\Xendnoidenticallinenumannotation</code>	26
<code>\Xendnolinenumberifannotation</code>	45
<code>\Xendnonumber</code>	43
<code>\Xendnotefontsize</code>	48
<code>\Xendnotenumfont</code>	48
<code>\Xendnumberonlyfirstinline</code>	41
<code>\Xendnumberonlyfirstintwolines</code>	41
<code>\Xendpagenumberonlyfirst</code>	42
<code>\Xendparagraph</code>	53
<code>\Xendsep</code>	53
<code>\Xendstorelineinfo</code>	1
<code>\Xendsublinesep</code>	45
<code>\Xendsymlinum</code>	41
<code>\Xendsymlinumannotation</code>	45
<code>\Xendsympagenum</code>	42
<code>\Xendtwolines</code>	43
<code>\Xendtwolinesbutnotmore</code>	43
<code>\Xendtxtbeforenotes</code>	51
<code>\Xendwrapcontent</code>	49
<code>\xflagref</code>	1
<code>\Xgroupbyline</code>	51
<code>\Xgroupbylineseparetwolines</code>	51
<code>\Xhangindent</code>	49
<code>\Xhsizethreecol</code>	50
<code>\Xhsizetwocol</code>	50
<code>\Xinnotemark</code>	66
<code>\Xinplaceoflemmaseparator</code>	47
<code>\Xinplaceofnumber</code>	46
<code>\Xinsertparafootsep</code>	1
<code>\Xledsetnormalparstuff</code>	1
<code>\xleft@appenditem</code>	1
<code>\Xlemmadisablefontselection</code>	48
<code>\Xlemmafont</code>	48
<code>\Xlemmaseparator</code>	47
<code>\Xlinenumannotationonlyfirst</code>	44
<code>\Xlinenumannotationonlyfirstintwo</code>	44
<code>\Xlinenumannotationposition</code>	25
<code>\Xlinenumannotationposition@side</code>	1

<code>\Xlinenumannotationpositionside</code> .....	25
<code>\Xlinerangeseparator</code> .....	43
<code>\xlineref</code> .....	<u>1</u> , 60
<code>\Xmaxhnotes</code> .....	52
<code>\Xmorethantwolines</code> .....	43
<code>\Xnoidenticallinenumannotation</code> .....	26
<code>\Xnolemmaseparator</code> .....	<u>1</u> , 47
<code>\Xnolinenumberifannotation</code> .....	45
<code>\Xnonbreakableafternumber</code> .....	45
<code>\Xnonote</code> .....	41
<code>\Xnonumber</code> .....	43
<code>\Xnopagenumberifcurrent</code> .....	43
<code>\Xnotboxingsubline</code> .....	46
<code>\Xnotefontsize</code> .....	48
<code>\Xnotenumfont</code> .....	48
<code>\Xnoteswidthliketwocolumns</code> .....	53
<code>\Xnumberonlyfirstinline</code> .....	41
<code>\Xnumberonlyfirstintwolines</code> .....	41
<code>\Xonlypstart</code> .....	44
<code>\Xpagelinesep</code> .....	45
<code>\xpageref</code> .....	<u>1</u> , 60
<code>\Xparafootsep</code> .....	51
<code>\Xparindent</code> .....	49
<code>\Xprenotes</code> .....	<u>1</u> , <u>1</u> , 52
<code>\Xprenotes@</code> .....	<u>1</u> , <u>1</u>
<code>\Xpstart</code> .....	44
<code>\Xpstarteverytime</code> .....	44
<code>\Xpstartonlyfirst</code> .....	44
<code>\xpstartref</code> .....	<u>1</u> , 60
<code>\Xpstartseparator</code> .....	44
<code>\XR@test</code> .....	<u>1</u>
<code>\XR@test@mac</code> .....	<u>1</u>
<code>\XR@test@mac@test</code> .....	<u>1</u>
<code>\Xragged</code> .....	51
<code>\xright@appenditem</code> .....	<u>1</u>
<code>\Xrigidbalance</code> .....	<u>1</u>
<code>\Xsethangindent</code> .....	<u>1</u>
<code>\Xsetparindent</code> .....	<u>1</u>
<code>\Xstanza</code> .....	44
<code>\Xstanzaonlyfirst</code> .....	44
<code>\Xstanzaseparator</code> .....	44
<code>\Xstorelineinfo</code> .....	<u>1</u>
<code>\xsublineref</code> .....	<u>1</u> , 60
<code>\Xsublinesep</code> .....	22, 45
<code>\Xsublinesepsid</code> .....	22
<code>\Xsymlinenum</code> .....	41
<code>\Xsymlinenumannotation</code> .....	44
<code>\Xtoendnotes</code> .....	29
<code>\Xtwolines</code> .....	43
<code>\Xtwolinesonlyinsamepage</code> .....	43
<code>\Xtxtbeforenotes</code> .....	51
<code>\Xtxtbeforenotesonlyonce</code> .....	51
<code>\Xtxtbeforenumber</code> .....	42
<code>\Xunvxh</code> .....	<u>1</u>

<code>\Xwidth</code> .....	53
<code>\Xwrapcontent</code> .....	49
<code>\Xwrapendlemma</code> .....	48
<code>\Xwraplemma</code> .....	48
<code>\Xwraplinenumannotation@ref</code> .....	<u>1</u>
<code>\Xwraplinenumannotation@side</code> .....	<u>1</u>
<code>\xxref</code> .....	<u>1</u> , 60

**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.		
General: Added tabmac code, and extended indexing	.....	1
\ifl@dmemoir: Added \ifl@dmemoir for memoir class having been used	.....	81
\morenoexpands: Added \l@dtabnoexpands to \no@expands	.....	133
\reledmac@error: Added \eledmac@error and replaced error messages	.....	82
v0.2.1.		
General: Added text about normal labeling	.....	61
Bug fixes and match with mempatch v1.8	.....	1
Major changes to insert code when memoir is loaded	.....	277
\@lab: Removed page setting from \@lab	.....	285
\edlabel: Tweaked \edlabel to get correct page numbers	.....	282
\l@ddodoreintrafeet: Renamed \dodoreintrafeet to \l@ddodoreintrafeet	.....	276
\l@ddoxtrafeet: Renamed \doxtrafeet to \l@ddoxtrafeet	.....	274
\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	.....	133
v0.2.2.		
General: Improved paragraph footnotes	.....	1
New Dekker example	.....	1
Used \providecommand for \@gobblethree to avoid clash with the amsfonts package	.....	89
\footfudgefiddle: Added \footfudgefiddle	.....	185
\next@line@list@stuff: Added initial write of page number in \line@list@stuff	.....	124
\para@footsetup: Added \footfudgefiddle to \para@footsetup	.....	185
\para@footsetupX: Added \footfudgefiddle to \para@footsetupX	.....	228
v0.3.0.		
General: Includes edstanza and more	.....	1
\@lab: Replaced \the\line@num by \linenumr@p\line@num in \@lab, and similar for sub-lines	.....	285
\@nl@reg: Added a bunch of code to \@nl for handling \setlinenum	.....	111
\ledlinenum: Added \linenumr@p and \sublinenumr@p to \leftlinenum and \rightlinenum	.....	102
\linenumberlist: Added \linenumberlist mechanism	.....	89
\printendlines: Added \linenumr@p and \sublinenumr@p to \printendlines	.....	247
\printlines@count: Added \linenumr@p and \sublinenumr@p to \printlines	.....	211
\sublinenumr@p: Added \linenumberstyle and \sublinenumberstyle	.....	102
v0.3.1.		
General: Not released. Added remarks about the parallel package	.....	1
v0.4.0.		
General: Added final/draft options	.....	78
Added ledgroup environment	.....	309
Added ledgroupsize environment	.....	309
Added minipage, etc., support	.....	1
\@iiiminipage: Modified kernel \@iiiminipage and \endminipage to cater for critical footnotes	.....	308
\edtext: Added \showlemma to \edtext	.....	134
\l@dfeetendmini: Added \l@dfeetbeginmini, \l@dfeetendmini and all their supporting code	.....	306
\mpnormalfootgroup: Added \mpnormalfootgroup	.....	184
\mpnormalvfootnote: Added \mpnormalvfootnote	.....	182



\showlemma: Added \showlemma	89
\Xarrangement@normal: Added minpage footnote setup to \footnormal	180
v0.4.1.	
General: Added code for changing \docclearpage	278
Not released. Minor editorial improvements and code tweaks	1
Only change \footnotetext and \footnotemark if memoir not used	215
\do@Xfeet: Changed \do@Xfeet code for easier extensions	275
\edindex: Let eledmac take advantage of memoir's indexing	314
\print@Xnotes: Added \opXfeet	275
v0.5.0.	
General: Added left and right sidenotes	298
Added sidenotes, familiar footnotes in numbered text	1
\@footnotetext: Enabled regular \footnote in numbered text	215
\@xympar: Eliminated \marginpar disturbance	298
v0.5.1.	
General: Added moveable sidenote	298
Fixed right line numbers killed in v0.5	1
Only change \hsize in ledgroupsize environment otherwise page number can be in wrong place	309
\affixline@num: Changed \affixline@num to cater for sidenotes	166
\l@dgetsidenote@margin: Added \sidenotemargin and \sidenote@margin	299
v0.6.0.	
General: Fixed long paragraphs looping	1
Fixed minor typos	1
Prepared for eledpar package	1
\@lopR: Added \@pend, \@pendR, \@lopL and \@lopR in anticipation of parallel processing	114
\@nl@reg: Extended \@nl to include the page number	111
\get@thisfootnote: Changed \l@dbfnote and \v1@dbfnote as originals could give incorrect markers in the footnotes	216
\set@this@page: Extended \new@line to output page numbers	125
v0.7.0.	
General: Added \setprintendlines for use by \printendlines	243
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	82
Replaced all \interAfootnotelinepenalty, etc., by just \interfootnotelinepenalty	1
Tidying up for eledpar and ledarab packages	1
\@nl@reg: Added \@nl@reg	111
\@ref@reg: Added \@ref@reg	121
\affixline@num: Added skipnumering to \affixline@num	166
\do@actions@fixedcode: Added \do@actions@fixedcode	162
\do@actions@next: Added number skipping to \do@actions	160
\do@insidelinehook: Added \do@linehook for use in \do@line	157
\endnumbering: Changed \endnumbering for eledpar	93
\fx@l@cks: Added \ch@cksub@l@ck, \ch@ck@l@ck and \fx@l@cks	168
\footsplitskips: Added \footsplitskips for use in many footnote styles	179
\get@linelistfile: Added \get@linelistfile	110
\initnumbering@reg: Added \initnumbering@reg	92
\l@advance@parledgroup@beforenormalnotes: Added \l@dunboxmpfoot containing some common code	308
\l@dcsnotetext@r: Added \l@demptyd@ta	158

<code>\l@dgetline@margin</code> : Added <code>\l@dgetline@margin</code> . . . . .	99
<code>\l@dgetlock@disp</code> : Added <code>\l@dgetlock@disp</code> . . . . .	101
<code>\l@dgetsidenote@margin</code> : Added <code>\l@dgetsidenote@margin</code> . . . . .	299
<code>\l@dnumstartsL</code> : Added <code>\l@dnumstartsL</code> , <code>\ifl@dpairing</code> and <code>\ifpst@rted</code> for/from <code>eledpar</code> . . . . .	90
<code>\l@drsn@te</code> : Added <code>\l@dlsn@te</code> and <code>\l@drsn@te</code> for use in <code>\do@line</code> . . . . .	158
<code>\l@dzeropenalties</code> : Added <code>\l@dzeropenalties</code> . . . . .	152
<code>\ledlinenum</code> : Added <code>\ledlinenum</code> for use by <code>\leftlinenum</code> and <code>\rightlinenum</code> . . . . .	102
<code>\list@clearing@reg</code> : Added <code>\list@clearing@reg</code> . . . . .	109
<code>\n@num</code> : Added <code>\n@num</code> . . . . .	119
<code>\next@line@list@stuff</code> : Deleted <code>\page@start</code> from <code>\line@list@stuff</code> . . . . .	124
<code>\normalbfnoteX</code> : Removed extraneous space from <code>\normalbfnoteX</code> . . . . .	221
<code>\pausenumbering@page@num</code> : Changed <code>\resumenumbering</code> for <code>eledpar</code> . . . . .	95
<code>\setprintlines</code> : Added <code>\setprintlines</code> for use by <code>\printlines</code> . . . . .	205
<code>\skipnumbering</code> : Added <code>\skipnumbering</code> and supports . . . . .	130
<code>\sublinenumincrement</code> : Added <code>\firstlinenum</code> , <code>\linenumincrement</code> , <code>\firstsublinenum</code> and <code>\linenumincrement</code> . . . . .	100
<code>\sublinenumr@p</code> : Using <code>\linenumrep</code> instead of <code>\linenumr@p</code> . . . . .	102
Using <code>\sublinenumrep</code> instead of <code>\sublinenumr@p</code> . . . . .	102
<code>\vnumfootnoteX</code> : Removed extraneous space from <code>\vnumfootnoteX</code> . . . . .	223
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 <code>\edtext</code> ; <code>\critex</code> ; <code>\lemma</code> fixed: we can now us non-switching commands . . . . .	1
v0.9.0.	
General: No more <code>ledpatch</code> . All patches are now in the main file. . . . .	1
v0.9.1.	
General: Fix some bugs linked to integrating <code>ledpatch</code> on the main file. . . . .	1
v0.10.0.	
General: Corrections to <code>\section</code> 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 <code>\hangingsymbol</code> to define the character. . . . .	1
v0.12.0.	
General: For compatibility with <code>eledpar</code> , possibility to use <code>\autopar</code> on the right side. . . . .	1
Possibility to number <code>\pstart</code> . . . . .	19
Possibility to number the <code>pstart</code> with the commands <code>\numberpstarttrue</code> . . . . .	1
<code>\l@dnumstartsL</code> : Added <code>\ifledRcol</code> and <code>\ifnumberingR</code> for/from <code>eledpar</code> . . . . .	90
v0.12.1.	
General: Do not number <code>\pstarts</code> of stanza. . . . .	1
The numbering of <code>\pstarts</code> restarts on each <code>\beginnumbering</code> . . . . .	1
v0.13.0.	
General: New <code>stanzaindentsrepetition</code> counter to repeat stanza indents every $n$ verses. . . . .	55
New <code>stanzaindentsrepetition</code> counter: to repeat stanza indents every $n$ verses. . . . .	1
<code>\managestanza@modulo</code> : New <code>stanzaindentsrepetition</code> counter to repeat stanza indents every $n$ verses. . . . .	324
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.	
General: Tweaked <code>\edlabel</code> to get correct line number if the command is first element of a paragraph. . . . .	1

\edlabel: Tweaked \edlabel to get correct line number if the command is first element of a paragraph. . . . .	282
v0.15.0.	
General: Line numbering can be reset at each pstart. . . . .	97
Possibility to print \pstart number at the sides of the text. . . . .	19
\affixline@num: Line numbering can be disabled. . . . .	166
\ifinserthangingsymbol: New management of hangingsymbol insertion, preventing undesirable insertions. . . . .	322
\printlines@count: Line numbering can be reset at each pstart. . . . .	210
v0.17.0.	
\ifinserthangingsymbol: New new management of hangingsymbol insertion, preventing undesirable insertions. . . . .	322
v1.0.0.	
General: \lemma can contain commands. . . . .	30
Debug in lineation command . . . . .	21
New generic commands to customize footnote display. . . . .	40, 264
Options nonum and nose in \Xfootnote. . . . .	28
Options of \Xfootnotes. . . . .	176
Possibility to have commands in sidenotes. . . . .	64
Some compatibility break with eledmac. Change of name: eledmac. . . . .	1
\morenoexpands: Change to be compatible with new features . . . . .	133
v1.0.1.	
General: Correction on \Xnumberonlyfirstinline with lineation by pstart or by page. . . . .	41
v1.1.0.	
General: Add \labelpstarttrue. . . . .	19
Add \Xnumberonlyfirstintwolines . . . . .	41
Add \Xpstart and \Xonlypstart . . . . .	44
New hook to add arbitrary code at the beginning of the notes . . . . .	49
New options for block of notes. . . . .	51
New package option: parapparatus. . . . .	1
New tools to change order of series . . . . .	263
Sectioning commands. . . . .	71
\settoggle@series: \settoggle@series switch the global value of the toggle, not only the local value. . . . .	265
\Xprenotes: New skip \Xprenotes@ . . . . .	236
v1.2.0.	
\endquote: Compatibility of \ledchapter with the <i>memoir</i> class. . . . .	360
\Xprenotes: Debug in familiar footnotes (bug introduced by v1.1). . . . .	236
v1.3.0.	
\endquote: <i>Quotation</i> and quote environment inside numbered sections. . . . .	360
v1.4.0.	
General: Compatibility with LuaTeX of RTL notes. . . . .	1
\edtext: Compatibility of \edtext with the right-to-left direction (with Polyglossia). . . . .	134
\ledsetnormalparstuffX: Direction of footnotes with polyglossia. . . . .	233
\newseries@: Remembers the language of the lemma, in order to create a correct direction for the footnote separator. . . . .	254
\rbracket: Switch the right bracket to a left bracket when the lemma is RTL (needs polyglossia or LuaTeX). . . . .	199
v1.4.1.	
\affixside@note: Remove spurious spaces. . . . .	304
\endquote: New option <i>noquotation</i> . . . . .	360
\get@thisfootnote: Compatibility of standard footnotes with eledmac when these footnotes contain any commands. . . . .	216
\labelrefsparsesubline: Fix a bug with \edlabel. . . . .	284

v1.4.2.	General: Debug with some special classes. . . . .	1
v1.4.3.	General: Add <code>\Xnonbreakableafternumber</code> . . . . .	45
	Spurious space after familiar footnotes. . . . .	1
v1.4.4.	General: Label inside familiar footnotes. . . . .	1
v1.4.5.	General: Bug with <code>komasscript + eledpar + chapter</code> . . . . .	1
v1.4.6.	General: Bug with memoir class introduced by 1.4.5. . . . .	1
v1.4.7.	<code>\endquote</code> : Compatibility of sectioning commands with <code>\autopar</code> . . . . .	360
v1.4.8.	General: Corrects a bug with parallel texts introduced by 1.1. . . . .	1
v1.4.9.	<code>\normalbfnoteX</code> : Allow to redefine <code>\thefootnoteX</code> with <code>alph</code> when some packages are loaded. . . . .	221
v1.5.0.	General: Correct indexing when the call is made in critical notes. . . . .	310
	<code>\do@insidelinehook</code> : Added <code>\do@insidelinehook</code> for use in <code>\do@line</code> . . . . .	157
	<code>\edindex</code> : Compatibility with <code>imakeidx</code> package, and possibility to use multiple index with <code>\edindex</code> . . . . .	314
v1.5.1.	<code>\managestanza@modulo</code> : Correct <code>stanzaindentsrepetition</code> counter . . . . .	324
	<code>\normalvfootnoteX</code> : Fix a bug with normal familiar footnotes when mixing RTL and LTR text. . . . .	218
v1.6.0.	<code>\newverse</code> : Add <code>\falseverse</code> macro. . . . .	327
v1.6.1.	General: Corrects a false hanging verse when a verse is exactly the length of a line. . . . .	1
	<code>\AtStartEveryPstart</code> : Spurious space in <code>\pstart</code> . . . . .	148
	<code>\ifinserthangingsymbol</code> : Hang verse is now not automatically flush right. . . . .	322
	<code>\l@dunhbox@line</code> : Move the call to <code>\inserthangingsymbol</code> to allow use <code>\hfill</code> inside. . . . .	154
	<code>\pend</code> : Spurious space in <code>\pend</code> . . . . .	150
v1.7.0.	General: New features for managing page breaks. . . . .	73
v1.8.0.	General: Compatibility with <code>parledgroup</code> option of <code>eledpar</code> package. . . . .	1
	If <code>imakeidx</code> and <code>hyperref</code> are loaded, adds <code>hyperref</code> in the index. . . . .	310
	<code>\endquote</code> : Correction of sectioning commands in parallel texts. . . . .	360
	<code>\get@index@command</code> : Debug <code>\get@index@command</code> and compatibility with <code>hyperref</code> package. . . . .	313
	<code>\newhookcommand@series@reload</code> : Debug <code>\beforenotesX</code> and <code>\maxhnotesX</code> which did not work. . . . .	267
	<code>\prevpage@num</code> : Correct <code>\parafootsep</code> when using with <code>ledgroup</code> . . . . .	190
v1.8.1.	General: Debug endnotes when more than one series is used (change the position where tools for endnotes are defined). . . . .	237
v1.8.2.	General: Debug compatibility problem with <code>hebrew</code> option of <code>babel</code> package. . . . .	1
v1.8.3.	General: Fix spurious spaces added by v1.7.0. . . . .	1

v1.8.5.	General: Debug indexing in right column, with eledpar. . . . .	310
v1.9.0.	<code>\l@ddoxtrafeet</code> : Add <code>\fnpos</code> to choice the order of footnotes. . . . .	274
	<code>\l@dfeetendmini</code> : Add <code>\mpfnpos</code> to choice the order of footnotes in minipage / ledgroup. . . . .	306
v1.10.0.	General: Add <code>\pstartref</code> and <code>\xpstartref</code> to refer to a pstart number (extension of <code>\edlabel</code> ). . . . .	1
	<code>\endquote</code> : Correction of sectioning commands in parallel texts. . . . .	360
v1.10.1.	General: Compatibility with cleveref. . . . .	1
v1.10.2.	General: Compatibility of stanza with v1.8a of babel-greek. . . . .	1
v1.10.3.	General: Debug of cross-referencing. . . . .	1
v1.10.4.	General: Debug of critical notes in edtabular environment. . . . .	1
v1.10.5.	General: Debug of <code>\pausenumbering</code> . . . . .	1
	Debug of <code>\xxref</code> . . . . .	1
v1.10.6.	General: Debug of interaction between <code>\autopar</code> and <code>\pausenumbering</code> . . . . .	1
v1.11.0.	General: Add hooks to disable the font selection for lemma in footnote. . . . .	48
v1.11.1.	General: Correct a bug when a critical note starts with plus or minus. . . . .	1
v1.12.0.	General: Add <code>\ledinnernote</code> and <code>\ledouternote</code> commands. . . . .	64
	Add <code>\Xendparagraph</code> and related settings. . . . .	53
	Add hyperlink to crossref (needs <code>hyperref</code> package). . . . .	59
	Compatibility with <code>musixtex</code> . . . . .	1
	Debug eledmac sectioning command after using <code>\resumenumbering</code> . . . . .	1
	Ensure that <code>imakeidx</code> is loaded <i>before</i> eledmac . . . . .	310
	New hooks: <code>\Xafterrule</code> and <code>\afterruleX</code> . . . . .	52
	New options for ragged-paragraph notes . . . . .	51
	New sectioning commands. . . . .	71
	Optional arguments for <code>\pstart</code> and <code>\pend</code> . . . . .	18
	<code>\@nl@reg</code> : To ensure compatibility with <code>musixtex</code> , <code>\@l</code> becomes <code>\@1</code> . Consequently, <code>\@l@reg</code> becomes <code>\@nl@reg</code> . . . . .	111
	<code>\AtStartEveryPstart</code> : New optional argument for <code>\pstart</code> , to execute code before it. . . . .	148
	<code>\edindex</code> : Use correctly default index when <code>imakeidx</code> is loaded. . . . .	314
	<code>\endquote</code> : <code>\ledxxx</code> sectioning commands are deprecated and replaced by <code>\eledxxx</code> commands. . . . .	360
	<code>\initnumbering@reg</code> : <code>\beginnumbering</code> is defined only on eledmac, not on eledpar. . . . .	92
	<code>\l@dgetsidenote@margin</code> : <code>\sidenotemargin</code> is now directly defined in eledmac to be able to manage eledpar. . . . .	299
	<code>\l@disnote</code> : <code>\l@dlsnote</code> , <code>\l@drsnote</code> and <code>\l@dcsnote</code> defined only one time, in eledmac, including needs for eledpar case. . . . .	300
	<code>\l@dnumpstartsL</code> : Add <code>\ifledRcol@</code> for eledpar . . . . .	90
	<code>\l@dunhbox@line</code> : <code>\do@line</code> is split in more little commands. . . . .	154
	<code>\newhookcommand@series@reload</code> : Debug <code>\beforenotesX</code> and <code>\maxhnotesX</code> which did not work when called after <code>\footparagraphX</code> . . . . .	267

Debug <code>\Xbeforenotes</code> and <code>\Xmaxhnotes</code> which did not work when called after <code>\footparagraph</code> .	267
<code>\pend</code> : New optional argument for <code>\pend</code> , to execute code after it.	150
<code>\stanza</code> : & can have an optional argument: content to be printed after.	327
<code>\Stanza</code> can have an optional argument: content to be printed before.	327
Add <code>\newverse</code> macro, <code>\falseverse</code> deprecated.	327
v1.12.1.	
<code>\wrap@edcrossref</code> : Fix spurious spaces.	287
v1.12.2.	
<code>\l@dunhbox@line</code> : Fix a bug with critical notes at the tops of pages (added by v12.0.0)	154
v1.12.3.	
General: Add macros for new messages since v0.7	82
Correct bug with side and familiar notes in tabular environments.	1
Debug <code>\eledxxx</code> with some paper size	1
Debug <code>\ledinnernote</code> and <code>\ledouternote</code> commands in the top of pages.	64
Debug left and right notes (bugs added by 1.12.0)	1
Underline lemma in <code>\eledxxx</code> when using draft mode.	1
<code>\flag@end</code> : <code>\flag@start</code> and <code>\flag@end</code> are now defined only one time for <code>eledmac</code> and <code>eledpar</code>	126
<code>\flag@start</code> send a error message when a <code>\edtext</code> is done without insert (note)	126
<code>\reledmac@error</code> : Replaced error messages	82
v1.12.4.	
General: Debug spurious page breaks before <code>\chapter</code> (bug added in 1.12.0)	1
v1.12.5.	
<code>\edindex@hyperref</code> : Debug <code>\edindex</code> when <code>hyperref</code> is not loaded	316
<code>\essect</code> : Debug <code>\eledchapter</code> in parallel with <code>memoir</code>	364
<code>\doinsidelinehook</code> : Added <code>\dolinehook</code> and <code>\doinsidelinehook</code>	157
<code>\endnumbering</code> : Allow to mix parallel columns and normal text when using <code>\pausenumbering</code>	93
<code>\l@dobbleoptarg</code> : <code>\l@dobblearg</code> becomes <code>\l@dgobbeloptarg</code>	345
<code>\l@drestoreforedtext</code> : Debug optional arguments of <code>\Xfootnote</code> in tabular context	345
<code>\pausenumbering@page@num</code> : Debug <code>\resumenumbering</code>	95
v1.12.7.	
<code>\wrap@edcrossref</code> : <code>\wrap@edcrossref</code> is now robust	287
v1.12.8.	
<code>\flag@end</code> : <code>\flag@start</code> do not send a error message when a <code>\edtext</code> is done without insert (note) but have a endnote	126
v1.13.0.	
General: Add <code>\Xnoteswidthliketwocolumns</code> and <code>\noteswidthliketwocolumnsX</code>	53
Added <code>widthliketwocolumns</code> option	78
<code>\newhooktoggle@series@reload</code> : Add <code>\newhookcommand@toggle@reload</code>	267
<code>\para@footsetupX</code> : In <code>\para@footsetupX</code> , use <code>\columnwidth</code> instead of <code>\hsize</code>	228
<code>\settoggle@series</code> : <code>\settoggle@series</code> can take an optional arguments to reload series setup.	265
v1.13.1.	
General: Coming back of page and line breaking penalties's management, deleted by error in v0.17.	1
Debug quotation environment inside of a <code>\pstart</code> preceded by a sectioning command.	1
<code>\ifat@every@pstart@star@</code> : Add <code>\l@dzeropenalties</code> in <code>\pstart</code>	148
<code>\print@line</code> : Coming back of <code>\add@penalties</code> , mistakenly deleted in version 0.17.	155
v1.13.2.	
General: Fix a bug with normal footnotes, added by v1.13.0.	1
<code>\l@dnumpstartsL</code> : Add <code>\ifl@dpadding</code> for <code>eledpar</code>	90

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.	
General: Debug spurious characters before endnotes. . . . .	237
Delete previous override of <code>\l@d@@wrindexhyp</code> at the beginning of a document when hyperref is not loaded. . . . .	318
Move gobbling command . . . . .	89
Provide <code>\@gobblefour</code> . . . . .	89
<code>\edindex</code> : Let <code>eledmac</code> take advantage of <code>imakeidx</code> even when memoir class is used . . . . .	314
v1.14.1.	
<code>\@ssect</code> : Debug sectioning commands when using both <code>handout</code> and <code>hyperref</code> package. . . . .	366
v1.14.2.	
<code>\@ssect</code> : Debug <code>\edtext</code> after starred sectioning commands when using <code>memoir</code> class. . . . .	364
v1.15.0.	
General: Fix a bug with footnotes layout when using some options of the geometry package (bug add by v1.13.0). . . . .	1
New commands <code>\AtEveryPstart</code> and <code>\AtEveryPend</code> . . . . .	18
New tools to prevent ambiguous references in lemma . . . . .	31
<code>\@edtext@level</code> : New boolean <code>\if@edtext@</code> . . . . .	133
<code>\arrangementX@paragraph</code> : Correct bug with paragraphed familiar footnotes setting. . . . .	228
<code>\endsub</code> : Restore subline feature (disabled by mistake in v1.8.0). . . . .	128
<code>\if@lemmacommand@</code> : New boolean <code>\iflemmacommand@</code> . . . . .	140
v1.15.1.	
<code>\next@line@list@stuff</code> : Revert modification of 1.5.2 which makes bug with numbering. Leave vertical mode to solve spurious space before minipage. . . . .	124
v1.16.0.	
General: <code>\edtext</code> is now defined only in <code>eledmac</code> , not in <code>eledpar</code> . Debug wrong numbering when using <code>\sameword + eledpar + \tag</code> command. . . . .	133
Compatibility of standard footnotes with some biblatex styles. . . . .	1
New <code>\stanzaindent</code> command. . . . .	1
v1.16.1.	
<code>\xlineref</code> : <code>\lineref</code> is not defined if defined by some other package, like <code>lineno</code> . <code>Eledmac</code> provides <code>\edlineref</code> instead. . . . .	288
v1.17.0.	
<code>\edtext</code> : Error message when calling <code>\edtext</code> outside of a numbered paragraph. . . . .	134
v1.18.0.	
General: Add <code>\Xpstarteverytime</code> . . . . .	44
Compatibility with Lua <sup>ℒ</sup> TeX RTL languages. . . . .	1
Debug <code>\Xonlypstart</code> when using <code>\Xnumberonlyfirstinline</code> and the current line number differs from the previous. . . . .	44
<code>\@edindex@hyperref</code> : Fix spurious space with <code>\edindex</code> when using <code>imakeidx/indextools + hyperref</code> . . . . .	316
<code>\edlabel</code> : <code>\edlabel</code> is now defined only one time for both <code>eledmac</code> and <code>eledpar</code> . . . . .	282
<code>\l@d@section</code> : Option <code>parapparatus</code> works for endnotes. . . . .	238
<code>\l@dnumpsstartsL</code> : Add <code>\ifl@dprintingpages</code> and <code>\@dprintingcolumns</code> for <code>eledpar</code> . . . . .	90
<code>\print@line</code> : Compatibility with Lua <sup>ℒ</sup> TeX RTL languages. . . . .	155
<code>\printlinefootnote</code> : Code refactoring in <code>\printlinefootnote</code> : the printing of the numbers are factorized in <code>\printlinefootnotearea</code> . . . . .	201
<code>\printpstart</code> : Debug <code>\Xpstart</code> with parallel pages and columns ( <code>eledpar</code> ) . . . . .	199

v1.19.0.	
General: <code>\Xmaxhnotes</code> and <code>\maxhnotesX</code> work now for both two-columns and three-columns setting. . . . .	1
Compatibility with <code>eledpar</code> v.1.13.0. . . . .	1
<code>\footsplitskips</code> : <code>\footsplitskips</code> doesn't set <code>\floatingpenalty</code> to <code>\@MM</code> when processing parallel pages. . . . .	179
<code>\xxref</code> : <code>\xxref</code> works also with right side numbers, when <code>\@Rlineflag</code> is not empty. . . . .	290
v1.19.1.	
General: Call <code>\correct@footinsX@box</code> and <code>\correct@Xfootins@box</code> directly in <code>\print@notesX@forpages</code> and <code>\print@Xnotes@forpages</code> , that is in <code>eledpar</code> . . . . .	1
v1.20.0.	
General: Add <code>\Xendboxlinenum</code> . . . . .	47
Add <code>\Xtwolines</code> and <code>\Xmorethantwolines</code> hooks . . . . .	43
Add series option. . . . .	1
Correct <code>\Xinplaceofnumber</code> hook. . . . .	1
Explicit error message when calling <code>\Xfootnote</code> outside of <code>\edtext</code> . . . . .	1
Fix a bug with line number typesetting direction when using <code>\eledsection</code> and similar commands for RTL texts with <code>Lua<math>\TeX</math></code> . . . . .	1
Fix issues with RTL text in notes when using <code>Lua<math>\TeX</math></code> . . . . .	1
Options fulllines in <code>\Xfootnote</code> . . . . .	28
The <code>\newifs</code> are not followed by boolean values set to false, because it is the $\TeX$ default setting. . . . .	1
<code>\printlines@count</code> : Added <code>\ifl@d@Xmorethantwolines</code> and <code>\ifl@d@Xmorethantwolines</code> to <code>\printlines</code> . . . . .	211
<code>\stanza</code> : <code>&amp;</code> and <code>\&amp;</code> can be preceded by spaces. . . . .	327
<code>\xxref</code> : Debug <code>\xxref</code> when not loading <code>eledpar</code> (Fix a bug added in 1.19.0). . . . .	290
v1.21.0.	
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 . . . . .	263
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 . . . . .	50
Add <code>\Xendtwolines</code> , <code>\Xendmorethantwolines</code> , <code>\Xendtwolinesbutnotmore</code> and <code>\Xendtwolinesonlyinsamepage</code> . . . . .	43
Add <code>\Xparindent</code> and <code>\hangindentX</code> . . . . .	49
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>noledsec</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</code> 1.0.0). . . . .	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>oldprintnnumspace</code> option allows to come back to previous setting	1
<code>parapparatus</code> option works now with familiar footnotes.	1
Provide <code>\@gobblefive</code>	89
<code>\@edindex@hyperref</code> : Look at the hyperindex option of <code>hyperref</code> before inserting <code>hyperref</code>	316
<code>\l@d@section</code> : <code>\endnotes</code> take five arguments.	238
<code>\ledinnotemark</code> : Add <code>\ledinnotemark</code> .	313
<code>\ledsetnormalparstuffX</code> : <code>\ledsetnormalparstuff</code> is deprecated and becomes <code>\ledsetnormalparstuffX</code> and <code>\Xledsetnormalparstuff</code> .	233
<code>\n@num</code> : <code>\n@num@ref</code> deleted	119
<code>\n@num</code> defined only one time for both <code>Eledmac</code> and <code>Eledpar</code>	119
<code>\newhookcommand@series</code> : <code>\newhookcommand@series</code> can take an optional argument.	266
<code>\newhooktoggle@series</code> : <code>\newhooktoggle@series</code> can take an optional argument.	267
<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>	235
<code>\seriesatend</code> : Fix spurious space in <code>\seriesatend</code>	264
<code>\skipnumbering</code> : <code>\skipnumbering</code> defined only one time for both <code>Eledmac</code> and <code>Eledpar</code> .	130
Correct <code>\skipnumbering</code> for stanza.	130
Delete <code>\skipnumbering@reg</code> .	130
v1.22.0.	
General: Add <code>\doendnotesbysection</code> command.	29
Add option for lemma separator inside endnotes	47
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.	29
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
<code>\ledinnote</code> : <code>\ledinnote</code> takes a first optional argument, which is the label for hyperlinks.	313
v1.22.1.	
General: Fix a bug (added on v1.22.0) with <code>\Xinplaceofnumber</code> hook.	1
<code>\prevpage@num</code> : Correct double symbol when using both <code>\parafootsep</code> and <code>\Xsymlinenum</code> .	190
v1.23.0.	
General: Add <code>\Xboxlinenumalign</code> and <code>\Xendboxlinenumalign</code> .	46
Add <code>\Xboxstartlinenum</code> , <code>\Xendboxstartlinenum</code> , <code>\Xboxendlinenum</code> , <code>\Xendboxendlinenum</code> .	46
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.	31
Fix a bug with <code>\sameword</code> when the same lemma is used for multiple notes or for nested <code>\edtexts</code> .	31
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.	31
<code>\@edtext@level</code> : The boolean <code>\if@edtext@</code> becomes the counter <code>\edtext@level</code> .	133

\lemma: Fix spurious space after \lemma command	139
\newseries@: Prevent spurious spaces when \Afootnote and similar commands are followed by spaces (bug added on 1.0.0).	254
\sameword: In order to allow use of \sameword with inputenc, we detokenize its mandatory argument before using it in control sequence names.	144
\SErefwithpage: Debug \Xendtwolines, \Xendmorethantwolines, \Xendtwolinesbutnotmore and \Xendtwolinesonlyinsamepage when using \apprefwithpage.	294
v1.23.1.	
General: Fix a bug with \lemma command in the right side.	1
v1.23.2.	
General: Compatibility with L <sup>A</sup> T <sub>E</sub> X's release 2015.	1
v1.24.0.	
General: We can reinitialize \AtEveryPstart and \AtEveryPend providing to it an empty argument.	1
v1.24.1.	
General: \lemma is disabled when using nocritical option.	1
v1.24.2.	
General: Fix incompatibility between unfamiliar option and memoir package.	1
v1.24.3.	
General: Restore marginal numbers and notes with sectioning command (bug introduced in v1.21.0)	1
v1.24.4.	
General: Fix spurious space with \edindex when using xindy+hyperref option.	1
v1.24.5.	
General: Fix a bug of indent, when a added in 1.1.0, when a \beginnumbering immediately follow a sectioning command.	1
v2.0.0.	
General: \@makecol, \@reinserts and \@docclearpage are patched instead of begin redefined	277
\doxtrafeeti becomes \Xdo@feet; \doxtrafeetii becomes \do@Xfeet; \@opxtrafeeti becomes \@opfeetX; \doreinxtrafeetii becomes \X@doreinfeet; \doreinxtrafeeti becomes \@doreinfeetX.	277
Add \Xendingplaceofnumber hook.	1
Add \Xendnonumber hook.	1
Add nonum 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 L <sup>A</sup> T <sub>E</sub> X's output macros are now patched and not override.	1
Package's name becomes reledmac.	1
Patch \@footnotemark instead of redefine it	215
Suppress indexing command specific to memoir.	314
\@iiiminipage: Patch \@iiiminipage instead of redefining it.	308
\@xympar: Patching \@xympar instead of redefining it	298
\endminipage: Patch \endminipage instead of redefining it.	308
\initnumbering@quote: \initnumbering@sectcmd becomes \initnumbering@quote	360
\l@advance@parledgroup@beforenormalnotes: Some conde of \l@dumboxmpfoot moved to \l@advance@parledgroup@beforenormalnotes	308
\newseries@: One endnotes file by series.	260
v2.0.1.	
General: Fix a bug in eledmac-compat option	1
Fix incompatibility between optional argument of \pstart and \numberpstarttrue	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>\Xbhooknote</code> and <code>\bhooknoteX</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.	
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
<code>\ledsetnormalparstuffX</code> : Replaced <code>\noindent</code> with <code>\parindent</code> set to 0pt.	233
v2.2.0.	
General: Fix a bug with combination of <code>\onehalfspacing</code> and two columns and three columns notes typeset.	1
Fix a bug with some setting command and optimization option.	1
Fix spurious space with paragraphed critical notes when using $\text{Lua}\TeX$ .	1
Increase line list version number to ensure compatibility with new options of <code>reledpar</code> package.	1
New setting tools for endnotes: <code>\Xendnumberonlyfirstinline</code> , <code>\Xendnumberonlyfirstintwolines</code> , <code>\Xendsymmlinenum</code> , <code>\Xendbeforenumber</code> , <code>\Xendafterrenumber</code> , <code>\Xendbeforemsymmlinenum</code> , <code>\Xendaftersymmlinenum</code> , <code>\Xendboxsymmlinenum</code> , <code>\Xendhangindent</code> , <code>\Xendbhooklinenum</code> , <code>\Xendahooklinenum</code> , <code>\Xendbhookinplaceofnumber</code> , <code>\Xendahookinplaceofnumber</code> .	1
v2.2.1.	
General: Compatibility with $\text{TeX}$ format 2015/10/01.	1
v2.2.2.	
General: Fix a bug in <code>\sethangingsymbol</code> .	1
Fix a bug with old version of <code>etex</code> .	1
v2.3.0.	
General: Disable empty lines as paragraph in stanza.	1
Fix incompatibility of paragraphed footnotes with <code>bidi</code> v17.9 and following.	1
Warning message when using some setting commands inside <code>rightside</code> environment (deprecated behavior)	1
v2.3.1.	
General: Fix spurious space when using optional argument of <code>\stanza</code> (introduced in v2.3.0).	1
v2.4.0.	
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 stanza (introduced in v2.3.0). . . . .	1
New hooks: <code>\Xlinerangeseparator</code> and <code>\Xendlinrangeseparator</code> . . . . .	43
Option <code>linrangeseq</code> for critical footnotes and endnotes. . . . .	43
<code>\footnoteoptions@</code> : First argument of <code>\footnoteoption@</code> is now mandatory, not optional. . . . .	176
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 $\LaTeX$ <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 $\LaTeX$ <code>\roman</code> command, like Greek. . . . .	1
v2.5.0.	
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>linrangeseq</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>\Xbhookgroup</code> and <code>\bhookgroupX</code> . . . . .	1
New tools to easily make cross-reference to a passage defined by a start and an end line	61
<code>\edlabel</code> : Fix a bug when calling <code>\edlabel</code> in a footnotes of the right side . . . . .	282
<code>\l@d@section</code> : <code>\endnotes</code> take six arguments. . . . .	238
<code>\printlines@count</code> : <code>\printlines</code> takes an eighth argument: the line flag . . . . .	210
<code>\SErefwithpage</code> : Debug <code>\setapprefprefixsingle</code> . . . . .	294
<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. . . . .	288
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.	
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> . . . . .	53
New hooks: <code>\Xlemmafnt</code> and <code>\Xendlemmafnt</code> . . . . .	48
New setting commands: <code>\setSErefonlypageprefixsingle</code> and <code>\setSErefonlypageprefixmore</code> . . . . .	1
Warning for duplicate and undefined labels are parsable by <code>latexmk</code> . . . . .	1

Warning for duplicate labels does not send any more a false line and page number . . .	1
When using <code>hyperref</code> package, add link in familiar footnotes between the footnote marks in the text and the footnote marks in the footnote . . . . .	1
When using <code>hyperref</code> package, add links for <code>\SEref</code> and related, <code>\appref</code> and related. . . . .	1
When using <code>hyperref</code> package, add links from critical footnotes and critical endnotes to the line of text they refers . . . . .	1
<code>\@k: \rigidbalance</code> is split in <code>\Xrigidbalance</code> and <code>\rigidbalanceX</code> . . . . .	191
<code>\l@d@section: \endnotes</code> take seven arguments. . . . .	238
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> . . . . .	53
Fix problem of hyphenation when using <code>hyperref</code> package (added in v2.7.0). . . . .	1
v2.8.0.	
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
<code>\l@d@section: \Xendhangindent</code> and <code>\Xendafternote</code> can take values which are relative to the font size of the endnote. . . . .	238
v2.8.1.	
General: Warnings for undefined labels are really parsable by <code>latexmk</code> . . . . .	1
v2.8.2.	
General: Fix a bug concerning <code>indent</code> in a paragraph immediately following a sectioning command (bug NOT fixed on <code>reledpar</code> ) . . . . .	1
Fix a bug with <code>\AtEveryPstart</code> added in version 2.0.0. . . . .	1
Fix a bug with vertical space after the between-sectioning command as optional argument of a <code>\pstart</code> and <code>\pstart</code> content . . . . .	1
v2.9.0.	
General: Allow continuing line numbering between normal text and parallel text, using <code>\pausenumbering</code> and <code>\resumenumbering</code> and the <code>continuousnumberingwithcolumns</code> option. . . . .	1
Fix a bug when using <code>\lineneation{page}</code> and <code>\pausenumbering</code> . . . . .	1
Fix a bug with three- and two-column footnote setting (added in v.2.4.0). . . . .	1
Fix spurious space inside three-column familiar footnote. . . . .	1
Write correct metadata in numbered files when using <code>\pausenumbering</code> . . . . .	1
v2.9.1.	
General: Fix a bug when notes start with <code>plus</code> or <code>minus</code> . . . . .	1
v2.9.2.	
General: Fix a bug with <code>hyperref</code> package when a lemma starts with <code>plus</code> 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.	
General: Add <code>\AtEveryStanza</code> and <code>\AtEveryStopStanza</code> . . . . .	1
Fix a bug in <code>\ledlsnotefontsetup</code> and <code>\ledrsnotefontsetup</code> which could not handle <code>\color</code> command properly. . . . .	1
More specific error messages. . . . .	1
New hooks: <code>\Xwrapcontent</code> , <code>\Xendwrapcontent</code> and <code>\wrapcontentX</code> . . . . .	49
New hooks: <code>\Xwraplemma</code> and <code>\Xendwraplemma</code> . . . . .	48

\print@lemma: Code refactoring between \parafootfmt, \twocolfootfmt, \threecolfootfmt and \normalfootfmt. . . . .	200
v2.10.1.	
General: Add nopenalalties 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.	
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
\do@actions@fixedcode: Add action 1010 . . . . .	162
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 noledsec 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 $\TeX$ code . . . . .	1
v2.13.0.	
General: Version 2.13.0 never existed. . . . .	1
v2.13.1.	
General: In critical footnotes, the right side flag is printed only if requested explicitly with \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 \footnoteX in the first argument of \edtext. . . . .	1
v2.14.1a.	
General: Fix problematic typos in the handbook. . . . .	1
v2.15.0.	
General: Add “byline” arrangement. . . . .	1
Fix \Xtxtbeforenotes in ledgroup. . . . .	1
v2.15.1.	
General: Fix \edindex in tabular environments. . . . .	1
v2.15.2.	
General: Fix a bug with fancyhdr 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>	277	
Add <code>\Xendpagenumberonlyfirst</code> , <code>\Xendpagenumberonlyfirstifsingle</code> , <code>\Xendpagenumberonlyfirstintwo</code> , <code>\Xendinplaceofpagenumber</code> and <code>\Xendsympagenum</code> hooks.	1	
Add <code>\Xpagelinesep</code> hook.	45	
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 <code>notenum</code> 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.	277	
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... \endnumbering</code> .	1	
Fix a bug with <code>\numberpstarttrue</code> when using multiple <code>\beginnumbering... \endnumbering</code> .	1	
v2.16.8.		
General: Fix a bug with <code>\edindex</code> in footnotes in parallel typesetting.	1	
v2.17.0.		
General: Add <code>\edglsadd</code> command.	1	

Add <code>\setmsdataposition</code> setting. . . . .	1
v2.17.1.	
General: Fix spurious space in paragraphed footnotes when using Lua $\TeX$ 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 <i>latexmk</i> . . . . .	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 $\TeX$ . . . . .	1
Fix a bug when using both <code>\Xnumberonlyfirstintwolines</code> or <code>\Xnumberonlyfirstinline</code> and <code>\Xparafotsep</code> and <code>\Xsymlinenum</code> . . . . .	1
v2.18.1.	
General: Fix a bug when using <code>\msdata</code> with Lua $\TeX$ or with the <code>hyperref</code> package. . . . .	1
v2.19.0.	
General: Add <code>\footnotemark</code> and <code>\footnotetext</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.	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
	<code>\next@line@list@stuff</code> : Add <code>\next@line@list@stuff</code> internal hook. . . . .	123
v2.24.1.	General: Compatibility with <code>reledpar 2.20.2</code> . . . . .	1
v2.24.2.	General: Fix bug between tabular environments and endnotes . . . . .	1
v2.25.0.	General: Add <code>\Xendtxtbeforenotes</code> hook . . . . .	1
v2.26.0.	General: Add a mechanism for annotation of stanza line numbering . . . . .	1
	Fix bug in <code>\xpstartref</code> . . . . .	1
v2.26.1.	General: Fix bug with <code>\linenumannotation</code> . . . . .	1
v2.26.10.	General: Add error message when insertion a <code>\pstart</code> or <code>\pend</code> inside <code>\edtext</code> arguments . . . . .	1
	Really fix bug with page number in footnote after <code>\mainmatter</code> (bug introduced in v2.26.7) . . . . .	1
v2.26.11.	General: Allow to use <code>\edtext</code> inside <code>\sameword</code> . . . . .	1
v2.26.12.	General: Precise minimal version of <code>xparse</code> package required. . . . .	1
v2.26.2.	General: Fix bug with <code>\linenumannotation</code> in parallel typesetting . . . . .	1
	Fix bug with <code>\Xgroupbyline</code> and <code>\Xparafootsep</code> . . . . .	1
	Fix bug with <code>\Xgroupbyline</code> and optional argument of critical footnotes. . . . .	1
	Fix bug with <code>\Xgroupbyline</code> for the series after the first one. . . . .	1
v2.26.3.	General: Fix bug added by v2.26.0 with notes in parallel typesetting . . . . .	1
v2.26.4.	General: Fix bug when using <code>\numberpstarttrue</code> and alternating between manual <code>\pstart</code> and automatic <code>\pstart</code> produced by <code>\autopar</code> . . . . .	1
v2.26.5.	General: Check that users have a enough recent version of <code>xparse</code> . . . . .	1
	Fix bug when using <code>\eledsection</code> with the <code>continuousnumberingwithcolumns</code> of <code>reledpar</code> . . . . .	1
	Fix bug with <code>continuousnumberingwithcolumns</code> . . . . .	1
	Fix bug with using <code>\eledsection</code> and related when alternate normal typesetting and parallel typesetting . . . . .	1
v2.26.6.	General: Fix bug with <code>\lineation</code> by page and nexted <code>\edtext</code> between two pages (added in v. 2.24.0) . . . . .	1
v2.26.7.	General: Fix page number in footnote and endnotes when using <code>sameparallelpagenumber</code> option of <code>reledpar</code> and typesetting critical text in single mode after critical text in double page mode . . . . .	1
v2.26.8.	General: Fix bug in some specific cases, introduced by v.2.27.7 . . . . .	1

v2.26.9.		
General: Fix bug with critical footnotes line number when alternating between <code>\numberlinefalse</code> and <code>\numberlinetrue</code> .	1	1
Fix bug with page number in footnote after <code>\mainmatter</code> (bug introduced in v2.26.7)	1	1
Workaround with a bug of $\TeX$ 2018/04/01 patch level 3 to generate the <code>.sty</code> file from a <code>.ins</code> file	1	1
v2.27.0.		
General: Add <code>\doinsidethislinehook</code> .	1	1
v2.27.1.		
General: Fix spurious space in <code>\hidenumbering</code> .	1	1
v2.28.0.		
General: Familiar footnotes can take an optional argument to manually define the footnote mark.	1	1
v2.28.1.		
General: Fix bug with the alignment of the columns of two and three columns critical footnotes.	1	1
v2.29.0.		
General: Add <code>noresetlinenumannotation</code>	1	1
Fix bug with <code>\linenumannotation</code> and <code>nofamiliar</code> option.	1	1
Fix little bug with <code>\msdata</code>	1	1
Really reset line number annotation at each line	1	1
v2.30.0.		
General: Add <code>\Xnoidenticallylinenumannotation</code> and <code>\Xendnoidenticallylinenumannotation</code> hooks	1	1
Add separator between line number annotations in margin. Add <code>\setlinenumannotationsep</code>	1	1
v2.31.0.		
General: Add <code>noprefix</code> and <code>prefixmore</code> to <code>\appref</code> and <code>\SEref</code> .	1	1
v2.31.1.		
General: Fix bug in vertical spacing with <code>\Xgroupbyline</code> .	1	1
Fix bug with <code>\Xgroupbyline</code> , which could produce spurious blank lines.	1	1
Fix bug with <code>\Xhangindent</code> and <code>\hangindentX</code> for right-to-left texts.	1	1
Fix bug with indentation with <code>\Xgroupbyline</code> .	1	1
Fix bug with multiple footnote layer and <code>\Xgroupbyline</code> .	1	1
Fix incompatibility between <code>\Xgroupbyline</code> and <code>hyperref</code> package.	1	1
Fix incompatibility between <code>\Xgroupbyline</code> and optional argument of critical footnotes.	1	1
v2.31.2.		
General: Fix bug with familiar footnote containing non-default font (introduced with v. 2.20.0).	1	1
v2.31.3.		
General: Fix bug with paragraphed footnote in RTL context.	1	1
v2.32.0.		
General: Add <code>\Xlinenumannotationonlyfirst</code> , <code>\Xlinenumannotationonlyfirstintwo</code> , <code>\Xsymlinenumannotation</code> , <code>\Xendlinenumannotationonlyfirst</code> , <code>\Xendlinenumannotationonlyfirstintwo</code> and <code>\Xendsymlinenumannotation</code> .	1	1
v2.32.1.		
General: Do not print <code>\Xendsep</code> if we print <code>\Xendsymlinenum</code> .	1	1
When using <code>\parindentX</code> or <code>\Xparindent</code> and stanza, use the normal indentation in the footnote, not the verse indentation.	1	1
v2.32.2.		
General: Fix spurious vertical space in familiar footnotes with $\text{\LaTeX}$ .	1	1

v2.32.3.	General: Fix bug with critical footnotes when typesetting Arabic text with polyglossia.	1
v2.32.4.	General: <code>\footnoteXmark</code> and <code>\footnoteXtext</code> can now have an optional argument, setting the footnote number.	1
	Fix bug <code>\footnoteXmark</code> and <code>\footnoteXtext</code> .	1
	Fix bug when using <code>\Xgroupbyline</code> and twocol arrangement.	1
	Fix bug with three and two columns footnotes when using <code>bidi</code> .	1
v2.32.5.	General: Fix bug with <code>\edindex</code> inside <code>\eledsection</code> and related.	1
	Fix bug with macro inside <code>\sameword</code> with no- $X_{\text{L}}\text{TeX}$ engines.	1
	Fix bug with page number in index with <code>\edindex</code> when indexing a critical notes.	1
v2.32.6.	General: Add hyperref link when inserting glossary entries.	1
	Fix bug with glossary entries inside the first argument of <code>\edtext</code> .	1
v2.32.7.	General: Fix bug with <code>\lineation{pstart}</code> and <code>\edtext</code> in the very first line of a <code>\pstart</code> .	1
	Fix bug with <code>\parafootsepX</code> .	1
	Fix spurious spaces at the very first line of <code>edtabular</code> environments.	1
v2.32.8.	General: Add subline number in manuscript apparatus.	1
v2.33.0.	General: Add <code>\Xinnotemark</code> and <code>\innotemarkX</code> hooks.	1
	Add antilabe feature.	1
	Add <code>startstanzaindentsrepetition</code> feature.	1
	<code>\ledinnotemark</code> : <code>\ledinnotemark</code> redefinition is deprecated, use <code>\Xinnotemark</code> and <code>\innotemarkX</code> instead.	313
	<code>\newhookcommand@series</code> : <code>\newhookcommand@series</code> can take a second optional argument (third argument).	266
v2.33.1.	General: Fix a spacing bug when the argument of <code>\antilabe</code> contains more than one word.	1
v2.34.0.	General: Add <code>\Xpstartonlyfirst</code> and <code>\Xstanzaonlyfirst</code> .	1
	Add <code>\Xpstartseparator</code> .	1
v2.34.1.	General: Fix bug with page number when using <code>stanazindentsrepetition</code> , bug introduced in v2.33.0.	1
v2.34.2.	General: Fix bug with tabular environment, in some specific use case, added in v2.32.7.	1
v2.34.3.	General: Fix bug with <code>\parafootsepX</code> when the footnotes are called outside of a numbered section.	1
	Fix incompatibility between <code>lineation</code> by <code>\pstart</code> and (a)stanza.	1
	Make commands like <code>\hidenumbering</code> working at the very beginning of a verse when the first argument of <code>\setstanzaindents</code> is equal to 0.	1
v2.34.4.	General: Fix bugs with <code>\parafootsepX</code> outside of numbered section.	1
	Fix some bugs with <code>lineation</code> by page, when a numbered section starts at the very beginning of page.	1
v2.34.5.	General: Fix incompatibility between <code>\apprefwithpage</code> / <code>\SErefwithpage</code> and line number annotation mechanism.	1

v2.34.6.	General: Fix incompatibility with calc package (added with v2.34.4). . . . .	1
v2.34.7.	General: Fix bug with \parafootsepX in some circumstances (added with v2.34.4). . . . .	1
v2.35.0.	General: Add \linenumannotationothersidettrue. . . . .	1
	Add new hooks: \Xnolinenumber, \Xendnolinenumber, \Xnolinenumberifannotation, \Xendnolinenumberifannotation. . . . .	1
v2.35.1.	General: Fix a bug when printing ending page number in endnotes and using the hyperref package. . . . .	1
v2.36.0.	General: Add \Xnonote and \nonoteX. . . . .	1
	Fix bug with sublineation on right side of parallel typesetting. . . . .	1
v2.36.1.	General: Fix bug with crossreferencing in parallel typesetting (added in v2.34.4) . . . . .	1
v2.37.0.	General: Fix bug with \Xnonote . . . . .	1
	Fix incompatibility between \eledsection (and related) and hyperref. Incompatibility was introduced by some changes in hyperref package. . . . .	1
	Improve syntaxis check. . . . .	1
v2.37.1.	General: Fix bug with \edgls and related in \edtext argument. Require glossaries-extra is glossaries is loaded. . . . .	1
v2.37.2.	General: Fix bug with \Xgroupbyline and two or three columns notes. . . . .	1
	Fix bug with \Xstanzaonlyfirst. . . . .	1
	Fix bug with parapparatus option and nested \edtext. . . . .	1
v2.38.0.	General: Add \Xbeforepagenumber hook. . . . .	1
	Add \Xnopagenumberifcurrent hook. . . . .	1
	Add hyperlinks between footnote and lemma when using abbreviated line range. . . . .	1
	Fix bug with \lemma and right-to-left typesetting with Xe <sub>La</sub> TeX. . . . .	1
	Fix indent bugs with stanza and right-to-left typesetting. . . . .	1
v2.38.1.	General: Fix incompatibility between reledpar's option continuousnumberingwithcolumns and lineation by page. . . . .	1
v2.38.2.	General: Fix bug when using \applabel in conjunction with hyperref package. . . . .	1
	Fix some trouble in indentation in stanza, when switching to some language with polyglossia. . . . .	1
v2.39.0.	General: Add \Xnotboxingsubline and \Xendnotboxingsubline hooks. . . . .	1
v2.39.1.	General: Fix bug with index in familiar and critical footnotes. . . . .	1
v2.39.2.	General: Fix bug with \edlabel in \eledsection and related. . . . .	1
	Fix bug with with continuousnumberingwithcolumns when right column starts a new page and we use lineation by page. . . . .	1
	Tools for reledpar v2.25.4 . . . . .	1
v2.39.3.	General: Fix bug with \edlabel after tabular environment (added by v2.39.2). . . . .	1
v2.39.4.	General: Fix bug with hyperlink when using \footnoteXmark and \footnoteXtext. . . . .	1

v2.39.5.	
General: The fix in v2.39.4 did'nt work well in tabular environment. . . . .	1
v2.39.6.	
General: Fix compatibility with memoir v3.8. . . . .	1
v2.39.7.	
General: Again fix with hyperref, footnote and tabular environment . . . . .	1
Fix compatibility with \pagenumbering of memoir v3.8. . . . .	1