

# 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 v2.40.2, last revised 2025/01/07.

<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> . . . . .	18
5.2.3 Content before specific <code>\pstart</code> and after specific <code>\pend</code> . . . . .	18
5.2.4 Content before every <code>\pstart</code> and after every <code>\pend</code> . . . . .	19
5.2.5 Numbering paragraphs ( <code>\pstart</code> ) . . . . .	19
5.2.6 Languages written in Right to Left . . . . .	19
5.2.7 Memory limits . . . . .	20
5.3 Lineation commands . . . . .	20
5.3.1 Disabling lineation . . . . .	20
5.3.2 Setting lineation start and step . . . . .	21
5.3.3 Setting lineation reset . . . . .	21
5.3.4 Setting line number margin . . . . .	21
5.3.5 Other settings . . . . .	22
5.4 Changing the line numbers . . . . .	22
5.4.1 Sublineation . . . . .	22
5.4.2 Locking lineation . . . . .	22
5.4.3 Setting and changing line number . . . . .	23
5.4.4 Line number style . . . . .	23
5.4.5 Skipping and hiding number . . . . .	23
5.5 Adding annotations to line numbers . . . . .	23
5.5.1 Resetting annotation . . . . .	25
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 . . . . .	26
5.5.5 Not printing the second identical annotations . . . . .	26
5.5.6 Separator between annotations . . . . .	26
5.5.7 Annotations in the apparatus . . . . .	27
5.6 Executing code at each line . . . . .	27
5.7 Executing code at a specific line . . . . .	27
<b>6 Apparatus commands</b>	<b>27</b>
6.1 Terminology . . . . .	27
6.2 Critical notes . . . . .	28
6.2.1 The lemma . . . . .	28
6.2.2 Footnotes . . . . .	28
6.2.3 Endnotes . . . . .	29
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	32
6.3.1 Basic use	32
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	33
6.3.5 Use with <code>\lemma</code> command	33
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	37
6.4.2 Layout of the apparatus of manuscripts	37
6.4.3 Settings	37
6.5 Familiar notes	38
6.5.1 Basic use	38
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	41
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	44
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	46
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	49
7.7.1 Wrapping lemmas	49
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	50
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 Printing only one footnote rule per page	52
7.12.6 Space after footnotes rules	52
7.12.7 Maximum height	53
7.12.8 Width	53
7.13 Footnotes and the <code>reledpar</code> columns	53
7.14 Endnotes in one paragraph	54
<b>8 Fonts</b>	<b>54</b>
<b>9 Verse</b>	<b>54</b>
9.1 Basic	54
9.2 Define stanza indents	55
9.3 Repeating stanza indents	55
9.4 Notes about the scope of the indent settings	56
9.5 Manual stanza indent	56
9.6 Stanza breaking	56
9.7 Hanging symbol	56
9.8 Long verse and page break	57
9.9 Content before/after verses	57
9.10 Numbering stanza	58
9.11 Antilabe	58
9.12 Various tools	59
9.13 Notes on empty lines	59
<b>10 Grouping</b>	<b>59</b>

<b>11 Cross referencing</b>	<b>59</b>
11.1 Basic use	60
11.2 Cross-referencing to a critical note	60
11.3 Cross-referencing which return a number in any case	60
11.3.1 Cross-referencing in order to define line number of a critical note	61
11.4 Not automatic cross-referencing	61
11.5 Normal $\LaTeX$ cross-referencing	61
11.6 References to start and end lines	62
11.6.1 Reference to main text lines	62
11.6.2 References to lines that are commented on in the apparatus	62
11.6.3 Settings	62
11.6.4 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>66</b>
13.1 Basics	66
13.2 Use with <code>imakeidx</code> or <code>indextools</code>	66
13.3 Referring to critical notes	66
13.4 Separator between page and line numbers	67
13.5 Using <code>xindy</code>	67
13.6 Advanced setting	68
<b>14 Glossary</b>	<b>68</b>
14.1 Preamble setting	68
14.2 Commands	68
<b>15 Tabular material</b>	<b>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	72
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 Loading packages	78
II.4 Package options	78
II.4.1 Options of reledpar	78
II.4.2 Options of reledmac	78
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	198
XIII.8 Footnote hanging indent	198
XIII.9 Critical notes presentation	198
XIII.9.1 Font tools	199
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>
XVIII.1 Printing only one footnote rule	235
XVIII.2 Dealing with multiple columns	235
<b>XIX Specific skip for first series of footnotes</b>	<b>236</b>
XIX.1 Overview	236
XIX.2 User level command	237
XIX.3 Internal commands	237
<b>XX Endnotes</b>	<b>238</b>
XX.1 Internal commands	238
XX.2 User level commands	242
XX.2.1 Inserting contents to endnotes	242
XX.2.2 Printing endnotes	243
<b>XXI Generate series of notes</b>	<b>251</b>
XXI.1 Test if series is still existing	252
XXI.2 Init specific to reledpar	252
XXI.3 For critical footnotes	252
XXI.3.1 Options	252
XXI.3.2 Create inserts, needed to add notes in foot	254
XXI.3.3 Create commands for critical apparatus, \Afootnote, \Bfootnote etc.	254
XXI.3.4 Set standard display	257
XXI.4 For familiar footnotes	257
XXI.4.1 Options	257
XXI.4.2 Create tools for familiar footnotes (\footnoteX)	258
XXI.5 The endnotes	261
XXI.5.1 The auxiliary file	261
XXI.5.2 The main macro	261
XXI.5.3 Tools	262
XXI.5.4 Internal commands	262



XXI.5.5 The options . . . . .	262
XXI.6 Init standards series (A,B,C,D,E) . . . . .	264
<b>XXII Setting series display</b>	<b>264</b>
XXII.1 Change series order . . . . .	264
XXII.2 Test series order . . . . .	265
XXII.2.1 Get the first series . . . . .	265
XXII.3 Series setting . . . . .	265
XXII.3.1 General way of working . . . . .	265
XXII.3.2 Tools to set options . . . . .	266
XXII.3.3 Tools to generate options commands . . . . .	267
XXII.3.4 Options for critical notes . . . . .	268
XXII.3.5 Options for familiar notes . . . . .	270
XXII.3.6 Options for endnotes . . . . .	271
XXII.4 Hooks for a particular footnote . . . . .	273
XXII.5 Alias . . . . .	275
<b>XXIII Output routine</b>	<b>275</b>
XXIII.1 Extra footnotes output . . . . .	275
XXIII.2 Patching standard output's commands . . . . .	278
<b>XXIV Page numbering in parallel typesetting</b>	<b>280</b>
<b>XXV Cross referencing</b>	<b>283</b>
XXV.1 Compatibility with xref . . . . .	298
<b>XXVI Sidenotes</b>	<b>299</b>
<b>XXVII Minipages and such</b>	<b>307</b>
<b>XXVIII Indexing</b>	<b>311</b>
XXVIII.1 Looking on package order . . . . .	311
XXVIII.2 Auxiliary macros for \edindex . . . . .	312
XXVIII.3 Code specific to \edindexin critical footnotes . . . . .	313
XXVIII.4 Analysis of command in indexed text . . . . .	314
XXVIII.5 Code for the formatted index . . . . .	314
XXVIII.6 Main code . . . . .	315
XXVIII.7 Hyperlink . . . . .	317
XXVIII.8 "innote" and "notenumber" option of indextols package . . . . .	319
<b>XXIX Glossaries</b>	<b>320</b>
<b>XXX Verse</b>	<b>323</b>
XXX.1 Hanging symbol management . . . . .	323
XXX.2 Using & character . . . . .	323
XXX.3 Code category setting . . . . .	324
XXX.4 Stanza count and indent . . . . .	324
XXX.5 Numbering stanza . . . . .	325
XXX.6 Stanza number in note . . . . .	326
XXX.7 Main work . . . . .	327
XXX.8 Restore catcode and penalties . . . . .	330

XXX.9 Dealing with antilabe . . . . .	330
<b>XXXI Apparatus of Manuscripts</b>	<b>331</b>
XXXI.1 User level macro . . . . .	331
XXXI.2 Setting macro . . . . .	332
XXXI.3 Counters and lists . . . . .	333
XXXI.4 Auxiliary file macros . . . . .	333
XXXI.5 Action macro . . . . .	334
XXXI.6 Inserting footnote . . . . .	339
XXXI.7 Other . . . . .	339
<b>XXXII Arrays and tables</b>	<b>340</b>
XXXII.1 Preamble: macro as environment . . . . .	340
XXXII.2 Tabular environments . . . . .	343
XXXII.2.1 Disabling and restoring commands . . . . .	343
XXXII.2.2 Counters, boxes and lengths . . . . .	347
XXXII.2.3 Tabular typesetting . . . . .	351
XXXII.2.4 Environments . . . . .	361
<b>XXXIII Quotation's commands</b>	<b>361</b>
<b>XXXIV Section's title commands</b>	<b>362</b>
XXXIV.1 Commands to disable some feature . . . . .	362
XXXIV.2 General overview . . . . .	363
XXXIV.3 <code>\beforeeledchapter</code> command . . . . .	363
XXXIV.4 Auxiliary commands . . . . .	364
XXXIV.5 Patching standard commands . . . . .	365
XXXIV.6 Main code of <code>\eledxxx</code> commands . . . . .	369
XXXIV.7 Macros written in the auxiliary file . . . . .	372
<b>XXXV Page breaking or no page breaking depending on specific lines</b>	<b>374</b>
<b>XXXVI Long verse: prevents being separated by a page break</b>	<b>375</b>
<b>XXXVII Tools for hyperref package</b>	<b>376</b>
<b>XXXVIII Compatibility with eledmac</b>	<b>377</b>
<b>Appendix A Things to do when changing versions</b>	<b>380</b>
A.1 Migrating from EDMAC to ledmac . . . . .	380
A.2 Migration from ledmac to eledmac . . . . .	381
A.3 Migration to eledmac 1.5.1 . . . . .	382
A.4 Migration to eledmac 1.12.0 . . . . .	382
A.5 Migration to eledmac 17.1 . . . . .	382
A.6 Migration to eledmac 1.21.0 . . . . .	383
A.6.1 <code>\Xledsetnormalparstuff</code> and <code>\ledsetnormalparstuffX</code> . . . . .	383
A.6.2 Endnotes . . . . .	383
A.7 Migration to eledmac 1.22.0 . . . . .	383
A.8 Migration to eledmac 1.23.0 . . . . .	383
A.9 Migration from eledmac to reledmac . . . . .	383
A.9.1 Risk of 'no room for a new' . . . . .	383

A.9.2 Multiple indices with memoir . . . . .	384
A.9.3 Deprecated commands and options . . . . .	384
A.9.4 <code>\renewcommand</code> replaced by <code>command</code> . . . . .	384
A.9.5 Commands the names of which have been changed . . . . .	385
A.9.6 Endnotes . . . . .	386
A.9.7 Z Series . . . . .	386
A.9.8 Internal commands . . . . .	386
A.10 Migration to reledmac 2.1.0 . . . . .	386
A.11 Migration to reledmac 2.1.3 . . . . .	387
A.12 Migration to reledmac 2.3.0 . . . . .	387
A.13 Migration to reledmac 2.4.0 . . . . .	387
A.14 Migration to reledmac 2.5.0 . . . . .	387
A.15 Migration to reledmac 2.7.0 . . . . .	387
A.16 Migration to reledmac 2.7.2 . . . . .	388
A.17 Migration to reledmac 2.8.0 . . . . .	388
A.18 Migration to reledmac 2.13.1 . . . . .	388
A.19 Migration to reledmac 2.18.0 . . . . .	388
A.20 Migration to reledmac 2.21.0 . . . . .	388
A.21 Migration to reledmac 2.24.0 . . . . .	388
A.22 Migration to reledmac 2.26.0 . . . . .	389
A.23 Migration to reledmac 2.27.1 . . . . .	389
A.24 Migration to reledmac 2.30.0 . . . . .	389
A.25 Migration to reledmac 2.31.1 . . . . .	389
A.26 Migration to reledmac 2.32.7 . . . . .	389
A.27 Migration to reledmac 2.33.0 . . . . .	389
<b>Appendix B Auxiliary softwares</b> . . . . .	<b>389</b>
B.1 <i>samewords</i> . . . . .	389
B.2 critical-keys for <i>Emacs</i> . . . . .	390
B.3 org-critical-edition for <i>Emacs</i> and <i>Org Mode</i> . . . . .	390
B.4 Import from TEI . . . . .	390
B.5 Import from TEI . . . . .	390
<b>References</b> . . . . .	<b>391</b>
<b>Index</b> . . . . .	<b>391</b>
<b>Change History</b> . . . . .	<b>441</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.

<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).

From 1991 through 1994, the macros continued to evolve, and were tested at a number of sites. We are very grateful to all the members of the (now defunct) `edmac@mailbase.ac.uk` discussion group who helped us with smoothing out the bugs and infelicities in the macros. Ron Whitney and our anonymous reviewer at the TUG were both of great help in ironing out last-minute wrinkles, while Ron made some important suggestions which may help to make future versions of EDMAC even more efficient. Wayne Sullivan, in particular, provided several important fixes and contributions, including adapting the Mittelbach/Schöpf “New Font Selection Scheme” for use with PLAIN  $\TeX$  and EDMAC. Another project Wayne has worked on is a DVI post-processor which works with an EDMAC that has been slightly modified to output `\specials`. This combination enables you to recover to some extent the text of each line as ASCII code, facilitating the creation of concordances, an *index verborum*, etc.

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

### 1.2.2 ledmac

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

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

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

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

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

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

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

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

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

<sup>8</sup>Geert H. M. Claassens, *De Middelnederlandse Kruisvaartromans*, (Amsterdam: 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 <https://www.gwlb.de/leibniz/digitale-ressourcen/repositorium-des-leibniz-archivs>)

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

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

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

### 1.2.3 `eledmac`

Important changes were put in version 1.0, to make `ledmac` more easily extensible (see 7 p. 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. 381.

### 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. 383.

## 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 [7.12.7 p. 53](#).

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

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

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

## 4 Options

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

### 4.1 Specific features

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

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

1.  $\TeX$  is not able to create folder. You should create it yourself.
2. The option does not change the default  $\text{LaTeX}$  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. 385](#));

**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. 73](#));

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

**onlyonerule** by default, each series of footnotes has its own rule; this option makes sure only one footnote rule is printed in each page. See § 7.12.5 (p. 52) for more details.

**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. 72).

**noend** disables tools for endnotes (`\Aendnote`, `\Bendnote` etc.). If you do not need endnotes, this option lets `reledmac` run faster. It will also preserve room for other packages.

**nofamiliar** disables tools for familiar footnotes (`\footnoteA`, `\footnoteB` etc.). If you do not need familiar footnotes, this option lets `reledmac` run faster. It will also preserve room for other packages.

**noledgroup** `reledmac` allows use of a series of critical notes and a new series of normal notes inside `minipage` and `ledgroup` environments (see 10 p. 59). However, such features use up computer memory, at the expense of other processing needs. So if you do not need this feature, use `noledgroup` option. This should make `reledmac` faster.

**series** `reledmac` defines five levels of notes: A, B, C, D, E. Using all these levels consumes memory space and processing speed. This is why, if your work does not require the entire A–E series, you can narrow down the available number of series. For example, if you only need A and B series, call the package with `series={A,B}` option.

# 5 Text lines and paragraphs numbering

## 5.1 Text lines numbering

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



```
\beginnumbering  
Text  
\endnumbering
```

The `\beginnumbering` macro resets the line number to zero, reads an auxiliary file called `\jobname.nn` (where `\jobname` is the name of the main input file for this job, and `nn` is 1 for the first numbered section, 2 for the second section, and so on), and then creates a new version of this auxiliary file to collect information during this run. The first instance of `\beginnumbering` also opens a file called `\jobname.<series>end` to receive the text of the endnotes. `\endnumbering` closes the `\jobname.nn` file.

If the line numbering of a text is to be continuous from start to end, then the whole text will be typed between one pair of `\beginnumbering` and `\endnumbering` commands. But your text will most often contain chapter or other divisions marking sections that should be independently numbered, and these will be appropriate places to begin new numbered sections.

`reledmac` has to read and store in memory a certain amount of information about the entire section when it encounters a `\beginnumbering` command, so it speeds up the processing and reduces memory use when a text is divided into a larger number of sections (at the expense of multiplying the number of external files that are generated).

## 5.2 Paragraphs

### 5.2.1 Basics

`\pstart` Within a numbered section, each paragraph of numbered text must be marked using the `\pend` `\pstart` and `\pend` commands like this:

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

Text that appears within a numbered section but is not marked with `\pstart` and `\pend` will not be numbered.

The following example shows the proper section and paragraph markup and the kind of output that would typically be generated:

```

\beginnumbering
\pstart
This is a sample paragraph, with
lines numbered automatically.
\pend

\pstart
This paragraph too has its
lines automatically numbered.
\pend

The lines of this paragraph are
not numbered.

\pstart
And here the numbering begins
again.
\pend
\endnumbering

```

### 5.2.2 Automatically producing `\pstart ... \pend`

`\autopar` You can use `\autopar` to avoid the nuisance of this paragraph markup and still have every paragraph automatically numbered. The `\autopar` command needs to be called inside a `\beginnumbering... \endnumbering` structure, as follows:

```

\beginnumbering
\autopar

A paragraph of numbered text.

Another paragraph of numbered
text.

\endnumbering

```

`\autopar` fails, however, on paragraphs that start with a `{` or with any other command that starts a new group before it generates any text. Such paragraphs need to be started explicitly, before the new group is opened, using `\indent`, `\noindent`, or `\leavevmode`, or using `\pstart` itself.<sup>13</sup>

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

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

```

---

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

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.

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:

```
\AtStartEveryPstart{*}
```

Instead of manually doing

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

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

It is possible to insert a number at every `\pstart` command; you must use the `\numberpstarttrue` command to have it. You can stop the numbering with `\numberpstartfalse`.

`\numberpstartfalse` You can redefine the command `\thepstart` to change style. You can change the value of the `pstart` number by using *after* `\beginnumbering`:

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

On each `\beginnumbering` the numbering restarts.

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

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

#### 5.2.6 Languages written in Right to Left

If you use languages written right to left with `LuaLaTeX` or `XLaTeX`, 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 L<sup>A</sup>T<sub>E</sub>X.**

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

The first solution is to get a larger L<sup>A</sup>T<sub>E</sub>X with increased memory.

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

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

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

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

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

## 5.3 Lineation commands

### 5.3.1 Disabling lineation

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

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

- It should not be use if you have critical footnotes after `\numberlinefalse`.
- It could be used in the following cases:
  - You want only familiar footnotes, not critical footnotes;
  - You want only parallel typesetting (using `reledpar`) but no critical footnotes.
  - You disable, for a portion of text, line numbering
- It must not be used if:

- You do not want to have the line number in the margins, but do want to use critical footnotes. In this case, set `\firstlinenum` to a big value, such as 100,000 (5.3.2 p. 21).
- 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.

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

```
\linenumberlist
```

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

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

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

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

### 5.3.3 Setting lineation reset

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

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

### 5.3.4 Setting line number margin

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

```
\linenummargin{left}
```

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

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

### 5.3.5 Other settings

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

## 5.4 Changing the line numbers

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

### 5.4.1 Sublineation

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

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

`\Xsublinesep` You can change the separator between the line number and the subline number ei-  
`\Xsublinesepside` ther by using `\Xsublinesep` without any optional argument (7.4.11 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  
`\endlock` value, until you insert `\endlock`. It can tell for itself whether you are in a patch of line or sub-line numbering. One use for line-number locking is in printing poetry: there the line numbers should be those of verse lines rather than of printed lines, even when a verse line requires several printed lines. But in this case you may use the `\stanza` mechanism, see 9 p. 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` `\advanceline` In some cases you may want to modify the line numbers that are automatically calculated: if you are printing only fragments of a work but want to print line numbers appropriate to a complete version, for example. The `\setline{<num>}` and `\advanceline{<num>}` commands may be used to change the current line's number (or the sub-line number, if sub-lineation is currently on). They change both the marginal line numbers and the line numbers passed to the notes. `\setline` takes one argument, the value to which you want the line number set; it must be 0 or greater. `\advanceline` takes one argument, an amount that should be added to the current line number; it may be positive or negative.

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

### 5.4.4 Line number style

`\linenumberstyle` `\sublinenumberstyle` 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` `\hidenumberingonrightpage` `\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.
- 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;

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



- 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. 31) 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,  $\langle v \rangle$  can be after or before to indicate position with regards to the line number.

`\linenumannotationothersidettrue`

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

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

#### 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[ $\langle s \rangle$ ]{ $\langle cmd \rangle$ }` for critical footnotes;  $\langle s \rangle$  indicates the series of notes to which it applies and may be a comma-separated list of series. When  $\langle s \rangle$  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{ $\langle cmd \rangle$ }` for annotations in the margins only. Note that `\Xwraplinenumannotation` without any optional argument will override this setting.
- `\Xwraplinenumannotationref{ $\langle cmd \rangle$ }` 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[ $\langle s \rangle$ ]{ $\langle cmd \rangle$ }` for critical endnotes, where  $\langle s \rangle$  indicates the series of notes to which it applies, which may be a comma-separated list.

$\langle cmd \rangle$  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`

`\Xendnoidenticallinenumannotation`

`\Xnoidenticallinenumannotation[ $\langle s \rangle$ ]`, for critical footnotes  
or `\Xendnoidenticallinenumannotation[ $\langle s \rangle$ ]`

#### 5.5.6 Separator between annotations

`\setlinenumannotationsep`

If there are more than one annotation in the same line, they are separated by a comma.

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

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

If you want to change the separator, you can use `\setlinenumannotationsep{<sep>}`

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

See the file [examples/2-subdivision-number-in-header.tex](#) for an example. Especially, the use of this command with  $\text{\LaTeX}$  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).
- `linrangesep=<c>` to change to `<c>` the separator between start line and end line for this particular note.

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

### 6.2.3 Endnotes

`\Aendnote` **Inserting endnotes** The package also maintains five separate series of endnotes.

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

`\Cendnote` `reledmac`.

`\Dendnote` The mechanism is similar to the one for footnotes: each macro takes one or more

`\Eendnote` 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. 48).
- `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:

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

For example:

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

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

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

However, in some cases, you may want to write a once-expanded<sup>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

<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.

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. 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}}
```

<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`.

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

### 6.3 Disambiguation of identical words in the apparatus

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

#### 6.3.1 Basic use

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

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

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

#### 6.3.2 Case setting

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

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

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

If you use UTF-8 processor, like `XgLaTeX` or `LuaLaTeX`, 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 `XgLaTeX`, add this line in your preamble: `\XeTeXinputnormalization 1`.
- If you use `LuaLaTeX`, use the `uninormalize` package of Michal Hoftich<sup>19</sup> with the `buffer` option set to true.

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



With these tools, Xe<sub>Λ</sub>TeX / LuaTeX 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 Xe<sub>Λ</sub>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 LuaTeX.

#### 6.3.5 Use with `\lemma` command

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

For example, in the following example:

```
some thing
  \edtext{\sameword{sw}
          and other \sameword{sw}
          and again \sameword{sw}
          it is all}%
{\lemma{\sameword{sw} \ldots all}\Afootnote{critical note}}.%
```

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

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

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

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

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

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

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

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

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

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

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

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

Here, the corresponding code:

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

1	A (1) A (2) A (3) A (4) A (5)
1	A <sup>1...A<sup>5</sup></sup> ] level 1
1	A <sup>1...A<sup>3</sup></sup> ] level 2
1	A <sup>2 (2)</sup> ] level 3

### 6.3.6 Sameword for a group of words

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

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

```
\sameword{per} \sameword{causam}
tamen scire
```

```

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

```

1        per causam tamen scire causam est per causam cognoscere causam

---

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

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

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

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

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

```

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

```

1        per causam tamen scire causam est per causam cognoscere causam

---

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

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

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

### 6.3.7 Customizing

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

```

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

```

### 6.3.8 Problems with some macros

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

```

\makeatletter
\appto{\swnoexpands}{%

```

```

\let\somemacro\@gobble%
}
\makeatother

```

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

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

```

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

```

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

```

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

```

### 6.3.9 Automatic sameword annotation

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

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

## 6.4 Apparatus of manuscripts

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

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

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

### 6.4.1 Marking sections of text

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

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

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

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

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

Which means that the part of text containing “Lorem ipsum dolor [...] amet” is witnessed by manuscripts A, B and C, while the part of text starting “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  $\TeX$  footnotes generated via `\footnote`, the package also provides five series of additional footnotes called `\footnoteA` through `\footnoteE`. These  
`\footnoteB` have the familiar marker in the text, and the marked text at the foot of the page can be  
`\footnoteC` formatted using any of the styles described for the critical footnotes. Note that the “regular”  
`\footnoteD` footnotes have the series letter at the end of the macro name whereas the critical  
`\footnoteE` 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  
`\bodyfootmarkA` series A is defined by the `\thefootnoteA` macro; the default is:  
`\footfootmarkA` `\renewcommand*{\thefootnoteA}{\arabic{footnoteA}}`  
 The appearance of the mark in the text is controlled by `\bodyfootmarkA` which is defined as:  
`\newcommand*{\bodyfootmarkA}{%`  
`\hbox{\textsuperscript{\normalfont\@nameuse{@thefnmarkA}}}`  
 The command `\footfootmarkA` controls the appearance of the mark at the start of the footnote text. It is defined as:  
`\newcommand*{\footfootmarkA}{\textsuperscript{\@nameuse{@thefnmarkA}}}`  
 There are similar command triples for the other series.

### 6.5.3 Separator for multiple footnotes

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

`\multfootsep` `\multfootsep` is used as the separator between footnote markers. Its default definition is:  
`\providecommand*{\multfootsep}{\textsuperscript{\normalfont,}}`  
 and can be changed if necessary.

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

`\footnoteXmark` In certain cases, you can't directly use `\footnoteX`; for example, when using `\uline`  
`\footnoteXtext`

command of the `ulem` package. You need to print the footnote mark first, then call the footnote text to be inserted.

For all  $\langle X \rangle$  command, `reledmac` provides a `\footnote\langle X \rangle\text{mark}` command and a `\footnote\langle X \rangle\text{text}` command, equivalent to standard L<sup>A</sup>T<sub>E</sub>X'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\text{nomk}` and `\footnote\langle X \rangle\text{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{\langle s \rangle}` to pull up a given series  $\langle s \rangle$  to the beginning, or `\seriesatend{\langle s \rangle}` to push it down to the end.

## 6.8 Position of critical and familiar footnotes

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

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

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

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

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

`<series><type>`

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

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

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

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

## 7 Apparatus customization

### 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 [*s*], which is the letter of the series – or a list of letters separated by comma – depending on which option is applied. If the optional argument is omitted or empty, the setting will apply to the entire series.

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

<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.



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

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

- `paragraph` formats all of the footnotes of a series as a single paragraph; if you use this arrangement, you are strongly encouraged to read [19.1.6 p. 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  $\langle s \rangle$ .

`\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⟩]`.

`numberonlyfirstintwolines` Suppose you have a lemma on line 2 and a lemma between line 2 and line 3.  
With `\Xnumberonlyfirstinline`, the second lemma is considered to be on the same

line as the first lemma. But if you use both `\Xnumberonlyfirstinline[⟨s⟩]` and `\Xnumberonlyfirstintwoline[⟨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`; `\Xendnumberonlyfirstintwoline` and `\Xendsymlinenum` are the equivalents of `\Xnumberonlyfirstinline`; `\Xnumberonlyfirstintwoline` 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` `\Xendlineprefixsingle[⟨s⟩]{⟨text⟩}` defines the text before the line number in endnotes when there is only one line. Default value is empty.

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

#### 7.4.4 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. 29](#).

`\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 subline number. Note that when you change the lineation system, the option is automatically switched:

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

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

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

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

#### 7.4.9 Printing stanza number

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

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

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

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

#### 7.4.10 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 `⟨symbol⟩` must be robust. Use `\robustify` to robustify a non-robust command.

`\xlinenumannotationonlyfirst` For endnotes, `\xendlinenumannotationonlyfirst`; `\xendlinenumannotationonlyfirstintwo` and `\xendsymlinenumannotation` are the equivalents of `\xlinenumannotationonlyfirst`; `\xlinenumannotationonlyfirstintwo` and `\xsymlinenumannotation`.

`\xnoLineNumberifAnnotation` The `\xnoLineNumberifAnnotation` disable line number printing only for critical notes when there are some annotations to the line number.

`\xnoLineNumberifAnnotation` The `\xnoLineNumberifAnnotation` is the same for endnotes.

#### 7.4.11 Separator between line and subline numbers

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

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

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

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

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

#### 7.4.12 Separator between page and line numbers

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

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

#### 7.4.13 Space around number

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

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

`\xendbeforenumber` `\xendbeforenumber` and `\xendaftersnumber` are the equivalents of `\xbeforenumber` and `\xaftersnumber` for endnotes.

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

#### 7.4.14 Space around line symbol

- `\Xbeforemyslinenum` With `\Xbeforemyslinenum[⟨s⟩]{⟨l⟩}` you can add some space before the line symbol in a footnote. The default value is value set by `\Xbeforenumber`.
- `\Xaftersmyslinenum` With `\Xaftersmyslinenum[⟨s⟩]{⟨l⟩}` you can add some space after the line symbol in a footnote. The default value is value set by `\Xafternumber`.
- `\Xendbeforemyslinenum` `\Xendbeforemyslinenum` and `\Xendaftersmyslinenum` are the equivalents of `\Xbeforemyslinenum` and `\Xaftersmyslinenum` 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}
```

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

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

`\Xendboxlinenum` `\Xnotboxingsubline[⟨s⟩`. `\Xendboxlinenum[⟨s⟩]{⟨l⟩}`, `\Xendboxlinenumalign[⟨s⟩]{⟨text⟩}`,  
`\Xendboxlinenumalign` `\Xendboxstartlinenum[⟨s⟩]{⟨l⟩}`, `\Xendnotboxingsubline[⟨s⟩]`, `\Xendboxendlinenum[⟨s⟩]{⟨l⟩}`  
`\Xendboxstartline` are the same as, respectively, `\Xboxlinenum`, `\Xboxlinenumalign`, `\Xboxstartlinenum`,  
`\Xendboxendlinenum` `\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

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



the lemma and the separator. If your lemma separator is empty, this space won't be printed. The default value is 0 em.

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

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

## 7.6 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  $\TeX$  font command to the lemma. For example, to have boldface lemma:

```
\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  $\TeX$  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  $\TeX$  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  $\TeX$  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  $\LaTeX$  command. For example, with the `bid` package, to ensure having a lemma written right to left, use `\Xwraplemma{\RL}`.

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

### 7.7.2 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 `\Xendbeforemsynlinenumber` 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 `\Xendaftersynlinenumber` 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}.}}
```

<code>\Xbhooknote</code>	<code>\Xbhooknote[⟨s⟩]{⟨code⟩}</code> is to be used at the beginning of each critical footnote.
<code>\bhooknoteX</code>	<code>\bhooknoteX[⟨s⟩]{⟨code⟩}</code> is to be used at the beginning of each familiar footnote.
<code>\Xendbhooknote</code>	<code>\Xendbhooknote[⟨s⟩]{⟨code⟩}</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. 49).

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

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

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

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

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

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

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

### 7.10.2 Size of the columns

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

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

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

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

## 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).

`\Xgroupbylineseparetwolines` 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 `\Xgroupbylineseparetwolines[⟨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

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

### 7.12.4 Spacing

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

**Be careful, the standard  $\LaTeX$  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  $\LaTeX$  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 Printing only one footnote rule per page

By default, each series of notes has its own footnote rule. Using `onlyonerule` when loading the `reledmac` packages ensures that only one rule will be printed in each page.

However, you may want to have only one footnote rule for all series of *critical* footnotes, but also one (and only one) footnote rule for all series of *familiar* footnotes.

For example, if in one page you have three series of critical footnotes and four series of familiar footnotes, you may want to have, in all, two rules.

To do that use

`\onlyonerule` • to produce only a single footnote rule for all critical footnotes;

`onlyoneruleX` • to produce only a single footnote rule for all familiar footnotes.

### 7.12.6 Space after footnotes rules

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

**Be careful, the standard  $\LaTeX$  footnote rule, which is used by `reledmac`, adds 2.6pt. These 2.6pt are not changed by this command.**

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

**Be careful, the standard 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.**

### 7.12.7 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`.

### 7.12.8 Width

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

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

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

Note that changes the with of the block of notes. If you want to change the width of each column when typesetting notes in columns, use `\Xhsizetwocol`, `\Xhsizethreecol`, `\hsizetwocolX`, `\hsizethreecolX`, see 7.10.2 p. 50.

## 7.13 Footnotes and the `reledpar` columns

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

`\noteswidthliketwocolumnsX`

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

## 7.14 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. 57)

## 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.

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

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

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

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

If you want the hanging verse to be flush right, you can use `\sethanginsymbol:` see p. 9.7 p. 57.

Enumeration is by stanza lines, not by print lines. In the above example the lines are indented one unit, two units, one unit, two units, with 3 units of hanging indentation in case a stanza line is too long to fit on one print line.

## 9.3 Repeating stanza indents

Since version 0.13, if the indentation is repeated every  $n$  verses of the stanza, you can define only the  $n$  first indentations, and indicate that they are repeated, defining the value of the `stanzaindentrepetition` counter at  $n$ . For example:

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

is like

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

**Be careful: the feature is changed in `eledmac` 1.5.1. See [A.3 p. 382](#).**

If you don't use the `stanzaindentrepetition` counter, make sure you have at least one more numerical entry in `\setstanzavalues` than the number of lines in the stanza.

If you want to disable this feature again, just put the counter to 0:

```
\setcounter{stanzaindentrepetition}{0}
```

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

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

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

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



```
\setcounter{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.

`\stanzaindent*`

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`.

`\numberstanzafalse`

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`

`\afterantilabe`

You can execute any arbitrary code before or after `\antilabe` macro just with a redefinition 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. 57).

## 10 Grouping

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

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

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

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

`ledgroupsize` The `ledgroupsize` environment is similar to `ledgroup` except that you must specify a width for the environment, as with a `minipage`.  
`\begin{ledgroupsize}[⟨pos⟩]{⟨width⟩}`.

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

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

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

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

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

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

## 11 Cross referencing

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

### 11.1 Basic use

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

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

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

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

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

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

### 11.2 Cross-referencing to a critical note

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

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

If you want to refer to starting and ending lines, you should use `\appref` and related tools (11.6.2 p. 62).

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

`\xpageref`  
`\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 `ℒTeX` 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. 31).

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.

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

- 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. 31) and `\lineannot` (5.5.2 p. 25) and sets the beginning page, line, subline numbers and line annotations to those of the place where `\edlabel{mouse}` was placed, and the ending ones to those where `\edlabel{elephant}` occurs.

For example, one might use the following:

```
\beginnumbering

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

\pstart
\edtext{Et videtur quod non\edlabel{non}.}{\xxref{Queritur}{non}\lemma{queritur \dots{} non}\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,  
`\ref` and operate in the familiar fashion.  
`\pageref`

## 11.6 References to start and end lines

### 11.6.1 Reference to main text lines

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

<code>\edlabelS</code>	Use <code>\edlabelS{&lt;label&gt;}</code> to mark the start line of the passage.
<code>\edlabelE</code>	Use <code>\edlabelE{&lt;label&gt;}</code> to mark the end the end line of the passage. These two commands just create two labels which are named <code>&lt;label&gt;:start</code> and <code>&lt;label&gt;:end</code> .
<code>\edlabelSE</code>	Use <code>\edlabelSE{&lt;label&gt;}</code> to mark just one location in the text. Contrary to a classical <code>\edlabel</code> , the <code>&lt;label&gt;</code> could be use with <code>\Seref</code> and <code>\Serefwithpage</code> .
<code>\Seref</code>	The main utility is to use them with three other commands. <code>\Seref{&lt;label&gt;}</code> will make a cross-reference printed as a reference in critical footnotes.
<code>\Serefwithpage</code>	<code>\Serefwithpage</code> will make a cross-reference printed as a reference in critical endnotes.
<code>\Serefonlypage</code>	<code>\Serefonlypage</code> 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.

<code>\applabel</code>	If you use <code>\applabel{&lt;label&gt;}</code> inside the second argument of a <code>\edtext</code> , <code>reledmac</code> will add a <code>\edlabel</code> at the beginning and end of the marked passage. The label at the beginning of the passage will have the title <code>&lt;label&gt;:start</code> , while the label at the end will have the title <code>&lt;label&gt;:end</code> . If you use <code>\linenum</code> (6.2.5 p. 31) to refer to these labels, <code>reledmac</code> will use your line settings to refer to the passage.
<code>\appref</code>	You can also use <code>\appref{&lt;label&gt;}</code> and <code>\apprefwithpage{&lt;label&gt;}</code> 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

<code>\setapprefprefixsingle</code>	<b>Specific to these tools</b> If you use <code>\apprefprefixsingle{&lt;prefix&gt;}</code> , <code>&lt;prefix&gt;</code> will be printed before the line numbers of a <code>\appref</code> -reference. If you use <code>\apprefprefixmore{&lt;prefix&gt;}</code> , <code>&lt;prefix&gt;</code> will be printed before the line numbers, if you refer to more than one line.
<code>\setapprefprefixmore</code>	

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.

<code>\setSerefprefixsingle</code>	<code>\setSerefprefixsingle</code> and <code>\setSerefprefixmore</code> are similar for <code>\Seref</code>
<code>\setSerefprefixmore</code>	command.
<code>\setSerefonlypageprefixsingle</code>	Use <code>\setSerefonlypageprefixsingle{&lt;prefix&gt;}</code> to set the page prefix for <code>\Serefonlypage</code>
<code>\setSerefonlypageprefixmore</code>	when there is only one page. Use <code>\setSerefonlypageprefixmore{&lt;prefix&gt;}</code> to set it

when there is more than one page. For example:

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

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

**Also note that `\setSErefonlypageprefixsingle` is only a shortcut for `\XendbeforepagenumberSErefonly` (see 11.6.3 p. 63). So if you use `\Xendbeforepagenumber` without any optional argument, it will override this setting.**

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

These commands are the following:

- `\Xlineflag` (for `reledpar`), enabled by default.
- `\Xlinerangeseparator`
- `\Xmorethantwolines`
- `\Xsublinesep`
- `\Xtwolines`
- `\Xtwolinesbutnotmore`
- `\Xtwolinesonlyinsamepage`
- `\Xlinenumannotationposition`
- `\Xwraplinenumannotation`
- `\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. 29).

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}`, `\SEref` to achieve this result.

### 11.7 Compatibility with `xr` package

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

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

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

## 12 Sidenotes

### 12.1 Basics

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

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

`\ledleftnote` `\ledsidenote{<text>}` will put `<text>` into the margin specified by the current

`\ledrightnote`

`\ledsidenote`

`\sidenotemargin`



setting of `\sidenotemargin{<location>}`. The permissible value for `<location>` is one out of the list `left`, `right`, `inner`, or `outer`, for example `\sidenotemargin{outer}`. 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  $\langle item \rangle$  and the current page number should be added to the raw index (`idx`) file. The `\edindex{<item>}` macro can be used in numbered text to specify that  $\langle item \rangle$  and the current page & linenummer should be added to the raw index file.

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

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

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

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

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

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

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

### 13.3 Referring to critical notes

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

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

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

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

`\innotemarkX`

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

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

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

For example:

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

### 13.4 Separator between page and line numbers

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

- is the default separator used by the MAKEINDEX program.

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

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

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

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>

<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`.

## 13.6 Advanced setting

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

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

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

## 14 Glossary

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

### 14.1 Preamble setting

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. 67).

### 14.2 Commands

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

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

## 15 Tabular material

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

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

produces the following parallel pair of verses.

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

`\edtabcolsep` The distance between the columns is controlled by the length `\edtabcolsep`.  
`\spreadmath` `\spreadmath{<math>}` typesets `{<math>}` but the `{<math>}` has no effect on the  
`\spreadtext` calculation of column widths. `\spreadtext{<text>}` is the analogous command for use  
in `edtabular` environments.

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

`\edrowfill` The macro `\edrowfill{<start>}{<end>}{<fill>}` fills columns number `<start>` to `<end>` inclusive with `<fill>`. The `<fill>` argument can be any horizontal “fill”. For example, `\hrulefill` or `\upbracefill`.

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

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

```
\begin{edtabularr}
1 & & & & & \\
Q & & & & & \\
v & & & & & \\
g & & & & & \\
& 2 & & 3 & & 4 & & 5 \\
& & & fd & & h & & qwertziohg \\
& wptz & & x & & y & & vb \\
& nnn & & \edrowfill{3}{5}{\upbracefill} & & & & \end{edtabularr}
```



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

For example:

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

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

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

```
\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 `\eledchapter` 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.  $n + 1$ .

`\ledpbsetting` However, you can change the behavior and decide they will have effect after the end of the line, adding `\ledpbsetting{after}` at the beginning of your file (better: in your preamble). With the previous example, l. 444 will be on p.  $n$  and l. 445 will be on p.  $n + 1$ .

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

### 18.2 Prevent page break in a long verses

`\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 `maxnotesX` and/or `Xmaxnotes` as explained in 7.12.7 p. 53 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` (7.12.7 p. 53), 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 targetted for `LaTeX2e`.

```

1 %<*code>
2 \NeedsTeXFormat{LaTeX2e}
3 \ProvidesPackage{reledmac}[2025/01/07 v2.40.2 typesetting critical editions
4 ]%
5 %
```

### II.3 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.

```

5 \RequirePackage{xargs}%
6 \RequirePackage{xparse}[2017/03/07]%
7 \RequirePackage{etoolbox}%
8 \@ifl@t@r\fmtversion{2015/10/01}%
9 {\ifboolexpr{not test{\@ifl@t@r\fmtversion{2016/03/31}} or (test{\
ifdefstring{\fmtversion}{2016/03/31}} and test {\ifnumless{\patch@level
}{3}})}%
10   {\PackageWarning{reledmac}{You are using a LaTeX version older than
2016/03/31 patch 3.%
11   \MessageBreak You are strongly encouraged to use a newer version.}}%
12   }%
13 }%
14 {\RequirePackage{etex}%
15 \csname reserveinserts\endcsname{32}%
16 }%
17 \RequirePackage{suffix}%
18 \RequirePackage{xstring}%
19 \RequirePackage{ifluatex}%
20 \RequirePackage{ragged2e}%
21 \RequirePackage{ifxetex}%
22 %

```

### II.4 Package options

<pre> \ifparledgroup \ifsameparallelpagenumber \ifprevpgnotnumbered \ifledfinal \ifnocritical@ \if@noeled@sec \ifnoend@ \ifnofamiliar@ \ifnoledgroup@ \ifparapparatus@ \ifnoquotation@ \iflednopbinverse \ifwidthliketwocolumns \ifxindy@ \ifxindyhyperref@ \ifeledmaccompat@ \ifnoresetlinenumnotation@ \ifantilabe@ </pre>	<p>Use this to remember which option is used, set and execute the options with <code>final</code> as the default. We use <code>xkeyval</code> in order to manage options with argument.</p> <pre> 23 \RequirePackage{xkeyval} 24 % </pre> <h4>II.4.1 Options of <code>reledpar</code></h4> <p>Here, we define some booleans specific to <code>reledpar</code> options, but which have consequences on <code>reledmac</code> internal commands.</p> <pre> 25 \newif\ifparledgroup 26 \newif\ifsameparallelpagenumber% 27 \newif\ifprevpgnotnumbered%% 28 % </pre> <h4>II.4.2 Options of <code>reledmac</code></h4>
--	---

```

29 \DeclareOptionX{series}[A,B,C,D,E]{\xdef\default@series{#1}}
30 \ExecuteOptionsX{series}%
31
32 \newif\if@noeled@sec%
33 \DeclareOptionX{noeledsec}{\@noeled@sectrue}
34
35 \newif\ifnocritical@%
36 \DeclareOptionX{nocritical}{\nocritical@true}%
37
38 \newif\ifnofamiliar@%
39 \DeclareOptionX{nofamiliar}{\nofamiliar@true}%
40
41 \newif\ifnoledgroup@%
42 \DeclareOptionX{noledgroup}{\noledgroup@true}%
43
44 \newif\ifnoend@%
45 \DeclareOptionX{noend}{%
46   \let\l@dend@open\@gobble%
47   \let\l@dend@close\relax%
48   \global\let\l@dend@stuff=\relax%
49   \noend@true%
50 }%
51
52 \newif\ifnoquotation@
53 \DeclareOptionX{noquotation}{\noquotation@true}
54
55 \newif\ifledfinal
56 \DeclareOptionX{final}{\ledfinaltrue}
57 \DeclareOptionX{draft}{\ledfinalfalse}
58 \ExecuteOptionsX{final}
59
60 \newif\ifparapparatus@
61 \DeclareOptionX{parapparatus}{\parapparatus@true}
62
63 \newif\iflednopbinverse
64 \DeclareOptionX{nopbinverse}{\lednopbinversetrue}
65
66 \newif\ifwidthliketwocolumns%
67 \DeclareOptionX{widthliketwocolumns}{\widthliketwocolumnstrue}%
68
69 \newif\ifcontinuousnumberingwithcolumns
70 \DeclareOptionX{continuousnumberingwithcolumns}{\
  continuousnumberingwithcolumnstrue}%
71
72 \newif\ifxindy@
73 \DeclareOptionX{xindy}[eledmac-markup-attr.xdy]{%
74   \AtBeginDocument{\immediate\openout\eledmac@xindy@out=#1}%
75   \newwrite\eledmac@xindy@out%
76   \xindy@true%
77   \gdef\eledmacmarkuplocdepth{:depth 1}%
78   \AtEndDocument{\immediate\closeout\eledmac@xindy@out}%
79 }%
80
81 \newif\ifxindyhyperref@

```

```

82 \DeclareOptionX{xindy+hyperref}{%
83   \xindyhyperref@true%
84 }%
85
86 \newif\ifeledmaccompat@%
87 \DeclareOptionX{eledmac-compat}{%
88   \eledmaccompat@true%
89 }%
90 \DeclareOptionX{nopenalties}{%
91   \AtBeginDocument{\let\add@penalties\relax}%
92 }
93 \def\l@auxdir{}%
94 \DeclareOptionX{auxdir}{%
95   \xdef\l@auxdir{#1}/}%
96 }%
97
98 \newif\ifsw@caseinsensitive%
99 \DeclareOptionX{swcaseinsensitive}{%
100   \sw@caseinsensitivetrue%
101 }%
102 \newif\ifnoresetlinenumannotation@
103 \DeclareOptionX{noresetlinenumannotation}{%
104   \noresetlinenumannotation@true%
105 }%
106 \newif\ifantilabe@%
107 \DeclareOptionX{antilabe}{%
108   \antilabe@true%
109 }%
110
111 \newtoggle{onlyonerule@}%
112 \DeclareOptionX{onlyonerule}{%
113   \toggletrue{onlyonerule@}%
114 }%
115 %

```

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.

```

116 \ProcessOptionsX*\relax
117
118 %

```

## II.5 Compatibility with LuaTeX

Here, we enable some primitives for LuaTeX.

```

119 \ifx\directlua\undefined\else%
120   \directlua{tex.enableprimitives("",{"textdir","pardir","bodydir"})}
121 \fi
122 \ifluatex%
123   \ifnum\luatexversion<110%
124     \PackageWarning{reledmac}{You need to use LuaTeX 1.1.0 or higher}{\@ehc}
125   }%

```



```

125 \fi%
126 \fi%
127 %

```

## II.6 Boolean flags

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

```

128 \newif\ifl@dmemoir
129 \@ifclassloaded{memoir}{\l@dmemoirtrue}{\l@dmemoirfalse}
130
131 %

```

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

```

132 \newif\if@ledgroup%
133 %

```

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

```

134 \newif\ifl@imakeidx
135 \@ifpackageloaded{imakeidx}{\l@imakeidxtrue}{}%False is the default value
136 %

```

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

```

137 \newif\ifl@indextools%
138 \@ifpackageloaded{indextools}{%
139   \l@indextoolstrue%
140   \l@imakeidxtrue%
141   \let\imki@wrindexentry\indtl@wrindexentry%
142 }{}%
143 %

```

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.

```

144 \newif\ifl@footmisc
145 \@ifpackageloaded{footmisc}{\l@footmisctrue}{}%False is the default value
146 %

```

`\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.

```

147 \ifdef{\if@RTL}{\newif\if@RTL}
148 %

```

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

```

149 \newif\if@firstlineofpage%
150 \newif\if@firstlineofpageR%
151 %

```

## II.7 Messages

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

`\reledmac@warning` Write a warning message.

```
152 \newcommand{\reledmac@warning}[1]{\PackageWarning{reledmac}{#1}}
153 %
```

`\reledmac@error` Write an error message.

```
154 \newcommand{\reledmac@error}[2]{\PackageError{reledmac}{#1}{#2}}
155 %
```

`\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}{%`

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

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

```

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

```
162 %
```

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

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

```
165 }%
```

```
166 %
```

`\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}`

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

```

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

```
172 }%
```

```
173 %
```

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

```

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

```
176 }%
```

```
177 %
```

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

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

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

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

```

183     before the footnote placement}%
184     \typeout{ and line numbering in this section are
185     correct.}}
186 %

```

```

\led@mess@SectionContinued87 \newcommand*\led@mess@SectionContinued}[1]{%
188     \message{Section #1 (continuing the previous section)}}
189 %

```

```

\led@err@LineationInNumbered90 \newcommand*\led@err@LineationInNumbered}{%
191     \reledmac@error{You can't use \string\lineation\space within
192     a numbered section}{\@ehc}}
193 %

```

```

\led@warn@BadLineation94 \newcommand*\led@warn@BadLineation}{%
\led@warn@BadLinenummargin95 \reledmac@warning{Bad \string\lineation\space argument}}
\led@warn@BadLockdisp96 \newcommand*\led@warn@BadLinenummargin}{%
\led@warn@BadSublockdisp97 \reledmac@warning{Bad \string\linenummargin\space argument}}
198 \newcommand*\led@warn@BadLockdisp}{%
199 \reledmac@warning{Bad \string\lockdisp\space argument}}
200 \newcommand*\led@warn@BadSublockdisp}{%
201 \reledmac@warning{Bad \string\sublockdisp\space argument}}
202 %

```

```

\led@warn@NoFile03 \newcommand*\led@warn@NoFile}[1]{%
204     \reledmac@warning{File `#1' not found}}
205 %

```

```

\led@warn@LineFileObsolete06 \newcommand*\led@warn@Obsolete}[1]{%
207     \reledmac@warning{Line-list file #1 was obsolete. We have not read it.
Please run LaTeX again.}}
208 %

```

```

\led@warn@BadAdvancelineSubline09 \newcommand*\led@warn@BadAdvancelineSubline}{%
\led@warn@BadAdvancelineLine10 \reledmac@warning{\string\advanceline\space produced a sub-line
211     number less than zero.}}
212 \newcommand*\led@warn@BadAdvancelineLine}{%
213 \reledmac@warning{\string\advanceline\space produced a line
214     number less than zero.}}
215 %

```

```

\led@warn@BadSetline16 \newcommand*\led@warn@BadSetline}{%
\led@warn@BadSetlinenum17 \reledmac@warning{Bad \string\setline\space argument}}
218 \newcommand*\led@warn@BadSetlinenum}{%
219 \reledmac@warning{Bad \string\setlinenum\space argument}}
220 %

```

```

\led@err@PstartNotNumbered221 \newcommand*{\led@err@PstartNotNumbered}{%
\led@err@PstartInPstart222 \reledmac@error{\string\pstart\space must be used within a
\led@err@PendNotNumbered223 numbered section %
\led@err@PendNoPstart224 (\string\beginnumbering\space ... \space\string\
\led@err@AutoparNotNumbered225 endnumbering)}{\@ehc}}%
\led@err@NumberingWithoutPstart226 \newcommand*{\led@err@PstartInPstart}{%
227 \reledmac@error{\string\pstart\space encountered while another
228 \string\pstart\space was in effect}{\@ehc}}
229 \newcommand*{\led@err@PendNotNumbered}{%
230 \reledmac@error{\string\pend\space must be used within a
231 numbered section}{\@ehc}}
232 \newcommand*{\led@err@PendNoPstart}{%
233 \reledmac@error{\string\pend\space must follow a \string\pstart}{\@ehc}}
234 \newcommand*{\led@err@AutoparNotNumbered}{%
235 \reledmac@error{\string\autopar\space must be used within a
236 numbered section}{\@ehc}}
237 \newcommand*{\led@err@NumberingWithoutPstart}{%
238 \reledmac@error{\string\beginnumbering... \string\endnumbering\space
without \string\pstart}{\@ehc}}%
%

\led@warn@BadAction339 \newcommand*{\led@warn@BadAction}{%
240 \reledmac@warning{Bad action code, value \next@action.}}
241 %

\led@warn@DuplicateLabel342 \newcommand*{\led@warn@DuplicateLabel}[1]{%
ed@warn@AppLabelOutSecondArgEdtext343 \reledmac@warning{Duplicate definition of label `#1'\@gobble}%
\led@warn@RefUndefined344 \@latex@warning@no@line{Label `#1' multiply defined}%
\led@warn@RefUndefined345 }%
246 \newcommand*{\led@warn@AppLabelOutSecondArgEdtext}[1]{%
247 \reledmac@warning{\string\applabel\space outside of the second argument
of an \string\edtext\space `#1' on page \thepage.}}%
248 \newcommand*{\led@warn@RefUndefined}[1]{%
249 \G@refundefinedtrue%
250 \reledmac@warning{Reference `#1' on page \thepage\space undefined.%
251 Using `000'.}%
252 \@latex@warning{Reference `#1' undefined\on@line}%
253 }%
254 \newcommand*{\led@warn@pairRefUndefined}[1]{%
255 \G@refundefinedtrue%
256 \reledmac@warning{Reference `#1:start' and/or `#1:end' on page \thepage\
space undefined.
257 Using `??'.}%
258 \@latex@warning{Reference `#1:start' and/or `#1:end' undefined\on@line}%
259 }
260 %

\led@warn@NoMarginpars61 \newcommand*{\led@warn@NoMarginpars}{%
262 \reledmac@warning{You can't use \string\marginpar\space in numbered text
}}
263 %

```

```

\led@warn@BadHookSeries264 \newcommand{\led@warn@BadHookSeries}[2]{%
265   \reledmac@warning{Bad series #1 for #2 hook. Skipped.}%
266 }%
267 %

\led@warn@BadSidenotemargin268 \newcommand*\led@warn@BadSidenotemargin{%
269   \reledmac@warning{Bad \string\sidenotemmargin\space argument}}
270 %

\led@warn@NoIndexFile271 \newcommand*\led@warn@NoIndexFile}[1]{%
272   \reledmac@warning{Undefined index file #1}}
273 %

\led@warn@SeriesStillExist274 \newcommand{\led@warn@SeriesStillExist}[1]{%
275   \reledmac@warning{Series #1 is still existing !}%
276 }%
277 %

\led@err@BadAction278 \newcommand*\led@err@StanzaIndentNotDefined{%
279   \reledmac@error{You have not defined the indentation for the line \number
\stanza@count}{\@ehc}}%
280 %

\led@err@ManySidenotes281 \newcommand{\led@err@ManySidenotes}{%
\led@err@ManyLeftnotes282   \ifledRcol%
\led@err@ManyRightnotes283   \reledmac@warning{\itemcount@\space sidenotes on line \the\line@numR\
space p. \the\page@numR}%
284   \else%
285     \reledmac@warning{\itemcount@\space sidenotes on line \the\line@num\
space p. \the\page@num}%
286   \fi%
287 }%
288 \newcommand{\led@err@ManyLeftnotes}{%
289   \ifledRcol%
290     \reledmac@warning{\itemcount@\space leftnotes on line \the\line@numR\
space p. \the\page@numR}%
291   \else%
292     \reledmac@warning{\itemcount@\space leftnotes on line \the\line@num\
space p. \the\page@num}%
293   \fi%
294 }%
295 \newcommand{\led@err@ManyRightnotes}{%
296   \ifledRcol%
297     \reledmac@warning{\itemcount@\space rightnotes on line \the\line@numR\
space p. \the\page@numR}%
298   \else%
299     \reledmac@warning{\itemcount@\space rightnotes on line \the\line@num\
space p. \the\page@num}%

```

```

300 \fi%
301 }%
302 %

\led@err@TooManyColumns303 \newcommand*\led@err@TooManyColumns}{%
\led@err@UnequalColumns304 \reledmac@error{Too many columns}{\@ehc}}
\led@err@LowStartColumn305 \newcommand*\led@err@UnequalColumns}{%
\led@err@HighEndColumn306 \reledmac@error{Number of columns is not equal to the number
\led@err@ReverseColumns307 in the previous row (or \protect\\ \space forgotten?)}{\@ehc}}
308 \newcommand*\led@err@LowStartColumn}{%
309 \reledmac@error{Start column is too low}{\@ehc}}
310 \newcommand*\led@err@HighEndColumn}{%
311 \reledmac@error{End column is too high}{\@ehc}}
312 \newcommand*\led@err@ReverseColumns}{%
313 \reledmac@error{Start column is greater than end column}{\@ehc}}
314 %

\led@err@toendnotes@outsidenumbering315 \newcommand\led@err@toendnotes@outsidenumbering}{%
316 \reledmac@error{\string\toendnotes\space and related commands must be
called inside a numbered texte (\string\beginnumbering\space ... \space\string\endnumbering)}{\@ehc}%
317 }%
318 %

\led@err@EdtextWithoutNote319 \newcommand\led@err@EdtextWithoutNote}[1]{%
320 \reledmac@error{edtext ``\unexpanded{#1}` without Xfootnote or Xendnote.
Check syntaxis.}{\@ehc}%
321 }%
322 %

\led@err@FootnoteNotInSecondArgEdtext323 \newcommand\led@err@FootnoteNotInSecondArgEdtext}[1]{%
324 \reledmac@error{#1footnote outside of the second argument of an edtext.
Check syntaxis.}{\@ehc}%
325 }%
326 %

\led@error@PackageAfterEledmac327 \newcommand\led@error@PackageAfterEledmac}[1]{%
328 \reledmac@error{#1 must be loaded before reledmac}{\@ehc}%
329 }%
330 %

\led@error@require@glossariesextra331 \newcommand\led@error@require@glossariesextra}{%
332 \reledmac@error{When using both reledmac and glossaries, you must also
use glossaries-extra}{\@ehc}%
333 }%
334 %

```

```

error@fail@patch@@makecol335 \newcommand{\led@error@fail@patch@@makecol}{%
336   \reledmac@error{Fail to patch \string\@makecol\space command}{\@ehc}%
337 }%
338 %

```

```

r@fail@patch@@reinserts339 \newcommand{\led@error@fail@patch@@reinserts}{%
340   \reledmac@error{Fail to patch \string\@reinserts\space command}{\@ehc}%
341 }%
342 %

```

```

ofail@patch@@docclearpage43 \newcommand{\led@error@fail@patch@@docclearpage}{%
344   \reledmac@error{Fail to patch \string\@docclearpage\space command}{\@ehc}%
345 }%
346 %

```

```

ofail@patch@@iiiminipage47 \newcommand{\led@error@fail@patch@@iiiminipage}{%
348   \reledmac@error{Fail to patch \string\@iiiminipage\space command}{\@ehc}%
349 }%
350 %

```

```

ofail@patch@endminipage51 \newcommand{\led@error@fail@patch@endminipage}{%
352   \reledmac@error{Failed to patch the \string\endminipage\space command}{\
@ehc}%
353 }%
354 %

```

```

ofail@patch@endminipage55 \newcommand{\led@error@fail@patch@makeindex}{%
356   \reledmac@error{Failed to patch the \string\makeindex\space command}{\
@ehc}%
357 }%
358 %

```

```

r@fail@patch@setcounter59 \newcommand{\led@error@fail@patch@setcounter}{%
360   \reledmac@error{Failed to patch the \string\setcounter\space command}{\
@ehc}%
361 }%
362 %

```

```

error@fail@patch@stpelt63 \newcommand{\led@error@fail@patch@stpelt}{%
364   \reledmac@error{Failed to patch the \string\@stpelt\space command}{\@ehc}
%
365 }%
366 %

```

```

\led@error@fail@patch@addtocounter 367 \newcommand{\led@error@fail@patch@addtocounter}{%
368   \reledmac@error{Failed to patch the \string\addtocounter\space command}{\
@ehc}%
369 }%
370 %

371 \newcommand{\led@error@fail@patch@stepcounter}{%
372   \reledmac@error{Failed to patch the \string\stepcounter\space command}{\
@ehc}%
373 }%
374 %

\led@error@fail@patch@mempnum 375 \newcommand{\led@error@fail@patch@mempnum}{%
376   \reledmac@error{Fail to patch \string@mempnum\space command}{\@ehc}%
377 }%
378 %

\led@warn@edinde@outsidenumbering 379 \newcommand{\led@warn@edinde@outsidenumbering}{%
380   \reledmac@warning{\string\edindex\space called outside of \string\
beginnumbering\space ..\space \string\endnumbering. \MessageBreak
Automatically switched to \string\index.}%
381 }%
382 %

\led@warning@hsizeX@deprecated 383 \newcommand{\led@warning@hsizeX@deprecated}{%
384   \reledmac@warning{\string\hsizeX\space command deprecated, use \string\
widthX\space instead.}%
385 }%
386 %

\led@warning@Xhsize@deprecated 387 \newcommand{\led@warning@Xhsize@deprecated}{%
388   \reledmac@warning{\string\Xhsize\space command deprecated, use \string\
Xwidth\space instead.}%
389 }%
390 %

\led@warning@msdatawithoutstop 391 \newcommand{\led@warning@msdatawithoutstop}{%
392   \reledmac@warning{\string\msdata\space without corresponding \string\
stopmsdata}%
393 }%
394 %

\led@warning@preXnotes@deprecated 395 \newcommand{\led@warning@preXnotes@deprecated}{%
396   \reledmac@warning@preXnotes@deprecated%
397 }%
398 %

```



## II.8 Gobbling

Here, we define some commands which gobble their arguments.

```
\@gobblethree99 \providecommand*\@gobblethree}[3]{}
\@gobblefour100 \providecommand*\@gobblefour}[4]{}
\@gobbleseven91 \providecommand*\@gobbleseven}[7]{}
402 %
```

## 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.

```
403 \newcommand{\l@wrap@ifnotemptybox}[2]{%
404   \setbox0=\hbox{#2}%
405   \ifdim\wd0=\z@ \else%
406     #1{#2}%
407   \fi%
408 }%
409 \newcommand{\l@wrapcs@ifnotemptybox}[2]{%
410   \l@wrap@ifnotemptybox{\csname #1\endcsname}{#2}%
411 }%
412 %
```

## II.10 Miscellaneous commands

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

```
413 \ifldefinal
414   \global\notbool{parapparatus@}\newcommand*{\newcommand}\showlemma
415   }[1]{#1}%
416 \else
417   \global\notbool{parapparatus@}\newcommand*{\newcommand}\showlemma
418   }[1]{\textbf{#1}}%
419 \fi
%
```

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

Initialize it as `\empty`.

```
420 \let\linenumberlist=\empty
421
422 %
```

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

`\@l@tempcnta` In imitation of  $\TeX$ , we create a couple of scratch counters.  
`\@l@tempcntb`  $\TeX$  already defines `\@tempcnta` and `\@tempcntb` but Peter Wilson found in the past that it can be dangerous to use these (for example one of the AMS packages did something nasty to the `ccaption` package's use of one of these).

```
423 \newcount\@l@tempcnta \newcount\@l@tempcntb
424 %
```

## 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 425 \newif\ifl@dpairing
\ifpst@rtedL 426 \newif\ifl@dpaging%
\l@dnumpststartsL 427 \newif\ifl@dprintingpages%
428 \newif\ifl@dprintingcolumns%
429 \newif\ifpst@rtedL
430 \newcount\l@dnumpststartsL
431 %
```

`\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`.

```
432 \newif\ifledRcol
433 \newif\ifledRcol@
434 %
```

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

```
435 \newif\ifnumberingR
436 %
```

`\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`.

```
437 \newif\ifafterendnumberingR%
438 %
```

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

```
439 \newif\ifXnote@%
440 %
```

## 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.

```
441 \providebool{indtl@innote}%
442 \providebool{indtl@notenumber}%
443 %
```

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

```
444 \newif\ifinstanzaL%
445 \newif\ifinstanzaR%
446 %
```

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

```
447 \newif\ifinastanzaL%
448 \newif\ifinastanzaR%
449 %
```

## 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.

```
450 \newcount\section@num
451 \section@num=0
452 \let\extensionchars=\empty
453 %
```

`\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.

```
454 \newif\ifnumbering
455 %
```

`\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.

```
456 \newcommand*{\beginnumbering}{%
457   \ifnumbering
458     \led@err@NumberingStarted
459     \endnumbering
460   \fi
461   \global\numberingtrue
462   \global\afterendnumberingRfalse%
463   \global\advance\section@num \@ne
464   \initnumbering@reg
465   \message{Section \the\section@num }%
466   \line@list@stuff{\jobname.\extensionchars\the\section@num}%
467   \ifcontinuousnumberingwithcolumns%
468     \unless\ifafterendnumberingR%
469       \unless\ifl@dpairing%
470         \ledRcoltrue%
471         \global\advance\section@numR \@ne%
472         \message{Section \the\section@numR R (continuousnumbering)}%
473         \line@list@stuffR{\jobname.\extensionchars\the\section@numR R}%
474         \ledRcolfalse%
475       \fi%
476     \fi%
477   \fi%
478   \l@dend@stuff
479   \setcounter{pstart}{1}
480   \ifl@dpairing
481     \global\l@dnumpstartsL \z@
482     \global\pst@rtedLfalse
483 %
```

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.

```

484 \else
485 \begingroup
486 \global\@afterindenttrue%In order to reestablish normal feature if the \
begingroup was not here
487 \initnumbering@quote
488 \ifwidthliketwocolumns%
489 \setwidthliketwocolumns%
490 \cuse{setpositionliketwocolumns@\columns@position}%
491 \fi%
492 \fi
493 \gdef\eled@sections@{ }%
494 \if@noeled@sec\else%
495 \makeatletter%
496 \InputIfFileExists%
497 {\l@auxdir\jobname.eledsec\the\section@num}%
498 {}%
499 {\led@warn@NoFile{\l@auxdir\jobname.eledsec\the\section@num}}%
500 \makeatother%
501 \immediate\openout\eled@sectioning@out=\l@auxdir\jobname.eledsec\the\
section@num\relax%
502 \fi%
503 }
504 \newcommand*\initnumbering@reg}{%
505 \global\pst@rtedLfalse
506 \global\l@dnumpststartsL \z@
507 \global\absline@num \z@
508 \gdef\normal@page@break{}
509 \gdef\l@prev@pb{}
510 \gdef\l@prev@nopb{}
511 \global\line@num \z@
512 \global\subline@num \z@
513 \global\@lock \z@
514 \global\sub@lock \z@
515 \global\sublines@false
516 \global\let\next@page@num=\relax
517 \global\let\this@section@next@page@num=\relax%
518 \global\let\sub@change=\relax
519 \global\last@page@num=-10000%
520 \ifdefined\line@numR%
521 \line@numR=\z@%
522 \last@page@numR=\z@%
523 \fi%
524 \resetprevline@
525 \resetprevpage@num
526 \global\stopmsdata@inserted@true%
527 \global\let\@msdata@list\relax%
528 \global\csundef{\msdata@\add@msd@c @data}%
529 }
530
531 %

```

`\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.

```

532 \def\endnumbering{%
533   \ifnumbering
534     \global\numberingfalse
535     \normal@pars
536     \ifnum\l@dnumpstartsL=0%
537       \led@err@NumberingWithoutPstart%
538     \fi%
539 %

```

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`.

```

540   \global\page@num=\this@section@page@num%
541   \global\last@page@num=\this@section@last@page@num%
542   \global\let\next@page@num\this@section@next@page@num%
543   \ifl@dpairing
544     \global\pst@rtedLfalse
545   \else
546     \ifx\insertlines@list\empty\else
547       \global\noteschanged@true
548     \fi
549     \ifx\line@list\empty\else
550       \global\noteschanged@true
551     \fi
552   \fi
553   \ifnoteschanged@
554     \led@mess@NotesChanged
555   \fi
556 \else
557   \led@err@NumberingNotStarted
558 \fi
559 \autoparfalse
560 \ifnoeled@sec\else%
561   \immediate\closeout\eled@sectioning@out%
562 \fi%
563 \ifl@dpairing\else
564   \global\l@dnumpstartsL=\z@%
565 \endgroup
566 \fi
567 }
568 %

```

`\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.

```

569 \newcount\pausenumbering@page@num%
570 \newcommand{\pausenumbering}{%

```

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

```

571 \ifx\this@section@next@page@num\relax%
572 \global\pausenumbering@page@num=0%
573 \else%
574 \global\pausenumbering@page@num=\this@section@next@page@num%
575 \fi%
576 \ifautopar\global\autopar@pausetrue\fi%
577 \endnumbering\global\numberingtrue}
578 %

```

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.

```

579 \newif\ifresumenumbering@start%
580 \newcommand*{\resumenumbering}{%
581 \ifnumbering
582 \ifautopar@pause\autopar\fi
583 \global\pst@rtedLtrue
584 \global\advance\section@num \@ne
585 \global\resumenumbering@starttrue%
586 \led@mess@SectionContinued{\the\section@num}%
587 \set@continuousnumberingforL%
588 \line@list@stuff{\jobname.\extensionchars\the\section@num}%
589 \ifcontinuousnumberingwithcolumns%
590 \unless\ifafterendnumberingR%
591 \unless\ifl@dpairing%
592 \ledRcoltrue%
593 \global\advance\section@numR \@ne%
594 \message{Section \the\section@numR R (continuousnumbering)}%
595 \line@list@stuffR{\jobname.\extensionchars\the\section@numR R}%
596 \ledRcolfalse%
597 \fi%
598 \fi%
599 \fi%
600 \l@dend@stuff
601 \ifl@dpairing\else%
602 \begin@group%
603 \initnumbering@quote%
604 \ifwidthliketwocolumns%
605 \setwidthliketwocolumns%
606 \csuse{setpositionliketwocolumns@\columns@position}%
607 \fi%
608 \fi%
609 \else
610 \led@err@NumberingShouldHaveStarted
611 \endnumbering
612 \beginnumbering
613 \fi}
614
615
616 %

```

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

```

617 \newcommand{\set@continuousnumberingforL}{%
618   \ifcontinuousnumberingwithcolumns%
619     \ifl@dpairing%
620       \unless\ifl@dpaging%
621         \global\c@pstartL=\c@pstart%
622         \global\c@pstartR=\c@pstart%
623       \fi%
624     \fi%
625   \fi%
626 }%
627 %

```

## 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.

```

628 \newcommand*{\list@create}[1]{%
629   \global\let#1=\empty%
630 }%
631 %

```

`\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 .

```

632 \newcommand*{\list@clear}[1]{%
633   \global\let#1=\empty%
634 }
635 %

```

`\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`.



```

636 \newtoks\led@toksa \newtoks\led@toksb
637 \global\led@toksa={\}
638 \long\def\xright@appenditem#1\to#2{%
639   \global\led@toksb=\expandafter{#2}%
640   \xdef#2{\the\led@toksb\the\led@toksa\expandafter{#1}}%
641   \global\led@toksb={}}
642 %

```

**\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`.

```

643 \long\def\xleft@appenditem#1\to#2{%
644   \global\led@toksb=\expandafter{#2}%
645   \xdef#2{\the\led@toksa\expandafter{#1}\the\led@toksb}%
646   \global\led@toksb={}}
647 %

```

**\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.

```

648 \def\gl@p#1\to#2{\expandafter\gl@poff#1\gl@poff#1#2}
649 \long\def\gl@poff\#1#2\gl@poff#3#4{\gdef#4{#1}\gdef#3{#2}}
650
651 %

```

## 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`

```

652 \newif\ifbypage@
653 \newif\ifbypstart@
654 %

```

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@R55 \newif\ifbypage@R
\ifbypstart@R56 \newif\ifbypstart@R
657 %

```

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

```
658 \newcommand*{\lineation}[1]{
659 %
```

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

```
660 \ifnumbering
661 \led@err@LineationInNumbered
662 \else
663 %
```

If the argument is page.

```
664 \def\@tempa{#1}\def\@tempb{page}%
665 \ifx\@tempa\@tempb
666 \global\bypage@true
667 \global\bystart@false
668 \unless\ifnocritical@%
669 \Xpstart[] [false]%
670 \fi%
671 %
```

If the argument is pstart.

```
672 \else
673 \def\@tempb{pstart}%
674 \ifx\@tempa\@tempb
675 \global\bypage@false
676 \global\bystart@true
677 \unless\ifnocritical@%
678 \Xpstart%
679 \fi%
680 %
```

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

```
681 \else
682 \def\@tempb{section}
683 \ifx\@tempa\@tempb
684 \global\bypage@false
685 \global\bystart@false
686 \unless\ifnocritical@%
687 \Xpstart[] [false]%
688 \fi%
689 %
```

In other case, it is an error.

```
690 \else
691 \led@warn@BadLineation
692 \fi
693 \fi
694 \fi
695 \fi}}
696 %
```

## 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.

```

697 \newcount\line@margin%
698 \newcount\line@margin@columns%Only for parallel typesetting
699 \line@margin@columns=\m@ne%
700
701 \newcommand*\linenummargin}[1]{%
702   \l@dgetline@margin{#1}%
703   \ifnum\@l@tempcntb>\m@ne
704     \ifledRcol
705       \global\line@marginR=\@l@tempcntb
706       \led@warn@setting@in@rightside{\linenummargin}%
707     \else
708       \global\line@margin=\@l@tempcntb
709     \fi
710   \fi}}
711
712 \newcommand*\l@dgetline@margin}[1]{%
713   \def\@tempa{#1}\def\@tempb{left}%
714   \ifx\@tempa\@tempb
715     \@l@tempcntb \z@
716   \else
717     \def\@tempb{right}%
718     \ifx\@tempa\@tempb
719       \@l@tempcntb \@ne
720     \else
721       \def\@tempb{outer}%
722       \ifx\@tempa\@tempb
723         \@l@tempcntb \tw@
724       \else
725         \def\@tempb{inner}%
726         \ifx\@tempa\@tempb
727           \@l@tempcntb \thr@@
728         \else
729           \led@warn@BadLinenummargin
730           \@l@tempcntb \m@ne
731         \fi
732       \fi
733     \fi
734   \fi}
735
736 %

```

## 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.

```

737 \newcounter{firstlinenum}
738   \setcounter{firstlinenum}{5}
739 \newcounter{linenumincrement}
740   \setcounter{linenumincrement}{5}
741 %

```

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

```

742 \newcounter{firstsublinenum}
743   \setcounter{firstsublinenum}{5}
744 \newcounter{sublinenumincrement}
745   \setcounter{sublinenumincrement}{5}
746
747 %

```

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

```

748 \newcommand*{\firstlinenum}[1]{%
749   \ifledRcol%
750     \setcounter{firstlinenumR}{#1}%
751     \led@warn@setting@in@rightside{\firstlinenum}%
752   \else%
753     \setcounter{firstlinenum}{#1}%
754   \fi%
755 }
756 \newcommand*{\linenumincrement}[1]{%
757   \ifledRcol%
758     \setcounter{linenumincrementR}{#1}%
759     \led@warn@setting@in@rightside{\linenumincrement}%
760   \else%
761     \setcounter{linenumincrement}{#1}%
762   \fi%
763 }
764 \newcommand*{\firstsublinenum}[1]{%
765   \ifledRcol%
766     \setcounter{firstsublinenumR}{#1}%
767     \led@warn@setting@in@rightside{\firstsublinenum}%
768   \else%
769     \setcounter{firstsublinenum}{#1}%
770   \fi%
771 }
772 \newcommand*{\sublinenumincrement}[1]{%
773   \ifledRcol%
774     \setcounter{sublinenumincrementR}{#1}%
775     \led@warn@setting@in@rightside{\sublinenumincrement}%
776   \else%
777     \setcounter{sublinenumincrement}{#1}%
778   \fi%
779 }
780 }
781

```

782 %

## 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.

```

783 \newcount\lock@disp
784 \newcommand{\lockdisp}[1]{%
785   \l@dgetlock@disp{#1}%
786   \ifnum\l@dtempcntb>\m@ne
787     \global\lock@disp=\l@dtempcntb
788   \else
789     \led@warn@BadLockdisp
790   \fi}}
791 \newcommand*{\l@dgetlock@disp}[1]{
792   \def\@tempa{#1}\def\@tempb{first}%
793   \ifx\@tempa\@tempb
794     \l@dtempcntb \z@
795   \else
796     \def\@tempb{last}%
797     \ifx\@tempa\@tempb
798       \l@dtempcntb \@ne
799     \else
800       \def\@tempb{all}%
801       \ifx\@tempa\@tempb
802         \l@dtempcntb \tw@
803       \else
804         \l@dtempcntb \m@ne
805       \fi
806     \fi
807   \fi}
808
809 %

```

`\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.

```

810 \newcount\sublock@disp
811 \newcommand{\sublockdisp}[1]{%
812   \l@dgetlock@disp{#1}%
813   \ifnum\l@dtempcntb>\m@ne
814     \global\sublock@disp=\l@dtempcntb
815   \else
816     \led@warn@BadSublockdisp
817   \fi}}
818
819 %

```

## 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`

```
820 \newcommand*{\linenumberstyle}[1]{%
821   \def\linenumrep##1{\@nameuse{#1}{##1}}
822 \newcommand*{\sublinenumberstyle}[1]{%
823   \def\sublinenumrep##1{\@nameuse{#1}{##1}}
824 %
```

Initialise the number styles to arabic.

```
825 \linenumberstyle{arabic}
826 \let\linenumr@p\linenumrep
827 \sublinenumberstyle{arabic}
828 \let\sublinenumr@p\sublinenumrep
829 %
830 %
```

## 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.

```
831 \newlength{\linenumsep}
832 \setlength{\linenumsep}{1pc}
833 \newcommand*{\numlabfont}{\normalfont\scriptsize}
834 \newcommand*{\ledlinenum}{%
835   \bgroup%
836   \ifluatex%
837     \textdir TLT%
838   \fi%
839   \numlabfont%
840   \unless\iflinenumannotationotherside%
841     \ifdefstring{\Xlinenumannotationposition@side}{before}{%
842       \l@wrap@ifnotemptybox{\Xwraplinenumannotation@side}{%
```

```

843     \csuse{annot@the\absline@num @the\section@num}%
844     }%
845   }%
846   \fi%
847   \linenumrep{\line@num}%
848   \ifsublines@
849     \ifnum\subline@num>0\relax
850       \unskip%
851       \Xsublinesep@side%
852       \sublinenumrep{\subline@num}%
853     \fi
854   \fi%
855   \unless\iflinenumannotationotherside%
856     \ifdefstring{\Xlinenumannotationposition@side}{after}{%
857       \l@wrap@ifnotemptybox{\Xwraplinenumannotation@side}{%
858         \csuse{annot@the\absline@num @the\section@num}%
859       }%
860     }%
861   \fi%
862   \egroup%
863 }%
864
865 \newcommand*{\leftlinenum}{%
866   \ledlinenum%
867   \kern\linenumsep%
868 }%
869 \newcommand*{\rightlinenum}{%
870   \kern\linenumsep%
871   \ledlinenum%
872 }%
873
874 %

```

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

```

875 \newcommand*{\leftlinenumannotation}{%
876   \l@wrap@ifnotemptybox{\Xwraplinenumannotation@side}{%
877     \csuse{annot@the\absline@num @the\section@num}%
878   }%
879   \kern\linenumsep%
880 }%
881 \newcommand*{\rightlinenumannotation}{%
882   \kern\linenumsep%
883   \l@wrap@ifnotemptybox{\Xwraplinenumannotation@side}{%
884     \csuse{annot@the\absline@num @the\section@num}%
885   }%
886 }%
887 %

```

## 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`.

```
888 \newcount\line@num
889 %
```

`\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.

```
890 \newcount\subline@num
891 %
```

`\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.

```
892 \newif\ifsublines@
893 \newif\ifsublines@R%
894 %
```

`\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.

```
895 \newcount\absline@num
896 %
```



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

## V.8 Line number locking counter

`\@lock`    The counts `\@lock` and `\sub@lock` tell us the state of line-number and sub-line-number locking. 0 means we are not within a locked set of lines; 1 means we are at the first line in the set; 2, at some intermediate line; and 3, at the last line.

```
897 \newcount\@lock
898 \newcount\sub@lock
899 %
```

## V.9 Line number associated to lemma

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

`\actionlines@list`  
`\actions@list`  
`\annot@list`

- `\line@list`: the page and line numbers for every lemma marked by `\edtext`. There are seven pieces of information, separated by vertical bars:

1. the starting page,
2. line, and
3. sub-line numbers, followed by the
4. ending page,
5. line, and
6. sub-line numbers, and then the
7. font specifier for the lemma.

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

```
23|35|0|24|3|0|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:

```

900 \list@create{\line@list}
901 \list@create{\insertlines@list}
902 \list@create{\actionlines@list}
903 \list@create{\actions@list}
904 \list@create{\annot@list}%
905 %

```

`\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
906 \newcount\page@num
907 \newcount\endpage@num
908 \newcount\endline@num
909 \newcount\endsubline@num
910 %

```

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

```

911 \newcount\this@section@page@num%
912 %

```

`\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.

```

913 \newif\ifnoteschanged@
914 %

```

`\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@15 \newcommand*\resetprevline@{%
916     \def\do##1{\global\csundef{prevline##1}}%
917     \dolistloop{\@series}%
918 }
919 %

```

`\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@20 \newcommand*{\resetprevpage@num}{%
921   \def\do##1{%
922     \ifcsdef{prevpage##1@num}{%
923       \global\csname prevpage##1@num\endcsname=\z@%
924       \global\csname prevpage##1@numR\endcsname=\z@%
925     }%
926   }%
927   \ifcsdef{##1prevpage@num}{%
928     \global\csname ##1prevpage@num\endcsname=\z@%
929     \global\csname ##1prevpage@numR\endcsname=\z@%
930   }%
931   }%
932 }%
933 \dolistloop{\@series}%
934 }
935 %

```

## 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.

```

936 \newread\@inputcheck
937 \newcommand*{\read@linelist}[1]{%
938   \ifledRcol%
939     \list@clearing@regR%
940   \else%
941     \list@clearing@reg%
942   \fi%
943 %

```

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).

```

944 \list@clear{\maxlinesinpar@list}
945 %

```

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.

```

946 \get@linelistfile{#1}%
947 \@stopmsd%Security if last \endms{} is forgotten
948 \unless\ifledRcol%Get the last line of the last page
949 \csnumgdef{@lastabsline@forpage@the\page@num}{\the\absline@num}%
950 \csnumgdef{@lastline@forpage@the\page@num}{\the\line@num}%
951 \csnumgdef{@lastsubline@forpage@the\page@num}{\the\subline@num}%
952 \else%
953 \csnumgdef{@lastabsline@forpageR@the\page@numR}{\the\absline@numR}%
954 \csnumgdef{@lastline@forpageR@the\page@numR}{\the\line@numR}%
955 \csnumgdef{@lastsubline@forpageR@the\page@numR}{\the\subline@numR}%
956 \fi%
957 \endgroup
958 %

```

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.

```

959 \ifledRcol
960 \global\page@numR=\m@ne
961 \ifx\actionlines@listR\empty
962 \gdef\next@actionlineR{1000000}%
963 \else
964 \gl@p\actionlines@listR\to\next@actionlineR
965 \gl@p\actions@listR\to\next@actionR
966 \fi
967 \else
968 \global\page@num=\m@ne
969 \ifx\actionlines@list\empty
970 \gdef\next@actionline{1000000}%
971 \else
972 \gl@p\actionlines@list\to\next@actionline
973 \gl@p\actions@list\to\next@action
974 \fi
975 \fi
976 }
977 %

```

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

```

978 \newcommand*{\list@clearing@reg}{%
979 \list@clear{\line@list}%
980 \list@clear{\insertlines@list}%
981 \list@clear{\actionlines@list}%
982 \list@clear{\actions@list}%
983 \list@clear{\linesinpar@listL}%
984 \list@clear{\linesonpage@listL}%
985 }%

```

```
986 %
```

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

```
987 \newcommand*\get@linelistfile}[1]{%
988   \InputIfFileExists{\l@auxdir#1}{%
989     \global\noteschanged@false
990     \begingroup
991       \catcode`\[=1 \catcode`\]=2
992       \makeatletter \catcode`\^M=9}{%
993     \led@warn@NoFile{\l@auxdir#1}%
994     \global\noteschanged@true
995     \begingroup}%
996   }
997
998 %
```

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. 20 above).

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

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

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

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

`\line@list@version` The `\line@list@version` check if the line-list file does not 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.

```
999 \newcommand{\line@list@version}[1]{%
1000   \IfStrEq{#1}{\this@line@list@version}%
1001   }%
1002   {\ifledRcol%
1003     \led@warn@Obsolete{\jobname.\extensionchars\the\section@num}%
```

```

1004     \else%
1005         \led@warn@Obsolete{\jobname.\extensionchars\the\section@num}%
1006     \fi%
1007     \endinput%
1008 }%
1009 }%
1010 %

```

`\@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`.

```

1011 \newcommand*{\@nl}[2]{%
1012     \@page{#1}%
1013     \ifledRcol%
1014         \@nl@regR%
1015         \csnumgdef{@page@forlineR@the\section@numR @the\absline@numR}{#1}%
1016     \else%
1017         \@nl@reg%
1018     \fi%
1019 }
1020 }
1021 \newcommand*{\@nl@reg}{%
1022     \ifx\l@dchset@num\relax \else
1023         \advance\absline@num \@ne
1024         \csgdef{l@dchset@num@the\absline@num}{}%To remember this line have
1025         \set@line@action
1026         \let\l@dchset@num=\relax
1027         \advance\absline@num \m@ne
1028         \advance\line@num \m@ne
1029     \fi
1030 %

```

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

```

1031     \reset@current@annot%
1032     \advance\absline@num \@ne
1033         \ifx\next@page@num\relax \else
1034             \page@action
1035             \let\next@page@num=\relax
1036     \fi

```

```

1037     \ifx\sub@change\relax \else
1038       \ifnum\sub@change>\z@
1039         \sublines@true
1040       \else
1041         \sublines@false
1042       \fi
1043       \sub@action
1044       \let\sub@change=\relax
1045     \fi
1046 %

```

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

```

1047     \ifcase\@lock
1048       \or
1049         \@lock \tw@
1050       \or \or
1051         \@lock \z@
1052     \fi
1053     \ifcase\sub@lock
1054       \or
1055         \sub@lock \tw@
1056       \or \or
1057         \sub@lock \z@
1058     \fi
1059 %

```

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

```

1060     \ifsublines@
1061       \ifnum\sub@lock<\tw@
1062         \advance\subline@num \@ne
1063       \fi
1064     \else
1065       \ifnum\@lock<\tw@
1066         \advance\line@num \@ne \subline@num \z@
1067       \fi
1068     \fi}
1069 %
1070 %

```

`\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`

```

1071 \newcount\last@page@num
1072 \last@page@num=-10000
1073
1074 \newcount\this@section@last@page@num%
1075 \this@section@last@page@num=-10000%
1076
1077 \newcommand*{\@page}[1]{%
1078   \ifledRcol
1079     \ifnum #1=\last@page@numR
1080     \else
1081       \csnumgdef{@lastabsline@forpageR@the\page@numR}{\the\absline@numR}%

```



```

1082     \csnumgdef{@lastline@forpageR@the\page@numR}{\the\line@numR}%
1083     \csnumgdef{@lastsubline@forpageR@the\page@numR}{\the\subline@numR}%
1084     \ifbypage@R
1085     \letcs{\prevsection@last@page@num}{last@page@num@the\numexpr\
section@num-1\relax}%
1086     \ifundef{\prevsection@last@page@num}{\def\prevsection@last@page@num
{-999}}{}%
1087     \ifx\l@dchset@num\relax%Not resetting if preceded by a \setlinenum
1088     \ifboolexpr{%
1089     (bool{continuousnumberingwithcolumns} or bool{
resumenumberingR@start})%
1090     and test {\ifnumequal{\last@page@numR}{-10000}}%
1091     and not (%
1092     bool{continuousnumberingwithcolumns}%
1093     and test {\ifnumcomp{\prevsection@last@page@num}{<}{#1}}%
1094     )%
1095     }%
1096     {}%
1097     {%
1098     \line@numR \z@%
1099     \subline@numR \z@%
1100     \global\csdef{reset@lineR\the\numexpr\absline@numR+1\relax @\
the\section@numR}{}%
1101     }%
1102     \fi%
1103     \fi
1104     \page@numR=#1%
1105     \global\this@section@page@numR=#1%
1106     \global\last@page@numR=#1%
1107     \global\this@section@last@page@numR=#1%
1108     \def\next@page@numR{#1}%
1109     \gdef\this@section@next@page@numR{#1}%
1110     \fi
1111     \else
1112     \ifnum #1=\last@page@num
1113     \else
1114     \csnumgdef{@lastabsline@forpage@the\page@num}{\the\absline@num}%
1115     \csnumgdef{@lastline@forpage@the\page@num}{\the\line@num}%
1116     \csnumgdef{@lastsubline@forpage@the\page@num}{\the\subline@num}%
1117     \ifbypage@
1118     \ifx\l@dchset@num\relax%Not resetting if preceded by a \setlinenum
1119     \line@num \z@%
1120     \subline@num \z@%
1121     \global\csdef{reset@line\the\numexpr\absline@num+1\relax @the\
section@num}{}%
1122     \fi%
1123     \fi
1124     \page@num=#1%
1125     \global\this@section@page@num=#1%
1126     \last@page@num=#1%
1127     \global\this@section@last@page@num=#1%
1128     \global\csdef{last@page@num@the\section@num}{#1}%
1129     \gdef\next@page@num{#1}%
1130 %

```

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.

```

1131 \gdef\this@section@next@page@num{#1}%
1132 \listxadd{\normal@page@break}{\the\absline@num}
1133 \fi
1134 \fi}
1135 %

```

`\@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.

```

1136 \newcommand*\@pend}[1]{%
1137 \newcommand*\@pendR}[1]{%
1138 \newcommand*\@lopL}[1]{%
1139 \newcommand*\@lopR}[1]{%
1140
1141 %

```

`\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.

```

1142 \newcommand*\sub@on{%
1143 \ifboolexpr{%
1144   bool{sublines@}%
1145   or%
1146   (bool{sublines@R} and bool{ledRcol})%
1147 }{%
1148   \let\sub@change=\relax
1149 }{%
1150   \def\sub@change{-1}%
1151 }%
1152 }%
1153 \newcommand*\sub@off{%
1154 \ifboolexpr{%
1155   bool{sublines@}%
1156   or%
1157   (bool{sublines@R} and bool{ledRcol})%
1158 }{%
1159   \def\sub@change{-1}%
1160 }{%
1161   \let\sub@change=\relax
1162 }%
1163 }%
1164
1165 %

```

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

```

1166
1167 \newcommand*\@adv}[1]{%

```

```

1168 \ifsublines@
1169 \ifledRcol
1170 \advance\subline@numR by #1\relax
1171 \ifnum\subline@numR<\z@
1172 \led@warn@BadAdvancelineSubline
1173 \subline@numR \z@
1174 \fi
1175 \else
1176 \advance\subline@num by #1\relax
1177 \ifnum\subline@num<\z@
1178 \led@warn@BadAdvancelineSubline
1179 \subline@num \z@
1180 \fi
1181 \fi
1182 \else
1183 \ifledRcol
1184 \advance\line@numR by #1\relax
1185 \ifnum\line@numR<\z@
1186 \led@warn@BadAdvancelineLine
1187 \line@numR \z@
1188 \fi
1189 \else
1190 \advance\line@num by #1\relax
1191 \ifnum\line@num<\z@
1192 \led@warn@BadAdvancelineLine
1193 \line@num \z@
1194 \fi
1195 \fi
1196 \fi
1197 \set@line@action}
1198
1199 %

```

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

```

1200
1201 \newcommand*{\@set}[1]{%
1202 \ifledRcol
1203 \ifsublines@R%
1204 \subline@numR=#1\relax
1205 \else
1206 \line@numR=#1\relax
1207 \fi
1208 \set@line@action
1209 \else
1210 \ifsublines@
1211 \subline@num=#1\relax
1212 \else
1213 \line@num=#1\relax
1214 \fi
1215 \set@line@action
1216 \fi}
1217

```

1218 %

`\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.

```
1219
1220 \newcommand*{\l@d@set}[1]{%
1221   \ifledRcol
1222     \line@numR=#1\relax
1223     \advance\line@numR \@ne
1224     \def\l@dchset@num{#1}
1225   \else
1226     \line@num=#1\relax
1227     \advance\line@num \@ne
1228     \def\l@dchset@num{#1}
1229   \fi}
1230 \let\l@dchset@num\relax
1231
1232 %
```

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

```
1233
1234 \newcommand*{\page@action}{%
1235   \ifledRcol
1236     \xright@appenditem{\the\absline@numR}\to\actionlines@listR
1237     \xright@appenditem{\next@page@numR}\to\actions@listR
1238   \else
1239     \xright@appenditem{\the\absline@num}\to\actionlines@list
1240     \xright@appenditem{\next@page@num}\to\actions@list
1241   \fi}
1242 %
```

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

```
1243
1244 \newcommand*{\set@line@action}{%
1245   \ifledRcol
1246     \xright@appenditem{\the\absline@numR}\to\actionlines@listR
1247     \ifsublines@R%
1248       \@l@tempcnta=-\subline@numR
1249     \else
1250       \@l@tempcnta=-\line@numR
1251     \fi
1252     \advance\@l@tempcnta by -5000\relax
1253     \xright@appenditem{\the\@l@tempcnta}\to\actions@listR
1254   \else
1255     \xright@appenditem{\the\absline@num}\to\actionlines@list
1256     \ifsublines@
1257       \@l@tempcnta=-\subline@num
1258     \else
```

```

1259     \@l@dtempcnta=-\line@num
1260     \fi
1261     \advance\@l@dtempcnta by -5000\relax
1262     \xright@appenditem{\the\@l@dtempcnta}\to\actions@list
1263     \fi}
1264 %

```

**\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.

```

1265 \newcommand*\sub@action){%
1266 \ifledRcol
1267 \xright@appenditem{\the\absline@numR}\to\actionlines@listR
1268 \ifsublines@R%
1269 \xright@appenditem{-1001}\to\actions@listR
1270 \else
1271 \xright@appenditem{-1002}\to\actions@listR
1272 \fi
1273 \else
1274 \xright@appenditem{\the\absline@num}\to\actionlines@list
1275 \ifsublines@
1276 \xright@appenditem{-1001}\to\actions@list
1277 \else
1278 \xright@appenditem{-1002}\to\actions@list
1279 \fi
1280 \fi}
1281 \fi}
1282 %

```

**\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.

```

1283 \newcommand*\lock@on){\futurelet\next\do@lockon}
1284
1285 \newcommand*\do@lockon){%
1286 \ifx\next\lock@off
1287 \global\let\lock@off=\skip@lockoff
1288 \else
1289 \ifledRcol
1290 \do@lockonR
1291 \else
1292 \do@lockonL
1293 \fi
1294 \fi}
1295
1296
1297 \newcommand*\do@lockonL){%
1298 \xright@appenditem{\the\absline@num}\to\actionlines@list
1299 \ifsublines@

```

```

1300 \xright@appenditem{-1005}\to\actions@list
1301 \ifnum\sub@lock=\z@
1302   \sub@lock \@ne
1303 \else
1304   \ifnum\sub@lock=\thr@@
1305     \sub@lock \@ne
1306   \fi
1307 \fi
1308 \else
1309   \xright@appenditem{-1003}\to\actions@list
1310   \ifnum\@lock=\z@
1311     \@lock \@ne
1312   \else
1313     \ifnum\@lock=\thr@@
1314       \@lock \@ne
1315     \fi
1316   \fi
1317 \fi}
1318
1319 %

```

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

```

\do@lockoff
\do@lockoffL
\skip@lockoff
1320 \newcommand*{\do@lockoffL}{%
1321 \xright@appenditem{\the\absline@num}\to\actionlines@list
1322 \ifsublines@
1323   \xright@appenditem{-1006}\to\actions@list
1324   \ifnum\sub@lock=\tw@
1325     \sub@lock \thr@@
1326   \else
1327     \sub@lock \z@
1328   \fi
1329 \else
1330   \xright@appenditem{-1004}\to\actions@list
1331   \ifnum\@lock=\tw@
1332     \@lock \thr@@
1333   \else
1334     \@lock \z@
1335   \fi
1336 \fi}
1337
1338 \newcommand*{\do@lockoff}{%
1339   \reset@current@annot%
1340   \ifledRcol
1341     \do@lockoffR
1342   \else
1343     \do@lockoffL
1344   \fi}
1345 \newcommand*{\skip@lockoff}{\global\let\lock@off=\do@lockoff}
1346 \global\let\lock@off=\do@lockoff
1347
1348 %

```

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

```

1349 \newcommand*{\n@num}{%
1350   \ifledRcol%
1351     \xright@appenditem{\the\absline@numR}\to\actionlines@listR
1352     \xright@appenditem{-1007}\to\actions@listR
1353   \else%
1354     \xright@appenditem{\the\absline@num}\to\actionlines@list%
1355     \xright@appenditem{-1007}\to\actions@list%
1356   \fi%
1357 }%
1358
1359 %

```

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

```

1360 \newcommand*{\n@num@stanza}{%
1361   \ifledRcol%
1362     \xright@appenditem{\the\absline@numR}\to\actionlines@listR%
1363     \xright@appenditem{-1008}\to\actions@listR%
1364   \else%
1365     \xright@appenditem{\the\absline@num}\to\actionlines@list%%
1366     \xright@appenditem{-1008}\to\actions@list%
1367   \fi%
1368 }
1369 %

```

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

```

\hidenumberingonleftpage
\hidenumberingonrightpage
1370 \newif\ifl@dhidenumber
1371 \newcommand*{\hidenumbering}{%
1372   \ifledRcol%
1373     \write\linenum@outR{\string\hide@num}%
1374   \else%
1375     \write\linenum@out{\string\hide@num}%
1376   \fi%
1377 }%
1378 \newcommand*{\hide@num}{%
1379   \ifledRcol%
1380     \xright@appenditem{\the\absline@numR}\to\actionlines@listR%
1381     \xright@appenditem{-1009}\to\actions@listR%
1382   \else%
1383     \xright@appenditem{\the\absline@num}\to\actionlines@list%%
1384     \xright@appenditem{-1009}\to\actions@list%
1385   \fi%
1386 }
1387 \newcommand*{\hidenumberingonleftpage}{%
1388   \ifledRcol%
1389     \write\linenum@outR{\string\hide@num@left}%
1390   \else%
1391     \write\linenum@out{\string\hide@num@left}%
1392   \fi%

```

```

1393 }%
1394
1395 \newcommand*{\hide@num@left}{%
1396   \ifledRcol%
1397     \ifodd\page@numR\else%
1398       \xright@appenditem{\the\absline@numR}\to\actionlines@listR%
1399       \xright@appenditem{-1009}\to\actions@listR%
1400     \fi%
1401   \else%
1402     \ifodd\page@num\else%
1403       \xright@appenditem{\the\absline@num}\to\actionlines@list%%
1404       \xright@appenditem{-1009}\to\actions@list%
1405     \fi%
1406   \fi%
1407 }%
1408
1409 \newcommand*{\hidenumberingonrightpage}{%
1410   \ifledRcol%
1411     \write\linenum@outR{\string\hide@num@right}%
1412   \else%
1413     \write\linenum@out{\string\hide@num@right}%
1414   \fi%
1415 }%
1416
1417 \newcommand*{\hide@num@right}{%
1418   \ifledRcol%
1419     \ifodd\page@numR%
1420       \xright@appenditem{\the\absline@numR}\to\actionlines@listR%
1421       \xright@appenditem{-1009}\to\actions@listR%
1422     \fi%
1423   \else%
1424     \ifodd\page@num%
1425       \xright@appenditem{\the\absline@num}\to\actionlines@list%%
1426       \xright@appenditem{-1009}\to\actions@list%
1427     \fi%
1428   \fi%
1429 }%
1430
1431 %

```

`\@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`.

```

1432 \newcount\insert@count
1433 %

```

- #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.

```
1434 \newcommand*\@dummy@ref}[2]{#2}
1435 %
```

`\@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.

```
1436 \newcommand*\@ref}[2]{%
1437   \ifledRcol%
1438     \@ref@regR{#1}{#2}%
1439   \else%
1440     \@ref@reg{#1}{#2}%
1441   \fi%
1442 }%
1443 \newcommand*\@ref@reg}[2]{%
1444   \global\insert@count=#1\relax
1445   \global\advance\@edtext@level by 1%
1446   \loop\ifnum\insert@count>\z@
1447     \xright@appenditem{\the\absline@num}\to\insertlines@list
1448     \global\advance\insert@count \m@ne
1449   \repeat
1450 %
```

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.

```
1451 \begingroup
1452   \let\@ref=\@dummy@ref
1453   \let\@lopL\@gobble
1454   \let\page@action=\relax
1455   \let\sub@action=\relax
1456   \let\set@line@action=\relax
1457   \let\@lab=\relax
1458   \let\@lemma=\relax%
1459   \let\@sw\@gobblethree%
1460   \let\store@annot@to@absline\@gobble%
1461   #2
1462   \global\endpage@num=\page@num
1463   \global\endline@num=\line@num
1464   \global\endsubline@num=\subline@num
1465   \global\let\endcurrent@annot=\current@annot%
1466 \endgroup
1467 %
```

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

```
1468 \xright@appenditem%
1469   {\the\page@num|\the\line@num|%
1470   \ifsublines@ \the\subline@num \else 0\fi|%
1471   \the\endpage@num|\the\endline@num|%
1472   \ifsublines@ \the\endsubline@num \else 0\fi}\to\line@list
```

```

1473 \xright@appenditem%
1474   {\current@annot|\endcurrent@annot}\to\annot@list%
1475 %

```

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

```

1476   \@ref@reg@parse{#2}%
1477 %

```

Decrease edtext level counter.

```

1478   \global\advance\@edtext@level by -1%
1479 }
1480 %

```

`\@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.

```

1481 \newcommand{\@ref@reg@parse}[1]{%
1482   \expandafter\list@create\expandafter{\csname sw@list@edtext@tmp@the\
1483   \@edtext@level\endcsname}%
1484   \providebool{lemmacommand@the\@edtext@level}%
1485   \boolfalse{lemmacommand@the\@edtext@level}%
1486 %

```

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

```

1486   #1%
1487 %

```

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.

```

1488   \ifnum\@edtext@level>0%
1489     \def\create@this@edtext@level{\expandafter\list@create\expandafter{\
1490     csname sw@list@edtext@the\@edtext@level\endcsname}}%
1491     \ifcsundef{sw@list@edtext@the\@edtext@level}{\create@this@edtext@level
1492     }{}%
1493     \letcs{\@tmp}{sw@list@edtext@the\@edtext@level}%
1494     \letcs{\@tmpp}{sw@list@edtext@tmp@the\@edtext@level}
1495     \xright@appenditem{\expandonce\@tmpp}\to\@tmp%
1496     \global\cslet{sw@list@edtext@the\@edtext@level}{\@tmp}%
1497   \fi%
1498 %
1499 %

```

`\@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.

```

1500 \newcommand{\@ref@later}[1]{%
1501   \global\advance\@edtext@level by \@ne%
1502   \ifledRcol%
1503     \@ref@reg@parseR{#1}%
1504   \else%
1505     \@ref@reg@parse{#1}%
1506   \fi%
1507   \global\advance\@edtext@level by -\@ne%
1508 }%
1509 %

```

## 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`.

```

1510 \newwrite\linenum@out
1511 %

```

`\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`.

```

1512 \newif\iffirst@linenum@out@
1513   \first@linenum@out@true
1514 %

```

`\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.

```

1515 \newcommand{\this@line@list@version}{8}%
1516 %

```

`\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`

```

1517 \let\next@line@list@stuff\relax%
1518 \newcommand*{\line@list@stuff}[1]{%
1519 %

```

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

```

1520 \global\newtoggle{notfirstrun@#1}%
1521 \IfFileExists{\l@auxdir#1}%
1522   {\global\toggletrue{notfirstrun@#1}}%
1523   {\global\togglefalse{notfirstrun@#1}}%
1524 %

```

A internal hook (not used yet).

```

1525 \next@line@list@stuff%
1526 \global\let\next@line@list@stuff\relax%
1527 %

```

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

```

1528 \read@linelist{#1}%
1529 %

```

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  $n$ .

```

1530 \iffirst@linenum@out@
1531   \global\first@linenum@out@false%
1532   \immediate\openout\linenum@out=\l@auxdir#1\relax%
1533   \immediate\write\linenum@out{\string\line@list@version{\
this@line@list@version}}%
1534   \ifl@dpaging%
1535     \immediate\write\linenum@out{\string\@par@sync@option{\
@par@this@sync@option}}%
1536   \fi%
1537 \else
1538 %

```

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.

```

1539   \if@minipage%
1540     \leavevmode%
1541     \fi%
1542     \closeout\linenum@out%
1543     \openout\linenum@out=\l@auxdir#1\relax%
1544     \write\linenum@out{\string\line@list@version{\this@line@list@version}}
%
1545     \ifl@dpaging%
1546       \write\linenum@out{\string\@par@sync@option{\@par@this@sync@option}}
%
1547     \fi%
1548   \fi%
1549 }%

```

```
1550
1551 %
```

`\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.

```
1552 \newcommand*{\new@line}{%
1553   \ifnumberline%
1554     \IfStrEq{\led@pb@setting}{after}%
1555       {\xifinlist{\the\absline@num}{\l@prev@nopb}%
1556         {\xifinlist{\the\absline@num}{\normal@page@break}%
1557           {\numgdef{\@next@page}{\c@par@page+\@ne}%
1558             \write\linenum@out{\string\@nl[\@next@page] [\@next@page]}}%
1559           }%
1560         {\write\linenum@out{\string\@nl[\the\@this@page] [\thepage]}}%
1561       }%
1562     {\write\linenum@out{\string\@nl[\the\@this@page] [\thepage]}}%
1563   {}%
1564   \IfStrEq{\led@pb@setting}{before}%
1565     {\numdef{\next@absline}{\the\absline@num+\@ne}%
1566     \xifinlist{\next@absline}{\l@prev@nopb}%
1567       {\xifinlist{\the\absline@num}{\normal@page@break}%
1568         {\numgdef{\nc@page}{\c@par@page+\@ne}%
1569           \write\linenum@out{\string\@nl[\nc@page] [\nc@page]}}%
1570         }%
1571       {\write\linenum@out{\string\@nl[\the\@this@page] [\thepage]}}%
1572     }%
1573     {\write\linenum@out{\string\@nl[\the\@this@page] [\thepage]}}%
1574   }%
1575   {}%
1576   \IfStrEqCase{\led@pb@setting}%
1577     {%
1578     {before}{\relax}%
1579     {after}{\relax}%
1580   }[%
1581     \write\linenum@out{\string\@nl[\the\@this@page] [\thepage]}}%
1582   ]%
1583 \fi%
1584 }
```

```

1585
1586 \newcount\@this@cc@page%%
1587 \newcommand{\set@this@cc@page}{%
1588   \ifboolexpr{%
1589     bool{sameparallelpagenunder}%
1590     or bool{prevpgnotnumbered}%
1591   }%
1592   {\global\@this@cc@page=\c@par@page}%
1593   {\global\@this@cc@page=\c@page}%
1594 }%
1595 %

```

`\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.

```

1596
1597 \newcommand*{\flag@start}{%
1598   \ifledRcol%
1599     \edef\next{\write\linenum@outR{%
1600       \string\@ref[\the\insert@countR] []}%
1601   }%
1602   \else%
1603     \edef\next{\write\linenum@out{%
1604       \string\@ref[\the\insert@count] []}%
1605   }%
1606   \fi%
1607   \next%
1608 }%
1609
1610 \newcommand*{\flag@end}{%
1611   \ifledRcol%
1612     \write\linenum@outR{]}%
1613   \else%
1614     \write\linenum@out{]}%
1615   \fi%
1616 }%
1617 %

```

`\flag@start@RTL` With  $\text{Xe}_{\text{L}}\text{TeX}$ , there is a problem when using RTL: the writing of a command in the numbered auxiliary files (.1, .2 etc) is reversed when the first argument of `\edtext` is typeset in one line, but it is **not** reversed when this first argument is typeset in two lines or more.<sup>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  $\text{Xe}_{\text{L}}\text{TeX}$  manages right-to-left typesetting. David Carlisle explains it on <http://tex.stackexchange.com/a/333373/7712> and provides a potential solution, using `\vadjust`. However in some cases this adds spurious vertical spaces in `reledmac`. That is why we are using the solution explained below.

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.

```

1618 \newcommand{\flag@start@RTL}{%
1619   \edlabel{edtext:start:\csuse{thisedtext@the\@edtext@level}}%
1620   \IfStrEq{\xabslineref{edtext:start:\csuse{thisedtext@the\@edtext@level}}%
1621     {000}%
1622     {}}%
1623     {%
1624       \ifnumequal%
1625         {\xabslineref{edtext:start:\csuse{thisedtext@the\@edtext@level}}}%
1626         {\xabslineref{edtext:end:\csuse{thisedtext@the\@edtext@level}}}%
1627         {\flag@end}%
1628         {\flag@start}%
1629       }%
1630     }%
1631 }%
1632 \newcommand{\flag@end@RTL}{%
1633   \edlabel{edtext:end:\csuse{thisedtext@the\@edtext@level}}%
1634   \IfStrEq{\xabslineref{edtext:start:\csuse{thisedtext@the\@edtext@level}}%
1635     {000}%
1636     {}}%
1637     {%
1638       \ifnumequal%
1639         {\xabslineref{edtext:start:\csuse{thisedtext@the\@edtext@level}}}%
1640         {\xabslineref{edtext:end:\csuse{thisedtext@the\@edtext@level}}}%
1641         {\flag@start}%
1642         {\flag@end}%
1643       }%
1644     }%
1645   %

```

`\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

```

1646 \newcommand*{\flag@start@later}{%
1647   \ifledRcol%
1648     \write\linenum@outR{\string\@ref@later []}%
1649   \else%
1650     \write\linenum@out{\string\@ref@later []}%
1651   \fi%
1652 }%
1653 \newcommand{\flag@end@later}{%
1654   \ifledRcol%
1655     \write\linenum@outR[]%
1656   \else%

```

```

1657 \write\linenum@out{]}%
1658 \fi%
1659 }
1660 %

```

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

We tinker with `\lastskip` because a command of either sort really needs to be attached to the last word preceding the change, not the first word that follows the change. This is because sub-lineation will often turn on and off in mid-line—stage directions, for example, often are mixed with dialogue in that way—and when a line is mixed we want to label it using the system that was in effect at its start. But when sub-lineation begins at the very start of a line we have a problem, if we don't put in this code.

```

1661
1662
1663 \newcommand*{\startsub}{%
1664 \dimen0\lastskip%
1665 \ifdim\dimen0>0pt%
1666 \unskip%
1667 \fi%
1668 \ifledRcol%
1669 \write\linenum@outR{\string\sub@on}%
1670 \else%
1671 \write\linenum@out{\string\sub@on}%
1672 \fi%
1673 \ifdim\dimen0>0pt%
1674 \hskip\dimen0%
1675 \fi%
1676 }%
1677
1678 \def\endsub{%
1679 \dimen0\lastskip%
1680 \ifdim\dimen0>0pt%
1681 \unskip%
1682 \fi%
1683 \ifledRcol%
1684 \write\linenum@outR{\string\sub@off}%
1685 \else%
1686 \write\linenum@out{\string\sub@off}%
1687 \fi%
1688 \ifdim\dimen0>0pt%
1689 \hskip\dimen0%
1690 \fi%
1691 }%
1692
1693 %

```

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



```

1694 \newcommand*{\advanceline}[1]{\leavevmode%
1695   \ifledRcol \write\linenum@outR{\string\@adv[#1]}%
1696   \else      \write\linenum@out{\string\@adv[#1]}%
1697   \fi%
1698 }
1699 %

```

**\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.

```

1700
1701 \newcommand*{\setline}[1]{%
1702   \leavevmode%
1703   \ifnum#1<\z@
1704     \led@warn@BadSetline%
1705   \else%
1706     \ifledRcol%
1707       \write\linenum@outR{\string\@set[#1]}%
1708     \else%
1709       \write\linenum@out{\string\@set[#1]}%
1710     \fi%
1711   \fi%
1712 }%
1713
1714 %

```

**\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.

```

1715
1716 \newcommand*{\setlinenum}[1]{%
1717   \ifnum#1<\z@%
1718     \led@warn@BadSetlinenum%
1719   \else%
1720     \ifledRcol%
1721       \write\linenum@outR{\string\l@d@set[#1]}%
1722     \else%
1723       \write\linenum@out{\string\l@d@set[#1]}%
1724     \fi%
1725   \fi%
1726 }%
1727
1728 %

```

**\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.

```

1729
1730 \newcommand*{\startlock}{%
1731   \ifledRcol \write\linenum@outR{\string\lock@on}%
1732   \else      \write\linenum@out{\string\lock@on}%
1733   \fi}
1734 \def\endlock{}

```

```

1735 \ifledRcol \write\linenum@outR{\string\lock@off}%
1736 \else      \write\linenum@out{\string\lock@off}%
1737 \fi}
1738 %

```

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

```

\l@dskipnumbertrue
\l@dskipnumberfalse
\skipnumbering
1739 \newif\ifl@dskipnumber
1740 \newif\ifl@dskipversenumber%
1741 \newcommand*{\skipnumbering}{%
1742 \leavevmode%
1743 \ifledRcol%
1744 \ifinstanza%
1745 \write\linenum@outR{\string\n@num@stanza}%
1746 \else%
1747 \write\linenum@outR{\string\n@num}%
1748 \fi%
1749 \advanceline{-1}%
1750 \else%
1751 \ifinstanza%
1752 \write\linenum@out{\string\n@num@stanza}%
1753 \else%
1754 \write\linenum@out{\string\n@num}%
1755 \fi%
1756 \advanceline{-1}%
1757 \fi%
1758 }%
1759 %
1760 %

```

## 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.

```
1761 \list@create{\end@lemmas}
1762 %
```

`\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.

```
1763 \newcommand{\dummy@edtext}[2]{#1}
1764 %
```

`\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`.

```
1765 \newcommand{\dummy@edtext@showlemma}[2]{\showlemma{#1}}%
1766 %
```

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>.)

```

1767 \newcommand*{\noexpands}{%
1768   \let\select@lemmafnt=0%
1769   \let\startsub=\relax \let\endsub=\relax
1770   \let\startlock=\relax \let\endlock=\relax
1771   \let\edlabel=\@gobble
1772   \let\setline=\@gobble \let\advanceline=\@gobble
1773   \let\sameword\sameword@inedtext%
1774   \let\edtext=dummy@edtext
1775   \let\edindex\dummy@edindex%
1776   \l@dtabnoexpands
1777   \l@noexpands@edgl%
1778   \let\linenumnotation=\@gobble%
1779   \morenoexpands}
1780 \let\morenoexpands=\relax
1781
1782 %

```

`\@tag` Now, we define an empty `\@tag` command. It will be redefine by `\edtext`: its value is the first argument. It will be used by the `\Xfootnote` commands.

```

1783 \newcommand{\@tag}{}
1784 %

```

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

```

1785 \newcount\@edtext@level%
1786 \@edtext@level=0%
1787 %

```

`\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.

```

1788 \newif\if@edtext@secondarg@%
1789 %

```

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

```

1790 \newcounter{edtext}
1791 \renewcommand{\theedtext}{edtxt@the\c@edtext}%
1792 %

```

`\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`

```

1793 \newcount\edtext@notes%
1794 \newcount\edtext@notes@start%
1795 %

```

`\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.

```

1796 \newif\if@noneed@notes%
1797 %

```

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

```

1798 \newcommand{\edtext}[2]{\leavevmode%
1799 %

```

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

```

1800 \ifnumberedpar%
1801 %

```

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

```

1802 \@check@edtext@args{#1}{#2}%
1803 %

```

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

```

1804 \global\advance\@edtext@level by 1%
1805 %

```

We let `\edtext@notes@start` equal to `\edtext@notes` to check later the number of notes associated to the edtext.

```

1806 \global\edtext@notes@start=\edtext@notes%
1807 %

```

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  $X_{\text{right}}\TeX$ .

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`.

```

1808 \stepcounter{edtext}%
1809 \csxdef{thisedtext@the\@edtext@level}{\theedtext}%
1810 %

```

By default, we do not use `\lemma`

```

1811 \global\@lemmacommand@false%
1812 %

```

```

1813 \begingroup%
1814 %

```

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`.

```

1815     \ifledRcol%
1816         \ifcsvoid{sw@list@edtextR@the\@edtext@level}%
1817             {\global\let\sw@inthisedtext\empty}%
1818             {\expandafter\gl@p\csname sw@list@edtextR@the\
@edtext@level\endcsname\to\sw@inthisedtext}%
1819         \else%
1820             \ifcsvoid{sw@list@edtext@the\@edtext@level}%
1821                 {\global\let\sw@inthisedtext\empty}%
1822                 {\expandafter\gl@p\csname sw@list@edtext@the\@edtext@level
\endcsname\to\sw@inthisedtext}%
1823         \fi%
1824 %

```

`\@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.

```

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

```

`\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`.

```

1829     \set@line%
1830 %

```

`\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`.

```

1831     \ifledRcol \global\insert@countR \z@%
1832     \else      \global\insert@count \z@ \fi%
1833 %

```

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.

```

1834     \@edtext@secondarg@true%
1835     \ignorespaces #2\relax%
1836     \@edtext@secondarg@false%
1837 %

```

Check if we indeed insert footnotes or critical notes.

```

1838     \ifnum\edtext@notes=\edtext@notes@start%
1839     \unless\if@noneed@notes%

```

```

1840     \led@err@EdtextWithoutNote{#1}%
1841     \fi%
1842     \fi%
1843 %

```

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.

```

1844     \if@RTL%
1845     \IfStrEq{\xabslineref{edtext:start:\csuse{thisedtext@the\
@edtext@level}}}%
1846     {000}%
1847     }%
1848     {%
1849     \ifnumequal%
1850     {\xabslineref{edtext:start:\csuse{thisedtext@the\
@edtext@level}}}%
1851     {\xabslineref{edtext:end:\csuse{thisedtext@the\@edtext@level
}}}%
1852     }%
1853     {\write@lemma}%
1854     }%
1855     \flag@start@RTL%
1856     \IfStrEq{\xabslineref{edtext:start:\csuse{thisedtext@the\
@edtext@level}}}%
1857     {000}%
1858     }%
1859     {%
1860     \ifnumequal%
1861     {\xabslineref{edtext:start:\csuse{thisedtext@the\
@edtext@level}}}%
1862     {\xabslineref{edtext:end:\csuse{thisedtext@the\@edtext@level
}}}%
1863     {\write@lemma}%
1864     }%
1865     }%
1866     \else%
1867     \flag@start%
1868     \write@lemma%
1869     \fi%
1870 %

```

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

```

1871     \endgroup%
1872     \ifdef{\hypertarget}%
1873     {%
1874     \Hy@raisedlink@left{\hypertarget{\csuse{thisedtext@the\
@edtext@level}:start}}}%

```



```

1875     \showlemma{#1}%
1876     \Hy@raisedlink{\hypertarget{\csuse{thisedtext@the\@edtext@level}:
end}{}}}%
1877     }%
1878     {%
1879     \showlemma{#1}%
1880     }%
1881 %

```

If using `hyperref` package, we put the final `hypertarget` for `\applabel`.

```

1882     \ifdef{\hypertarget}%
1883     {%
1884     \def\do##1{\Hy@raisedlink{\hypertarget{##1:end}{}}}%
1885     \ifcsdef{applabel@thisedtext@the\@edtext@level}{%
1886     \dolistcsloop{applabel@thisedtext@the\@edtext@level}%
1887     \global\cslet{applabel@thisedtext@the\@edtext@level}{\
empty}%
1888     }{}%
1889     }%
1890     {}%
1891 %

```

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.

```

1892     \ifx\end@lemmas\empty \else%
1893     \glp\end@lemmas\to\x@lemma%
1894     \x@lemma%
1895     \global\let\x@lemma=\relax%
1896     \fi%
1897     \ifRTL%
1898     \flag@end@RTL%
1899     \else%
1900     \flag@end%
1901     \fi%
1902 %

```

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`.

```

1903     \global\advance\@edtext@level by -1%
1904     \global\@lemmacommand@false%
1905 %

```

We also reset `\@beforeinsertofthisedtext`

```

1906     \global\let\@beforeinsertofthisedtext\relax%
1907 %

```

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

```

1908 \else%
1909 \showlemma{#1} (\textbf{\textsc{Edtext outside numbered paragraph}})\
led@err@edtextoutsidestart%
1910 \fi%
1911 }%
1912
1913
1914 %

```

`\@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.

```

1915 \newcommand{\@check@edtext@args}[2]{%
1916 \begingroup%
1917 \noexpandarg%
1918 \IfSubStr{#1}{\pstart}{\led@err@PstartInEdtext{first}}{}%
1919 \IfSubStr{#1}{\pend}{\led@err@PendInEdtext{first}}{}%
1920 \IfSubStr{#2}{\pstart}{\led@err@PstartInEdtext{second}}{}%
1921 \IfSubStr{#2}{\pend}{\led@err@PendInEdtext{second}}{}%
1922 \endgroup%
1923 }%
1924 %
1925 %

```

`\@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`.

```

1926 \let\@beforeinsertofthisedtext\relax
1927 %

```

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

```

1928 \newif\ifnumberline
1929 \numberlinetrue
1930 %

```

`\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.

```

1931 \newcommand*{\set@line}{%
1932 \ifledRcol
1933 \ifx\line@listR\empty
1934 \global\noteschanged@true

```

```

1935 \xdef\l@d@nums{000|000|000|000|000|000|000|\edfont@info}%
1936 \else
1937 \glp\line@listR\to\@tempb
1938 \xdef\l@d@nums{\@tempb|\edfont@info}%
1939 \global\let\@tempb=\undefined
1940 \fi
1941 \ifx\annot@listR\empty%
1942 \xdef\l@current@annotR{||}%
1943 \else%
1944 \glp\annot@listR\to\@tempb%
1945 \xdef\l@current@annotR{\@tempb}%
1946 \global\let\@tempb=\undefined%
1947 \fi%
1948 \else
1949 \ifx\line@list\empty
1950 \global\noteschanged>true
1951 \xdef\l@d@nums{000|000|000|000|000|000|000|\edfont@info}%
1952 \else
1953 \glp\line@list\to\@tempb
1954 \xdef\l@d@nums{\@tempb|\edfont@info}%
1955 \global\let\@tempb=\undefined
1956 \fi
1957 \ifx\annot@list\empty%
1958 \xdef\l@current@annot{||}%
1959 \else%
1960 \glp\annot@list\to\@tempb%
1961 \xdef\l@current@annot{\@tempb}%
1962 \global\let\@tempb=\undefined%
1963 \fi%
1964 \fi}
1965
1966 %

```

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

```

1967 \newcommand*\edfont@info{\f@encoding/\f@family/\f@series/\f@shape}
1968
1969 %

```

## 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).

```

1970 \newcommand*\lemma}[1]{%
1971 \global\@lemmacommand>true%
1972 \global\renewcommand{\@tag}{%
1973 \no@expands #1%
1974 }%
1975 \ignorespaces%
1976 }%
1977 %

```

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

`\write@lemma`

```

1978 \newcommand{\@lemma}{%
1979   \booltrue{lemmacommand@the\@edtext@level}%
1980 }%
1981 \newcommand{\write@lemma}{%
1982   \if@lemmacommand@%
1983     \ifledRcol%
1984       \write\linenum@outR{\string\@lemma}%
1985     \else%
1986       \write\linenum@out{\string\@lemma}%
1987     \fi%
1988   \fi%
1989 }%
1990 %

```

`\if@lemmacommand@` This boolean is set to TRUE inside a `\edtext` (or `\critext`) when a `\lemma` command is called. That is useful for some commands which can have a different behavior if the lemma in the note is different from the lemma in the main text.

```

1991 \newif\if@lemmacommand@%
1992 %

```

### VI.3 Substitute line numbers

`\linenum` The `\linenum` macro can change any or all of the page and line numbers that are passed on to the notes.

As argument `\linenum` takes a set of seven parameters separated by vertical bars, in the format used internally for `\l@d@nums` (see [V.9 p. 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. However, you can omit any parameters you do not want to change, and you can omit a string of vertical bars at the end of the argument. Hence `\linenum{18|4|0|18|7|1|0}` is an invocation that changes all the parameters, but `\linenum{|3}` only changes the starting line number, and leaves the rest unaltered.

We use `\\` as an internal separator for the macro parameters.

```

1993 \newcommand*{\linenum}[1]{%
1994   \xdef\@tempa{#1|||||}\noexpand\\l@d@nums}%
1995   \global\let\l@d@nums=\empty
1996   \expandafter\line@set\@tempa|\\\ignorespaces}
1997 %

```

`\line@set` `\linenum` calls `\line@set` to do the actual work; it looks at the first number in the argument to `\linenum`, sets the corresponding value in `\l@d@nums`, and then calls itself to process the next number in the `\linenum` argument, if there are more numbers in `\l@d@nums` to process.

```

1998 \def\line@set#1|#2\\#3|#4\\{%
1999   \gdef\@tempb{#1}%
2000   \ifx\@tempb\empty
2001     \l@d@add{#3}%
2002   \else

```

```

2003     \l@d@add{#1}%
2004     \fi
2005     \gdef\@tempb{#4}%
2006     \ifx\@tempb\empty\else
2007         \l@d@add{|\line@set#2\|#4\}%
2008     \fi}
2009 %

```

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

```

2010 \newcommand{\l@d@add}[1]{\xdef\l@d@nums{\l@d@nums#1}}
2011
2012 %

```

`\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.

```

2013 \newcommand*\lineannot}[1]{%
2014     \lineannot@set#1|}%
2015 }%
2016 \def\lineannot@set#1|#2|{%
2017     \expandafter\parse@annot#1|#2|%
2018     \IfStrEq{#1}{}%
2019         {\let\@tempa\annot@start}%
2020         {\def\@tempa{#1}}%
2021     \IfStrEq{#2}{}%
2022         {\let\@tempb\annot@start}%
2023         {\def\@tempb{#2}}%
2024     \xdef\l@current@annot{\@tempa|\@tempb}%
2025 }%
2026 %

```

## 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@txt 27 \newcommand{\get@sw@txt}[1]{%
2028   \begingroup%
2029   \swnoexpands%
2030   %.
```

Using case sensibility option.

```
2031   \ifsw@caseinsensitive%
2032     \def\@tmpa##1{\lowercase{##1}}%
2033   \else%
2034     \def\@tmpa##1{##1}%
2035   \fi%
2036   %
```

And now, define `\sw@txt`.

```
2037     \@tmpa{\protected@xdef\sw@txt{##1}}%
2038   \endgroup%
2039 }%
2040 %
```

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}{%
2042   \let\sameword\l@secondmandarg%Allow to have nested \sameword
2043   \let\emph\@firstofone%
2044   \let\textit\@firstofone%
2045   \let\textbf\@firstofone%
2046   \let\textsc\@firstofone%
2047   \let\framebox\@firstofone%
2048   \let\edtext\dummy@edtext%
2049   \RenewExpandableDocumentCommand{\edindex}{om}{}%
2050   \ifdefined\index%
2051     \RenewExpandableDocumentCommand{\index}{om}{}%
2052   \fi%
2053   \let\selectlanguage\@gobble%
2054   \let\foreignlanguage\@secondoftwo%
2055   \ifdefined\xpg@loaded%
2056     \renewcommand\do[1]{%
2057       \expandafter\RenewExpandableDocumentCommand\expandafter{\csname
text##1\endcsname}{om}{###2}%
2058     }%
2059     \expandafter\docsvlist\expandafter{\xpg@loaded}%
2060   \fi%
2061 }%
2062 %

```

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

```

2063 \newcommandx{\sameword}[2][1,usedefault]{%
2064   \ifxetex%
2065     \if@RTL%
2066       \led@err@samewordRTL%
2067     \fi%
2068   \fi%
2069   \leavevmode%
2070   \get@sw@txt{#2}%
2071 %

```

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

```

2072   \unless\ifledRcol%
2073     \csnumgdef{sw@\sw@txt}{\csuse{sw@\sw@txt}+\@ne}%
2074 %

```

Then, write its value to the numbered file.

```

2075   \protected@write\linenum@out{\string\@sw{\sw@txt}{\csuse{sw@\sw@txt}
}}{#1}}%
2076 %

```

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

```

2077   \else%
2078     \csnumgdef{sw@\sw@txt}{\csuse{sw@\sw@txt}+\@ne}%

```



```

2079 \protected@write\linenum@outR{}{\string\@sw{\sw@txt}{\csuse{sw@\sw@txt
}}{#1}}%
2080 \fi%
2081 %

```

And print the word.

```

2082 #2%
2083 }%
2084 %

```

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

```

\if@addsw5 \newif\if@addsw%
2086 %

```

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

```

2087 \newcommand{\@sw}[3]{%
2088 \get@sw@txt{#1}%
2089 \unless\ifledRcol%
2090 %

```

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

```

2091 \csxdef{sw@\sw@txt @\the\absline@num @\the\section@num}{#2}%
2092 %

```

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

```

2093 \numdef{\prev@line}{\the\absline@num-1}%
2094 \ifcsundef{sw@\sw@txt @\prev@line @\the\section@num}{%
2095 \csnumgdef{sw@\sw@txt @\prev@line @\the\section@num}{#2-1}%
2096 }{}%
2097 %

```

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

```

2098 \numdef{\the@sw}{#2-\csuse{sw@\sw@txt @\prev@line @\the\section@num}}%
2099 %

```

And do the same thing for the right side.

```

2100 \else%
2101 \csxdef{sw@\sw@txt @\the\absline@numR @\the\section@numR @R}{#2}%
2102 \numdef{\prev@line}{\the\absline@numR-1}%
2103 \ifcsundef{sw@\sw@txt @\prev@line @\the\section@numR @R}{%
2104 \csnumgdef{sw@\sw@txt @\prev@line @\the\section@numR @R}{#2-1}%
2105 }{}%
2106 \numdef{\the@sw}{#2-\csuse{sw@\sw@txt @\prev@line @\the\section@numR @R
}}%
2107 \fi%
2108 %

```

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

```

2109 \@tempcnta=\@edtext@level
2110 \@whilenum{\@tempcnta>0}\do{%
2111   \ifcsdef{sw@list@edtext@tmp@the\@tempcnta}%
2112     {%
2113       \@addswfalse%
2114       \notbool{lemmacommand@the\@tempcnta}%
2115       {\@addswtrue}%
2116       {\IfStrEq{#3}{inlemma}%
2117        {\@addswtrue}%
2118        {%
2119          \def\do##1{%
2120            \ifnumequal{##1}{the\@tempcnta}%
2121              {\@addswtrue\listbreak}%
2122              {%
2123                }%
2124                \docsvlist{#3}%
2125                }%
2126              }%
2127              \if@addsw%
2128                \letcs{\@tmp}{sw@list@edtext@tmp@the\@tempcnta}%
2129                \ifledRcol%
2130                  \xright@appenditem{\the@sw}{\the\absline@numR}}\to\@tmp%
2131                \else%
2132                  \xright@appenditem{\the@sw}{\the\absline@num}}\to\@tmp%
2133                \fi%
2134                \cslet{sw@list@edtext@tmp@the\@tempcnta}{\@tmp}%
2135                \fi%
2136              }%
2137              {}%
2138              \advance\@tempcnta by -1%
2139            }%
2140          }%
2141        %

```

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

```

2142 \newcommandx{\sameword@inedtext}[2][1,usedefault]{%
2143   \get@sw@txt{#2}%
2144   \unless\ifledRcol%
2145   %

```

Just a precaution.

```

2146   \ifx\sw@list@inedtext\empty%
2147     \def\the@sw{999}%
2148     \def\this@absline{-99}%
2149   \else%
2150   %

```

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

```

2151   \gl@p\sw@list@inedtext\to\@tmp%
2152   \edef\the@sw{\expandafter\@firstoftwo\@tmp}%
2153   \edef\this@absline{\expandafter\@secondoftwo\@tmp}%
2154   \fi%

```

```

2155 %
First, calculate the number of occurrences of the word in the current line
2156 \ifcsdef{sw@sw@txt @\this@absline @\the\section@num}{%
2157 \numdef{\prev@line}{\this@absline-1}%
2158 \numdef{\sw@atthisline}{\csuse{sw@sw@txt @\this@absline @\the\
section@num}-\csuse{sw@sw@txt @\prev@line @\the\section@num}}%
2159 }%
2160 {\numdef{\sw@atthisline}{0}}%
2161 %
Finally, print the rank, but only if there is more than one occurrence of the word in the
current line.
2162 \ifnumgreater{\sw@atthisline}{1}%
2163 {\showwordrank{#2}{\the@sw}}%
2164 {#2}%
2165 %
And the same for right side.
2166 \else%
2167 \ifx\sw@list@inedtext\empty%
2168 \def\the@sw{999}%
2169 \def\this@absline{-99}%
2170 \else%
2171 \glp\sw@list@inedtext\to\@tmp%
2172 \edef\the@sw{\expandafter\@firstoftwo\@tmp}%
2173 \edef\this@absline{\expandafter\@secondoftwo\@tmp}%
2174 \fi%
2175 \ifcsdef{sw@sw@txt @\this@absline @\the\section@numR @R}{%
2176 \numdef{\prev@line}{\this@absline-1}%
2177 \numdef{\sw@atthisline}{\csuse{sw@sw@txt @\this@absline @\the\
section@numR @R}-\csuse{sw@sw@txt @\prev@line @\the\section@numR @R}}%
2178 }%
2179 {\numdef{\sw@atthisline}{0}}%
2180 \ifnumgreater{\sw@atthisline}{1}%
2181 {\showwordrank{#2}{\the@sw}}%
2182 {#2}%
2183 \fi%
2184 }%
2185 %

```

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

```

2186 \newcommand{\showwordrank}[2]{%
2187 #1\textsuperscript{#2}%
2188 }%
2189 %

```

## 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.

```
2190 \newbox\raw@text
2191 \newif\ifnumberedpar@
2192 \newcount\num@lines
2193 \newbox\one@line
2194 \newcount\par@line
2195 %
```

`\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`.

```
2196 \newcount\pstarts@typeset@L%
2197 \newcount\pstarts@read@L%
2198 %
```

`\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
2199 \if@every@pstart@star@
2200 \newcommand{\AtStartEveryPstart}[1]{%
2201   \ifstrempy{#1}%
2202   {\gdef\@at@start@every@pstart{}}%
2203   {\gdef\@at@start@every@pstart{#1}}%
2204 }%
2205 \def\@at@start@every@pstart{}}%
```

```

2206 \newif\ifat@every@pstart@star@%
2207 \newcommand{\AtEveryPstart}[1]{%
2208   \ifstrempy{#1}%
2209   {\gdef\at@every@pstart{}}%
2210   {\gdef\at@every@pstart{\noindent#1}}%
2211 \global\at@every@pstart@star@false%
2212 }%
2213
2214 \WithSuffix\newcommand\AtEveryPstart*[1]{%
2215   \ifstrempy{#1}%
2216   {\gdef\at@every@pstart{}}%
2217   {\gdef\at@every@pstart{#1}}%
2218   \global\at@every@pstart@star@true%
2219 }%
2220 \def\at@every@pstart{}%
2221
2222 \newcounter{pstart}
2223 \renewcommand{\thepstart}{\bfseries\@arabic\c@pstart}. }
2224 \newif\ifnumberpstart
2225 \numberpstartfalse
2226 \newif\iflabelpstart
2227 \labelpstartfalse
2228 \newcommandx*\{pstart}[2][1,2,usedefault]{%
2229   \normal@pars%
2230   \ifboolexpr{%
2231     test {\ifstrempy{#1}}%
2232     and test {\ifstrempy{#2}}%
2233   }%
2234   {\at@every@pstart}%
2235   {%
2236     \ifstrempy{#1}{\noindent#1}%
2237     \ifstrempy{#2}{#2}%
2238   }%
2239   \ifautopar%
2240     \autopar%
2241   \fi%
2242   \ifluatex%
2243     \edef\l@luatextextdir@L{\the\textdir}%
2244   \fi%
2245   \@nobreaktrue%
2246   \ifnumbering \else%
2247     \led@err@PstartNotNumbered%
2248     \beginnumbering%
2249   \fi%
2250   \ifnumberedpar@%
2251     \led@err@PstartInPstart%
2252     \pend%
2253   \fi%
2254   \list@clear{\inserts@list}%
2255   \global\let\next@insert=\empty%
2256   \begingroup\normal@pars%
2257   \global\advance \l@dnumpstartsL\@ne
2258   \global\advance \pstarts@typeset@L\@ne%
2259   \global\advance \pstarts@read@L\@ne%

```

```

2260 \global\setbox\raw@text=\vbox\bgroup%
2261 \if@nobreak%
2262 \if@afterindent\else%
2263 \unless\ifinstanza%
2264 \noindent%
2265 \fi%
2266 \global\@afterindenttrue%
2267 \fi%
2268 \fi%
2269 \ifboolexpr{%
2270 bool{autopar}%
2271 and bool{by@autopar}%
2272 }%
2273 {}%
2274 {%
2275 \ifnumberpstart%
2276 \ifinstanza\else%
2277 \ifsidepstartnum\else%
2278 \thepstart%
2279 \fi%
2280 \fi%
2281 \fi%
2282 }%
2283 \numberedpar@true%
2284 \iflabelpstart%
2285 \protected@edef\@currentlabel{\p@pstart\thepstart}%
2286 \fi%
2287 \l@dzeropenalties%
2288 \@at@start@every@pstart%
2289 \global\by@autoparfalse%
2290 \ignorespaces%because not automatically ignored if an optional argument
is used (classical TeX behavior for space after commands)
2291 }
2292 %

```

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

```

2293 \newcommand*{\pend}[2][1,2,usedefault]{\ifnumbering \else%
2294 \led@err@PendNotNumbered%
2295 \fi%
2296 \global\l@dskipversenumberfalse%
2297 \ifnumberedpar@ \else%
2298 \led@err@PendNoPstart%
2299 \fi%
2300 %

```

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.

```

2301 \l@dzeropenalties%
2302 \@at@end@every@pend%

```

```

2303 \endgraf\global\num@lines=\prevgraf\egroup%
2304 \global\par@line=0%
2305 \get@firsthalf@antilabe@wd%
2306 %

```

Output the lines.

```

2307 \loop\ifvbox\raw@text%
2308 \do@line%
2309 \repeat%
2310 %

```

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

```

2311 \flush@notes%
2312 \endgroup%
2313 \ignorespaces%
2314 %

```

Increase `pstart` counter.

```

2315 \ifnumberpstart%
2316 \global\pstartnumtrue%
2317 \fi%
2318 \addtocounter{pstart}{1}%
2319 \ifcontinuousnumberingwithcolumns%
2320 \addtocounter{pstartL}{1}%
2321 \addtocounter{pstartR}{1}%
2322 \fi%
2323 %

```

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

```

2324 \normal@pars%
2325 \ifboolexpr{%
2326 test {\ifstrempy{#1}}%
2327 and test {\ifstrempy{#2}}%
2328 }%
2329 {\at@every@pend}%
2330 {%
2331 \ifstrempy{#1}{\noindent#1}%
2332 \ifstrempy{#2}{#2}%
2333 }%
2334 %

```

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.

```

2335 \@nobreakfalse%
2336 \ifautopar%
2337 \autopar%
2338 \fi%
2339 %

```

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.

```

2340 \ifbypstart@%
2341 \begingroup%
2342 \unless\ifinstanza%
2343 \let\leavevmode\relax%
2344 \setlinenum{0}%
2345 \fi%
2346 \endgroup%
2347 \resetprevline@%
2348 \fi%
2349 }
2350 %

```

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.

```

\AtEveryPend51
\at@every@pend52
\ifat@every@pend@star@53 \newif\ifat@every@pend@star@%
2354 \newcommand{\AtEveryPend}[1]{%
2355 \ifstrempy{#1}%
2356 {\gdef\at@every@pend{}}%
2357 {\gdef\at@every@pend{\noindent#1}}%
2358 \global\at@every@pend@star@false%
2359 }%
2360 \WithSuffix\newcommand\AtEveryPend*[1]{%
2361 \ifstrempy{#1}%
2362 {\gdef\at@every@pend{}}%
2363 {\gdef\at@every@pend{#1}}%
2364 \global\at@every@pend@star@true%
2365 }%
2366 \xdef\at@every@pend{}%
2367
2368 %

```

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

```

2369 \newcommand{\AtEndEveryPend}[1]{%
2370 \ifstrempy{#1}%
2371 {\xdef\@at@end@every@pend{}}%
2372 {\gdef\@at@end@every@pend{#1}}%
2373 }%
2374 \def\@at@end@every@pend{}%
2375 %

```

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

```

2376 \newcommand*\l@dzero penalties{%
2377 \brokenpenalty \z@ \clubpenalty \z@
2378 \displaywidowpenalty \z@ \interlinepenalty \z@ \predisplaypenalty \z@
2379 \postdisplaypenalty \z@ \widowpenalty \z@}
2380
2381 %

```



`\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.

```

2382 \newif\ifautopar
2383 \newif\ifby@autopar%
2384 \newcommand*{\autopar}{
2385   \ifledRcol
2386     \ifnumberingR \else
2387       \led@err@AutoparNotNumbered
2388       \beginnumberingR
2389       \fi
2390     \else
2391       \ifnumbering \else
2392         \led@err@AutoparNotNumbered
2393         \beginnumbering
2394         \fi
2395       \fi
2396       \autopartrue
2397       \everypar{\setbox0=\lastbox
2398         \endgraf \vskip-\parskip
2399         \global\by@autopartrue%
2400         \pstart \noindent \kern\wd0%
2401         \ifnumberpstart%
2402           \ifinstanza\else%
2403             \thepstart%
2404             \fi%
2405           \fi%
2406           \let\par=\pend}%

```

```
2407 \ignorespaces}
2408 %
```

`\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.

```
2409 \newcommand*{\normal@pars}{\everypar{} \let\par\endgraf}
2410
2411 %
```

`\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.

```
2412 \newif\ifautopar@pause
2413 %
```

## 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.

```
2414 \newcommand*{\l@dunhbox@line}[1]{\unhbox #1}
2415 \newcommand*{\do@line}{%
2416   {\vbadness=10000
2417     \splittopskip=\z@
2418     \do@linehook
2419   \l@emptyd@ta
2420     \global\setbox\one@line=\vsplit\raw@text to\baselineskip}%
2421   \unvbox\one@line \global\setbox\one@line=\lastbox
2422   \getline@num
2423   \IfStrEq{\led@pb@setting}{before}{\led@check@pb\led@check@nopb}{ }
2424   \ifnum\@lock>\@ne
2425     \inserthangingsymboltrue
2426   \else
2427     \inserthangingsymbolfalse
2428   \fi
2429   \check@pb@in@verse
2430   \ifl@dhidenumber%
2431     \global\l@dhidenumberfalse%
2432     \f@x@l@cks%
2433   \else%
2434     \affixline@num%
2435   \fi%
2436 %
```

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

```

2437 \xifinlist{\the\pstarts@typeset@L}{\eled@sections@}%
2438   {\print@eledsection}%
2439   {\print@line}%
2440 \IfStrEq{\led@pb@setting}{after}{\led@check@pb\led@check@nopb}{-}
2441 }%
2442 %

```

### VII.2.2 Process for “normal” line

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

```

2443 \def\print@line{
2444 %

```

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

```

2445   \affixpstart@num%
2446 %

```

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

```

2447   \hb@xt@ \linewidth{%
2448 %

```

User hooks.

```

2449   \do@insidelinehook%
2450   \csuse{insidethis@the\absline@num @the\section@num}%
2451   \global\csundef{insidethis@the\absline@num @the\section@num}%
2452 %

```

Left line number

```

2453   \l@dld@ta%
2454 %

```

Prepare text to be inserted before notes.

```

2455   \if@firstlineofpage%
2456   \set@Xtxtbeforenotes%
2457   \set@txtbeforenotesX%
2458   \global\@firstlineofpagefalse%
2459   \fi%
2460 %

```

Insert footnotes made of manuscripts data and critical footnotes.

```

2461   \ifdefstring{\ms@data@position}{msdata-regular}{%
2462   \insert@msdata%
2463   \add@inserts%
2464   \add@Xgroupbyline%
2465   }{%
2466   \add@inserts%
2467   \add@Xgroupbyline%
2468   \insert@msdata%
2469   }%
2470 %

```

Insert marginal notes.

```

2471     \affixside@note%
2472 %
Print left notes.
2473     \l@dlsn@te
2474 %
Boxes the line, writes information about new line in the numbered file.
2475     {\ledllfill\hb@xt@ \wd\one@line{%
2476       \new@line%
2477 %
If we use the continuousnumberingwithcolumns, we increase right line number.
2478     \continuousnumberingwithcolumns@sync@linenumber@singletext%
2479 %
If we use Lua $\TeX$  then restore the direction.
2480     \ifluatex%
2481       \textdir\l@luatextextdir@L%
2482     \fi%
2483 %
Insert, if needed, the hanging symbol.
2484     \inserthangingsymbol%
2485 %
And so, print the line.
2486     \l@dunhbox@line{\one@line}}%
2487 %
Right line number
2488     \ledrlfill\l@drd@ta%
2489 %
Print right notes.
2490     \l@drsn@te%
2491     }}%
2492 %
And reinsert penalties (for page breaking)...
2493     \add@penalties%
2494     }
2495 %

```

### 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.

```

2496 \def\print@eledsection{%
2497   \disable@edindex%
2498   \disable@edlabel%
2499   \if@firstlineofpage%
2500     \set@Xtxtbeforenotes%
2501     \set@txtbeforenotesX%
2502     \global\@firstlineofpagefalse%
2503   \fi%
2504   \ifdefstring{\ms@data@position}{msdata-regular}{%
2505     \insert@msdata%
2506     \add@inserts%
2507     \add@Xgroupbyline%
2508   }{%
2509     \add@inserts%
2510     \add@Xgroupbyline%
2511     \insert@msdata%
2512   }%
2513   \affixside@note%
2514   \numdef{\temp@}{\pstarts@typeset@L-1}%
2515   \xifinlist{\temp@}{\eled@sections@@}{\@nobreaktrue}{\@nobreakfalse}%
2516   \@eled@sectioningtrue%
2517   \csuse{eled@sectioning@the\pstarts@typeset@L}%
2518   \@eled@sectioningfalse%
2519   \global\csundef{eled@sectioning@the\pstarts@typeset@L}%
2520   \if@RTL%
2521     \hspace{-3\paperwidth}%
2522     {\hbox{\l@dunhbox@line{\one@line}} \new@line}%
2523   \else%
2524     \hspace{3\paperwidth}%
2525     {\new@line \hbox{\l@dunhbox@line{\one@line}}}%
2526   \fi%
2527   \vskip-\baselineskip%
2528   \continuousnumberingwithcolumns@sync@linenumber@singletext%
2529   \restore@edindex%
2530   \restore@edlabel%
2531 }
2532 %

```

#### 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.

```

2533 \newcommand*{\do@linehook}{}
2534 \newcommand*{\do@insidelinehook}{}
2535 %

```

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

```

2536 \newcommand*{\dolinehook}[1]{\gdef\do@linehook{#1}}%
2537 \newcommand*{\doinsidelinehook}[1]{\gdef\do@insidelinehook{#1}}%
2538

```

2539 %

`\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.

```
2540 \newcommand{\doinsidethislinehook}[1]{%
2541   \leavevmode%In case it begins with a \pstart, ensure the \@insidethisline
is written after \@nl
2542   \ifledRcol%
2543     \write\linenum@outR{\string\@insidethisline[\unexpanded{#1}]}%
2544   \else%
2545     \write\linenum@out{\string\@insidethisline[\unexpanded{#1}]}%
2546   \fi%
2547 }%
2548 %
```

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

```
2549 \newcommand{\@insidethisline}[1]{%
2550   \ifledRcol%
2551     \csgappto{insidethisR@the\absline@numR @\the\section@numR}{#1}%
2552   \else%
2553     \csgappto{insidethis@the\absline@num @\the\section@num}{#1}%
2554   \fi%
2555 }%
2556 %
```

### 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
2557 \newcommand*{\l@emptyd@ta}{%
2558   \gdef\l@dld@ta{}%
2559   \gdef\l@drd@ta{}%
2560   \gdef\l@dcsnotetext@l{}%
2561   \gdef\l@dcsnotetext@r{}%
2562   \gdef\l@dcsnotetext{}%
2563
2564 %
```

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

```
2565 \newcommand{\l@dlsn@te}{%
2566   \ifboolexpr{%
2567     bool {l@dprintingcolumns}%
2568     and bool {ledRcol@}%
2569   }{% If we are on a right column
2570     \@tempdima=\@morespace@leftnote@rightcolumn%
2571   }{%
2572     \@tempdima=\z@%
```

```

2573 }%
2574 \hb@xt@ \z@{\hss\box\l@dlp@rbox\kern\ledlsnotesep\hskip\@tempdima}%
2575 }%
2576 \newcommand{\l@drsn@te}{%
2577 \ifboolexpr{%
2578     bool {l@dprintingcolumns}%
2579     and not bool {ledRcol@}%
2580 }{% If we are on a left column
2581     \@tempdima=\@morespace@rightnote@leftcolumn%
2582 }{%
2583     \@tempdima=\z@%
2584 }%
2585 \hb@xt@ \z@{\hskip\@tempdima\kern\ledrsnotesep\box\l@drp@rbox\hss}%
2586 }%
2587
2588 %

```

`\ledllfill` These macros are called at the left (`\ledllfill`) and the right (`\ledllfill`) of each numbered line.

```

2589 \newcommand*\ledllfill{\unless\if@RTL\hfil\fi}%
2590 \newcommand*\ledrlfill{\if@RTL\hfil\fi}%
2591
2592 %

```

## 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.

```

2593 \newcommand*\getline@num{%
2594     \global\advance\absline@num \@ne%
2595     \do@actions
2596     \do@ballast
2597     \ifnumberline
2598         \ifsublines@
2599             \ifnum\sub@lock<\tw@
2600                 \global\advance\subline@num \@ne
2601             \fi
2602         \else
2603             \ifnum\@lock<\tw@
2604                 \global\advance\line@num \@ne
2605                 \global\subline@num \z@
2606             \fi
2607         \fi
2608     \fi
2609 }
2610 %

```

`\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`

```
2611 \newcount\ballast@count
2612 \newcounter{ballast}
2613 \setcounter{ballast}{0}
2614 %
```

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.

```
2615 \newcommand*{\do@ballast}{\global\ballast@count \z@
2616 \begingroup
2617 \advance\absline@num \@ne
2618 \ifnum\next@actionline=\absline@num
2619 \ifnum\next@action>-1001\relax
2620 \global\advance\ballast@count by -\c@ballast
2621 \fi
2622 \fi
2623 \endgroup}
2624 %
```

`\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`.

```
2625 \newcommand*{\do@actions}{%
2626 \global\let\do@actions@next=\relax
2627 \ifnum\absline@num<\next@actionline\else
2628 %
```

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.

```
2629 \ifnum\next@action>-1001
2630 \global\page@num=\next@action
2631 \ifresumenummering@start%
2632 \setbox0=\hbox{}%Required to get the correct page number, when the
\resumenummering is just after a \newpage
2633 \ifnum\pausenummering@page@num<\page@num%
2634 \global\resumenummering@startfalse%
2635 \fi%
2636 \fi%
2637 \ifboolexpr{%
2638 bool{resumenummering@start}%
2639 and test {\ifdimgreater{\pagedepth}{\z@}}%
2640 }%
2641 {}%
```



```

2642     {\global\@firstlineofpagetrue}%
2643     \ifcsdef{reset@line\the\absline@num @\the\section@num}%
2644     {%
2645         \global\line@num=\z@%
2646         \global\subline@num=\z@%
2647         \resetprevline@%
2648     }%
2649     {}%
2650     \global\resumenumering@startfalse%
2651     \add@msdata@firstlineofpage%
2652 %

```

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`.)

```

2653     \else
2654         \ifnum\next@action<-4999
2655             \@l@dttempcnta=-\next@action
2656             \advance\@l@dttempcnta by -5001
2657             \ifsublines@
2658                 \global\subline@num=\@l@dttempcnta
2659             \else
2660                 \global\line@num=\@l@dttempcnta
2661             \fi
2662 %

```

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

```

2663         \else
2664             \@l@dttempcnta=-\next@action
2665             \advance\@l@dttempcnta by -1000
2666             \do@actions@fixedcode
2667         \fi
2668     \fi
2669 %

```

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.

```

2670     \ifx\actionlines@list\empty
2671         \gdef\next@actionline{1000000}%
2672     \else
2673         \gl@p\actionlines@list\to\next@actionline
2674         \gl@p\actions@list\to\next@action
2675         \global\let\do@actions@next=\do@actions
2676     \fi
2677 \fi
2678 %

```

Make the recursive call, if necessary.

```

2679 \do@actions@next}
2680
2681 %

```

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

```

2682 \newcommand*{\do@actions@fixedcode}{%
2683   \ifcase\l@dttempcnta
2684   \or%                % 1001 = starting sublineation
2685     \global\sublines@true
2686   \or%                % 1002 = ending sublineation
2687     \global\sublines@false
2688   \or%                % 1003 = starting locking number
2689     \global\@lock=\@ne
2690   \or%                % 1004 = ending locking number
2691     \ifnum\@lock=\tw@
2692       \global\@lock=\thr@@
2693     \else
2694       \global\@lock=\z@
2695     \fi
2696   \or%                % 1005 = starting locking subnumber
2697     \global\sub@lock=\@ne
2698   \or%                % 1006 = ending locking subnumber
2699     \ifnum\sub@lock=\tw@
2700       \global\sub@lock=\thr@@
2701     \else
2702       \global\sub@lock=\z@
2703     \fi
2704   \or%                % 1007 = skipping numbering
2705     \l@dskipnumbertrue
2706   \or%                % 1008 = skipping numbering in stanza
2707     \l@dskipversenumbertrue%
2708   \or%                % 1009 = hiding number
2709     \l@dhidenumberttrue
2710   \or%                % 1010 = inserting msdata
2711     \add@msdata%
2712   \else
2713     \led@warn@BadAction
2714   \fi}
2715
2716
2717 %

```

### 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.

```

2718 \newcommand{\continuousnumberingwithcolumns@sync@linenumber@singletext}{%
2719   \ifcontinuousnumberingwithcolumns%
2720     \unless\ifafterendnumberingR%
2721       \newlineR%
2722       \xappto\next@line@list@stuffR{%
2723         \unexpanded{\global\line@numR=}\the\line@num%
2724       }%
2725     \fi%
2726   \fi%
2727 }%
2728 %

```

## 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`

```

2729 \newcommand{\linenumannotation}[1]{%
2730   \leavevmode%In case it begins with a \pstart, ensure the \@annot is
written after \@nl
2731   \ifledRcol%
2732     \write\linenum@outR{\string\@annot[#1]}%
2733   \else%
2734     \write\linenum@out{\string\@annot[#1]}%
2735   \fi%
2736 }%
2737
2738 \newcommand{\resetlinenumannotation}[0]{%
2739   \leavevmode%In case it begins with a \pstart, ensure the \@resetannot is
written after \@nl
2740   \ifledRcol%
2741     \write\linenum@outR{\string\@resetannot}%
2742   \else%
2743     \write\linenum@out{\string\@resetannot}%
2744   \fi%
2745 }%
2746
2747 %

```

`\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.

```
2748 \def\Xlinenumannotationposition@side{after}%
2749 %
```

`\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`.

```
2750 \def\Xwraplinenumannotation@ref{\textsuperscript}%
2751 \def\Xwraplinenumannotation@side{\textsuperscript}%
2752 \newtoggle{Xnoidenticallinenumannotation@ref}%
2753 \newtoggle{Xnoidenticallinenumannotation@side}%
2754 %
```

`\iflinenumannotationotherside` The `\iflinenumannotationotherside` boolean allows to print the annotation on the side opposite the line number.

```
2755 \newif\iflinenumannotationotherside%
2756 %
```

`\@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.

```
2757 \newcommand{\@annot}[1]{%
2758   \store@annot@to@absline{#1}%
2759   \def\current@annot{#1}%
2760 }%
2761 \newcommand{\store@annot@to@absline}[1]{%
2762   \ifledRcol%
2763     \ifcsdef{annotR@the\absline@numR @the\section@numR}{%
2764       \csgappto{annotR@the\absline@numR @the\section@numR}{\
@linenumannotationsep#1}%
2765     }{%
2766       \csgdef{annotR@the\absline@numR @the\section@numR}{#1}%
2767     }%
2768   \else%
2769     \ifcsdef{annot@the\absline@num @the\section@num}{%
2770       \csgappto{annot@the\absline@num @the\section@num}{\
@linenumannotationsep#1}%
2771     }{%
```

```

2772 \csgdef{annot@the\absline@num @the\section@num}{#1}%
2773 }%
2774 \fi%
2775 }%
2776 %

```

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

```

2777 \let\current@annot=\empty%
2778 %

```

`\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.

```

2779 \newcommand{\reset@current@annot}{%
2780 \unless\ifnoresetlinenumannotation@
2781 \let\current@annot\empty%
2782 \fi
2783 }%
2784 \newcommand{\@resetannot}{%
2785 \let\current@annot\empty%
2786 }%
2787 %

```

`\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.

```

2788 \def\parse@annot#1|#2|{%
2789 \gdef\annot@start{#1}%
2790 \gdef\annot@end{#2}%
2791 }%
2792 %

```

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

```

\@linenumannotationsep
2793 \newcommand{\setlinenumannotationsep}[1]{\gdef\@linenumannotationsep{#1}}%
2794 \def\@linenumannotationsep{, }%
2795 %

```

## 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.

```
2796 \newcommand*{\affixline@num}{%
2797 %
```

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).

```
2798 \ifledgroupnotesL\else
2799 \ifnumberline
2800 \ifl@dskipnumber
2801 \global\l@dskipnumberfalse
2802 \else
2803 \ifsublines@
2804 \l@tempcntb=\subline@num
2805 \ifnum\subline@num>\c@firstsublinenum
2806 \l@tempcnta=\subline@num
2807 \advance\l@tempcnta by-\c@firstsublinenum
2808 \divide\l@tempcnta by\c@sublinenumincrement
2809 \multiply\l@tempcnta by\c@sublinenumincrement
2810 \advance\l@tempcnta by\c@firstsublinenum
2811 \else
2812 \l@tempcnta=\c@firstsublinenum
2813 \fi
2814 %
```

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.

```
2815 \ch@cksub@l@ck
2816 %
```

Now the line number case, which works the same way.

```
2817 \else
2818 \l@tempcntb=\line@num
2819 %
```

Check on the  $\backslash\text{linenumberlist}$  If it is  $\backslash\text{empty}$  use the standard algorithm.

```
2820 \ifx\linenumberlist\empty
2821 \ifnum\line@num>\c@firstlinenum
2822 \l@tempcnta=\line@num
2823 \advance\l@tempcnta by-\c@firstlinenum
2824 \divide\l@tempcnta by\c@linenumincrement
2825 \multiply\l@tempcnta by\c@linenumincrement
2826 \advance\l@tempcnta by\c@firstlinenum
2827 \else
```

```

2828         \@l@dtmpcnta=\c@firstlinenum
2829         \fi
2830     \else
2831 %

```

The `\linenumberlist` was not `\empty`, so here is Wayne's numbering mechanism. This takes place in  $\TeX$ 's mouth.

```

2832         \@l@dtmpcnta=\line@num
2833         \edef\rem@inder{\,\linenumberlist,\number\line@num,}%
2834         \edef\sc@n@list{\def\noexpand\sc@n@list
2835             ###1,\number\@l@dtmpcnta,###2|{\def\noexpand\rem@inder
2836 {####2}}}%
2837         \sc@n@list\expandafter\sc@n@list\rem@inder|%
2838         \ifx\rem@inder\empty%
2839             \advance\@l@dtmpcnta\@ne
2840         \fi
2841 %

```

A locking check for lines, just like the version for sub-line numbers above.

```

2842     \ch@ck@l@ck
2843     \fi
2844 %

```

The following tests are true if we need to print a line number.

```

2845     \ifnum\@l@dtmpcnta=\@l@dtmpcntb
2846     \ifl@dskipversenumber\else
2847 %

```

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  $\mathcal{E}\TeX$  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
2848     \if@twocolumn
2849     \if@firstcolumn
2850         \gdef\l@dld@ta{\llap{\leftlinenum}}%
2851         \iflinenumannotationotherside%
2852             \gdef\l@drd@ta{\rlap{\rightlinenumannotation}}%
2853         \fi%
2854     \else
2855         \gdef\l@drd@ta{\rlap{\rightlinenum}}%
2856         \iflinenumannotationotherside%
2857             \gdef\l@dld@ta{\llap{\leftlinenumannotation}}%
2858         \fi%
2859     \fi

```

```

2860         \else
2861             \ifboolexpr{bool {l@dprintingcolumns} and test {\
ifnumgreater{\line@margin@columns}{\m@ne}}}%
2862                 {\@l@tempcntb=\line@margin@columns}%
2863                 {\@l@tempcntb=\line@margin}%
2864             \ifnum\@l@tempcntb>\@ne
2865                 \advance\@l@tempcntb \page@num
2866             \fi
2867             \ifboolexpr{%
2868                 bool {l@dprintingcolumns}%
2869                 and (%
2870                     (test {\ifdefstring{\linenum@OnlyPages@ForColumns}{left
}}}%
2871                     and test {\ifnumodd{\page@num}}%
2872                     )%
2873                 or%
2874                     (test {\ifdefstring{\linenum@OnlyPages@ForColumns}{right
}}}%
2875                     and not test {\ifnumodd{\page@num}}%
2876                     )%
2877                 )%
2878             }%
2879             {}%
2880             {%
2881             \ifodd\@l@tempcntb%
2882                 \gdef\l@drd@ta{\rlap{{\rightlinenum}}}%
2883                 \iflinenumannotationotherside%
2884                     \gdef\l@dld@ta{\llap{{\leftlinenumannotation}}}%
2885                 \fi%
2886             \else%
2887                 \gdef\l@dld@ta{\llap{{\leftlinenum}}}%
2888                 \iflinenumannotationotherside%
2889                     \gdef\l@drd@ta{\rlap{{\rightlinenumannotation}}}%
2890                 \fi%
2891             \fi%
2892             }%
2893             \fi
2894         \fi
2895     \fi
2896 %

```

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

```

2897         \f@x@l@cks
2898         \fi
2899         \fi
2900     \fi
2901 }
2902 %
2903 %

```

`\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.

```

2904 \newcommand*{\ch@cksub@l@ck}{%
2905   \ifcase\sub@lock
2906     \or
2907       \ifnum\sublock@disp=\@ne
2908         \@l@tempcntb=\z@ \@l@tempcnta=\@ne
2909       \fi
2910     \or
2911       \ifnum\sublock@disp=\tw@ \else
2912         \@l@tempcntb=\z@ \@l@tempcnta=\@ne
2913       \fi
2914     \or
2915       \ifnum\sublock@disp=\z@
2916         \@l@tempcntb=\z@ \@l@tempcnta=\@ne
2917       \fi
2918   \fi}
2919 %

```

Similarly for line numbers.

```

2920 \newcommand*{\ch@ck@l@ck}{%
2921   \ifcase\@lock
2922     \or
2923       \ifnum\lock@disp=\@ne
2924         \@l@tempcntb=\z@ \@l@tempcnta=\@ne
2925       \fi
2926     \or
2927       \ifnum\lock@disp=\tw@ \else
2928         \@l@tempcntb=\z@ \@l@tempcnta=\@ne
2929       \fi
2930     \or
2931       \ifnum\lock@disp=\z@
2932         \@l@tempcntb=\z@ \@l@tempcnta=\@ne
2933       \fi
2934   \fi}
2935 %

```

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

```

2936 \newcommand*{\f@x@l@cks}{%
2937   \ifcase\@lock
2938     \or
2939       \global\@lock=\tw@
2940     \or \or
2941       \global\@lock=\z@
2942     \fi
2943   \ifcase\sub@lock
2944     \or
2945       \global\sub@lock=\tw@
2946     \or \or
2947       \global\sub@lock=\z@
2948     \fi}
2949 %
2950 %

```

## 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.

```

\leftpstartnum51
\rightpstartnum52 \newif\ifsidepstartnum
\ifsidepstartnum53 \newcommand*{\affixpstart@num}{%
2954   \ifsidepstartnum
2955     \if@twocolumn
2956       \if@firstcolumn
2957         \gdef\l@dld@ta{\llap{\leftpstartnum}}%
2958       \else
2959         \gdef\l@drd@ta{\rlap{\rightpstartnum}}%
2960       \fi
2961     \else
2962       \l@dttempcntb=\line@margin%
2963       \ifnum\l@dttempcntb>\@ne
2964         \advance\l@dttempcntb \page@num
2965       \fi
2966       \ifodd\l@dttempcntb
2967         \gdef\l@drd@ta{\rlap{\rightpstartnum}}%
2968       \else
2969         \gdef\l@dld@ta{\llap{\leftpstartnum}}%
2970       \fi
2971     \fi
2972   \fi
2973 }
2974 %
2975
2976
2977 \newif\ifpstartnum
2978 \pstartnumtrue
2979 \newcommand*{\leftpstartnum}{
2980   \ifpstartnum\thepstart
2981   \kern\linenumsep\fi
2982   \global\pstartnumfalse
2983 }
2984 \newcommand*{\rightpstartnum}{
2985   \ifpstartnum
2986   \kern\linenumsep
2987   \thepstart
2988   \fi
2989   \global\pstartnumfalse
2990 }

```

```
2991 %
```

## 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.

```
2992 \list@create{\inserts@list}
2993 %
```

`\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`.

```
2994 \newcommand*{\add@inserts}{%
2995   \global\let\add@inserts@next=\relax
2996 %
```

If `\inserts@list` is empty, there are not any more notes or insertions for this paragraph, and we need not waste our time.

```
2997   \ifx\inserts@list\empty \else
2998 %
```

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.

```
2999   \ifx\next@insert\empty
3000     \ifx\insertlines@list\empty
3001       \global\noteschanged@true
3002       \gdef\next@insert{100000}%
3003     \else
3004       \glp\insertlines@list\to\next@insert
3005     \fi
3006   \fi
3007 %
```

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.

```

3008 \ifnum\next@insert=\absline@num
3009 \glp\inserts@list\to\@insert
3010 \@insert
3011 \global\let\@insert=\undefined
3012 \global\let\next@insert=\empty
3013 \global\let\add@inserts@next=\add@inserts
3014 \fi
3015 \fi
3016 %

```

Make the recursive call, if necessary.

```

3017 \add@inserts@next}
3018
3019 %

```

`\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`.

```

3020 \newcommand{\add@Xgroupbyline}{%
3021 \unless\ifnocritical@%
3022 \def\do##1{%Looping on the series
3023 \let\olddo\do%Save the old \do macro, that is this macro itself!
3024 \def\do####1{%Looping on the ##1@forinserting command
3025 \ifcsdef{##1@forinserting@####1}{%
3026 \X@beforeinsertion{##1}%
3027 \if@ledgroup%
3028 \global\setbox\@nameuse{mp##1footins}=\vbox%
3029 \else%
3030 \insert\csname ##1footins\endcsname%
3031 \fi%
3032 {%
3033 \Xsetparindent{##1}%
3034 \ifcsdef{Xhsize\csuse{series@display##1}@##1}%
3035 {\hsize \csuse{Xhsize\csuse{series@display##1}@##1}}%
3036 {}%
3037 \if@ledgroup%
3038 \unvbox\@nameuse{mp##1footins}%
3039 \fi%
3040 \X@atbegininsertion{##1}%
3041 \ifcsstring{series@display##1}{normal}%
3042 {%
3043 \Xledsetnormalparstuff{##1}%
3044 \rule\z@\splittopskip%
3045 }%
3046 {}%
3047 \csuse{##1@forinserting@####1}%
3048 \strut\par\allowbreak%
3049 }%
3050 \global\csundef{##1@forinserting@####1}%

```

```

3051     }%
3052     {}%
3053     }%
3054     \ifcsdef{##1@forinserting}{%
3055     \dolistcslloop{##1@forinserting}%
3056     }{}%
3057     \global\csundef{##1@forinserting}%
3058     \let\do\olddo%Restore old do
3059     }%
3060     \dolistloop{\@series}%
3061     \fi%
3062 }%
3063
3064
3065 %

```

## 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  $-10000$ .

```

3066 \newcommand*{\add@penalties}{\@l@dttempcnta=\ballast@count
3067   \ifnum\num@lines>\@ne
3068     \global\advance\par@line \@ne
3069     \ifnum\par@line=\@ne
3070       \advance\@l@dttempcnta \clubpenalty
3071       \fi
3072       \@l@dttempcntb=\par@line \advance\@l@dttempcntb \@ne
3073       \ifnum\@l@dttempcntb=\num@lines
3074         \advance\@l@dttempcnta \widowpenalty
3075         \fi
3076       \ifnum\par@line<\num@lines
3077         \advance\@l@dttempcnta \interlinepenalty
3078         \fi
3079       \fi
3080       \ifnum\@l@dttempcnta=\z@
3081         \relax
3082       \else
3083         \ifnum\@l@dttempcnta>-10000
3084           \penalty\@l@dttempcnta
3085         \else
3086           \penalty -10000
3087         \fi
3088       \fi}
3089

```

```
3090 %
```

### 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

```
3091 \newcommand*{\flush@notes}{%
3092   \iftoggle{notfirstrun@ $\backslash$ jobname. $\backslash$ extensionchars $\backslash$ the $\backslash$ section@num}{%
3093     \@xloop%
3094     \ifx\inserts@list\empty \else%
3095       \gl@p\inserts@list\to\@insert%
3096       \@insert%
3097       \global\let\@insert=\undefined%
3098     \repeat%
3099   }%
3100 }%
3101
3102
3103 %
```

`\@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 Kabelschacht in *TUGboat* 8 (1987), pp. 184–5.

```
3104 \def\@xloop#1\repeat{%
3105   \def\body{#1\expandafter\body\fi}%
3106   \body}
3107
3108 %
```

### XII.4 Text before notes

`\set@Xtxtbeforenotes` The `\set@Xtxtbeforenotes` macro resets the `Xtxtbeforenotes@ $\langle$ series $\rangle$ @typeset` boolean to false. Just before the first note of the  $\langle$ series $\rangle$  in a page, the `\Xtxtbeforenotes` will be inserted.

```
3109 \newcommand{\set@Xtxtbeforenotes}{%
3110   \unless\ifnocritical@%
```

```

3111 \def\do##1{%
3112   \nottoggle{Xtxtbeforenotesonlyonce@##1}{-%
3113     \global\togglefalse{Xtxtbeforenotes@##1@typeset}%
3114   }{-%
3115   }%
3116   \dolistloop{\@series}%
3117   \fi%
3118 }%
3119 %

```

`\set@txtbeforenotesX` The `\set@txtbeforenotesX` does the same for the `\txtbeforenotesX`.

```

3120 \newcommand{\set@txtbeforenotesX}{%
3121   \unless\ifnofamiliar@%
3122   \def\do##1{%
3123     \nottoggle{txtbeforenotesonlyonceX@##1}{-%
3124       \global\togglefalse{txtbeforenotesX@##1@typeset}%
3125     }{-%
3126     }%
3127     \dolistloop{\@series}%
3128     \fi%
3129 }%
3130 %

```

`\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.

```

3131 \newcommand{\insert@Xtxtbeforenotes}[1]{%
3132   \nottoggle{Xtxtbeforenotes@#1@typeset}{-%
3133     \global\toggletrue{Xtxtbeforenotes@#1@typeset}%
3134     \ifcvoid{Xtxtbeforenotes@#1}{-%
3135       \ifcstring{series@display#1}{paragraph}%
3136       {\noindent\csuse{Xtxtbeforenotes@#1}}%
3137       {\expandafter\insert\cscname#1footins\endcscname%
3138         \bgroup%
3139           \noindent%
3140           \ifcsdef{\csuse{series@display#1}@begin@insert}{-%
3141             \csuse{\csuse{series@display#1}@begin@insert}{#1}%
3142           }{-%
3143             \strut\csuse{Xnotefontsize@#1}}%
3144             \csuse{Xtxtbeforenotes@#1}}%
3145         \egroup%
3146       }%
3147     }%
3148   }%
3149 }%
3150 }%
3151
3152
3153 \newcommand{\insert@txtbeforenotesX}[1]{%
3154   \nottoggle{txtbeforenotesX@#1@typeset}{-%

```

```

3155 \global\toggletrue{txtbeforenotesX@#1@typeset}%
3156 \ifcsvoid{txtbeforenotesX@#1}{-}{%
3157   \ifcsstring{series@displayX#1}{paragraph}%
3158     {\noindent\csuse{txtbeforenotesX@#1}}%
3159     {\expandafter\insert\csname footins#1\endcsname%
3160       \bgroup%
3161         \noindent%
3162         \ifcsdef{\csuse{series@displayX#1}@begin@insert}{%
3163           \csuse{\csuse{series@displayX#1}@begin@insert}{#1}%
3164         }-}%
3165         \strut\csuse{notefontsizeX@#1}\csuse{txtbeforenotesX@#1}%
3166       \egroup%
3167     }%
3168   }%
3169 }%
3170 {}%
3171 }%
3172
3173
3174 %

```

## 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.

```

3175 \def\select@lemmafont#1|#2|#3|#4|#5|#6|#7|{\select@lemmafont#7|}
3176 \def\select@lemmafont#1/#2/#3/#4|{%
3177   {\fontencoding{#1}\fontfamily{#2}\fontseries{#3}\fontshape{#4}}%
3178   \selectfont}
3179
3180 %

```

### 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.



```

3181 \newcommand*{\footnoteoptions@}[3]{%
3182   \def\do##1{%
3183     \ifstrequal{##1}{L}{% On the left side
3184       \xright@appenditem{\noexpand\setkeys[mac]{#3footnoteoption}{\
unexpanded{##1}}}\to\inserts@list%
3185       \global\advance\insert@count \@ne% Increment the left insert
counter.
3186     }%
3187     {%
3188       \xright@appenditem{\noexpand\setkeys[mac]{#3footnoteoption}{\
unexpanded{##1}}}\to\inserts@listR%
3189       \global\advance\insert@countR \@ne% Increment the right insert
counter insert.
3190     }%
3191   }%
3192   \notblank{#2}{\docsvlist{#2}}}% Parsing all options
3193 }
3194 %

```

### 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.

```

3195 \newcommandx*{\footnotelang@lua}[1][1=L,usedefault]{%
3196   \ifstrequal{##1}{L}{%
3197     \xright@appenditem{\csxdef{footnote@luatextextdir}{\the\textdir}}\to\
inserts@list%Know the dir of lemma
3198     \global\advance\insert@count \@ne%
3199     \xright@appenditem{\csxdef{footnote@luatexpardir}{\the\pardir}}\to\
inserts@list%Know the dir of lemma
3200     \global\advance\insert@count \@ne%
3201   }%
3202   {%
3203     \xright@appenditem{\csxdef{footnote@luatextextdir}{\the\textdir}}\to\
inserts@listR%Know the dir of lemma
3204     \global\advance\insert@countR \@ne%
3205     \xright@appenditem{\csxdef{footnote@luatexpardir}{\the\pardir}}\to\
inserts@listR%Know the dir of lemma
3206     \global\advance\insert@countR \@ne%
3207   }%
3208 }
3209 %

```

`\footnotelang@poly` `\footnotelang@poly` is called to remember the information about the language of a lemma when polyglossia is used.

```

3210 \newcommandx*{\footnotelang@poly}[1][1=L,usedefault]{%
3211   \ifstrequal{##1}{L}{%
3212     \if@RTL%
3213       \xright@appenditem{\csxdef{footnote@dir}{@RTLtrue}}\to\
inserts@list%Know the language used in the lemma
3214       \global\advance\insert@count \@ne%
3215     }%
3216   }%

```

```

3216 \xright@appenditem{\csxdef{footnote@dir}{@RTLfalse}}\to\
inserts@list%Know the language of lemma
3217 \global\advance\insert@count \@ne%
3218 \fi%
3219 \xright@appenditem{\csxdef{footnote@lang}{\expandonce\language}}\
to\inserts@list%Know the language of lemma
3220 \global\advance\insert@count \@ne%
3221 }%
3222 {%
3223 \ifRTL
3224 \xright@appenditem{\csxdef{footnote@dir}{@RTLtrue}}\to\
inserts@listR%Know the language of lemma
3225 \global\advance\insert@countR \@ne%
3226 \else
3227 \xright@appenditem{\csxdef{footnote@dir}{@RTLfalse}}\to\
inserts@listR%Know the language of lemma
3228 \global\advance\insert@countR \@ne%
3229 \fi
3230 \xright@appenditem{\csxdef{footnote@lang}{\expandonce\language}}\
to\inserts@listR%Know the language of lemma
3231 \global\advance\insert@countR \@ne%
3232 }%
3233 }
3234 %

```

### 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*.
- `\parfillskip` for ending of paragraph. Otherwise, we let it to `\@MM`, which is the standard L<sup>A</sup>T<sub>E</sub>X `\floatingpenalty`.

```

3235 \newcommand*\footsplitskips}{%
3236   \interlinepenalty=\interfootnotelinepenalty
3237   \unless\ifl@dprintingpages%
3238     \floatingpenalty=\@MM%
3239   \fi%
3240   \splittopskip=\ht\strutbox%
3241   \splitmaxdepth=\dp\strutbox%
3242   \leftskip=\z@skip%
3243   \rightskip=\z@skip%
3244   \parfillskip=0pt plus 1fil%
3245 }
3246
3247 %

```

`\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.

```

3248 \let\normalfootnoterule=\footnoterule
3249 %

```

## 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.

```

3250 \newcommandx{\Xarrangement}[2][1,usedefault]{%
3251   \def\do##1{%
3252     \csname Xarrangement@#2\endcsname{##1}%
3253   }%
3254   \ifstrempy{#1}%
3255     {%
3256       \dolistloop{\@series}%
3257     }%
3258     {
3259       \docsvlist{#1}%
3260     }%
3261 }%
3262 %

```

### 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\vsizerule
\let\Afootnote=\normalvfootnote \let\Afootfmt=\normalfootfmt
\let\Afootstart=\normalfootstart \let\Afootgroup=\normalfootgroup
\let\Afootnoterule=\normalfootnoterule
```

(Read *The TeXbook* in order to understand what are the counter, skip and dimen associated to an insertion.)

Instead of repeating ourselves, we define a `\Xarrangement@normal` macro that makes all these assignments for us, for any given series letter. This command is called when people use `\Xarrangement[⟨series⟩]{normal}`

Now we set up the `\Xarrangement@normal` macro itself. It takes one argument: the footnote series letter.

```
3263 \newcommand*{\Xarrangement@normal}[1]{%
3264   \csgdef{series@display#1}{normal}
3265   \expandafter\let\csname #1footstart\endcsname=\normalfootstart
3266   \expandafter\let\csname v#1footnote\endcsname=\normalvfootnote
3267   \expandafter\let\csname #1footfmt\endcsname=\normalfootfmt
3268   \expandafter\let\csname #1footgroup\endcsname=\normalfootgroup
3269   \expandafter\let\csname #1footnoterule\endcsname=%
3270                                     \normalfootnoterule
3271   \count\csname #1footins\endcsname=1000
3272   \dimen\csname #1footins\endcsname=\csuse{Xmaxhnotes@#1}
3273   \skip\csname #1footins\endcsname=\csuse{Xbeforenotes@#1}%
3274   \advance\skip\csname #1footins\endcsname by\csuse{Xafterrule@#1}%
3275   %
```

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.

```
3276   \csxdef{default@#1footins}{1000}%Use this to confine the notes to one
side only
3277   %
```

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).

```
3278   \ifnoledgroup@else%
3279     \expandafter\let\csname mpv#1footnote\endcsname=\mpnormalvfootnote
3280     \expandafter\let\csname mp#1footgroup\endcsname=\mpnormalfootgroup
3281     \count\csname mp#1footins\endcsname=1000
3282     \dimen\csname mp#1footins\endcsname=\csuse{Xmaxhnotes@#1}
3283     \skip\csname mp#1footins\endcsname=\csuse{Xbeforenotes@#1}%
3284     \advance\skip\csname mp#1footins\endcsname by\csuse{Xafterrule@#1}%
3285   \fi
3286 }
3287
3288 %
```

`\normalvfootnote` We now begin a series of commands that do “normal” footnote formatting: a format much like that implemented in PLAIN T<sub>E</sub>X, in which each footnote is a separate paragraph.

`\normalvfootnote` takes the series letter as #1 and the entire text of the footnote is #2. It does the `\insert` for this note, calling on the `\footfmt` macro for this note series to format the text of the note.

```

3289 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\normalvfootnote}[2]{%
3290   \iftoggle{Xgroupbyline@#1}{%In the case we use \Xgroupbyline, the
insertion is done later, in \add@Xgroupbyline.
3291     \prepare@Xgroupbyline{#1}{#2}{\normalvfootnote@inserted}%
3292   }{%In the case we don't use \Xgroupbyline, the insertion is made directly
3293     \X@beforeinsertion{#1}%
3294     \insert\csname #1footins\endcsname{%
3295       \X@atbegininsertion{#1}%
3296       \normalvfootnote@inserted{#1}{#2}%
3297     }%
3298   }%
3299 }%
3300 %

```

`\normalvfootnote@inserted` The `\normalvfootnote@inserted` macro is expanded to the content to be added to a `\insert` for normal critical footnote.

```

3301 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\
normalvfootnote@inserted}[2]{%
3302   \nottoggle{Xgroupbyline@#1}{\noindent}{\csuse{Xbhooknote@#1}%
3303   \csuse{Xnotefontsize@#1}%
3304   \iftoggle{Xgroupbyline@#1}{\strut}{}%
3305   \footsplitskips
3306   \ifl@dpairing\ifl@dpaging\else%
3307     \setXnoteswidthliketwocolumns@{#1}%
3308   \fi\fi%
3309   \setXnotespositionliketwocolumns@{#1}%
3310   \spaceskip=\z@skip \xspaceskip=\z@skip%
3311   \csname #1footfmt\endcsname #2{#1}%
3312 }%
3313 %

```

`\X@beforeinsertion`<sup>14</sup> `\newcommand{\X@beforeinsertion}[1]{%`

```

3315   \if@ledgroup\else%
3316     \insert@Xtxtbeforenotes{#1}%
3317   \fi%
3318   \csuse{Xbeforeinserting@#1}%
3319 }%
3320 %

```

`\beforeinsertion@X`<sup>21</sup> `\newcommand{\beforeinsertion@X}[1]{%`

```

3322   \if@ledgroup\else%
3323     \insert@txtbeforenotesX{#1}%
3324   \fi%
3325   \csuse{beforeinsertingX@#1}%

```

```
3326 }%
3327 %
```

```
\X@atbegininsertion28 \newcommand{\X@atbegininsertion}[1]{%
3329 \hsize=\expandafter\dimexpr\csuse{Xwidth@#1}\relax%
3330 }%
3331 %
```

And somewhat different versions of `\normalvfootnote` and `\normalvfootnote@inserted` for minipages.

```
\mpnormalvfootnote32 \notbool{parapparatus@}{\newcommand*{\newcommand}{\mpnormalvfootnote}[2]{%
3333 \iftoggle{Xgroupbyline@#1}{%
3334 \prepare@Xgroupbyline{#1}{#2}{\mpnormalvfootnote@inserted}%
3335 }%
3336 {%
3337 \global\setbox\@nameuse{mp#1footins}%
3338 \vbox{%
3339 \unvbox\@nameuse{mp#1footins}%
3340 \mpnormalvfootnote@inserted{#1}{#2}%
3341 }%
3342 }%
3343 }%
3344 %
3345 %
```

```
\mpnormalvfootnote@inserted46 \newcommand{\mpnormalvfootnote@inserted}[2]{%
3347 \noindent\csuse{Xhooknote@#1}%
3348 \csuse{Xnotefontsize@#1}%
3349 \hsize\columnwidth%
3350 \@parboxrestore%
3351 \color@begingroup%
3352 \csname #1footfmt\endcsname #2{#1}\color@endgroup%
3353 }%
3354 %
```

`\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.

```
3355 \notbool{parapparatus@}{\newcommand*{\newcommand}{\normalfootfmt}[4]{%
3356 \Xstorelineinfo{#1}{#4}%
3357 \nottoggle{Xgroupbyline@#4}{\Xledsetnormalparstuff{#4}}{ }%
3358 \Xsethangindent{#4}%
3359 \nottoggle{Xgroupbyline@#4}{\rule\z@\splittopskip}{ }%
3360 {\printlinefootnote{#1}{#4}}%
3361 \print@lemma{#1}{#2}{#4}%
3362 \csuse{Xwrapcontent@#4}{#3}%
```

```

3363 \nottoggle{Xgroupbyline@#4}{\strut\par}{}%
3364 }%
3365 %

```

`\normalfootstart` `\normalfootstart` is a standard footnote-starting macro, called in the output routine whenever there are footnotes of this series to be printed: it skips a bit and then draws a rule.

Any `\footstart` macro must put onto the page something that takes up space exactly equal to the `\skip\Xfootins` value for the associated series of notes.  $\TeX$  makes page computations based on that `\skip` value, and the output pages will suffer from spacing problems if what you add takes up a different amount of space.

But if the `\skip\Xprenotes@` is greater than 0 pt, it is used instead of `\skip\footins` for the first printed series in one page.

The `\leftskip` and `\rightskip` values are both zeroed here. Similarly, these skips are cancelled in the `\vfootnote` macros for the various types of notes. Strictly speaking, this is necessary only if you are using paragraphed footnotes, but we have put it here and in the other `\vfootnote` macros too so that the behavior of `reledmac` in this respect is general across all footnote types. What this means is that any `\leftskip` and `\rightskip` you specify applies to the main text, but not the footnotes. The footnotes continue to be of width `\hsize`.

```

3366 \newcommand*{\normalfootstart}[1]{%
3367 %

```

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. 236](#). Here is part of this algorithm, when the block of notes are ready to be printed.

```

3368 \ifdimequal{0pt}{\Xprenotes@}{}%
3369 {%
3370 \iftoggle{Xprenotes@}{%
3371 \togglefalse{Xprenotes@}%
3372 \skip\csname #1footins\endcsname=%
3373 \glueexpr\cuse{Xprenotes@}+\cuse{Xafterrule@#1}\relax%
3374 }%
3375 }%
3376 }%
3377 \vskip\skip\csname #1footins\endcsname%
3378 %

```

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.

```

3379 \leftskip0pt \rightskip0pt
3380 \parfillskip=0pt plus 1fil
3381 \ifl@dpairing\else%
3382 \hsize=\old@hsize%
3383 \fi%
3384 \setXnoteswidthliketwocolumns@{#1}%
3385 \setXnotespositionliketwocolumns@{#1}%
3386 %

```

And now, print the footnote's rule to finish the footnote's introduction.

```

3387 \print@Xfootnoterule{#1}%
3388 }%
3389 %

```

**\normalfootgroup** `\normalfootgroup` is a standard footnote-grouping macro: it sends the contents of the footnote-insert box to the output page without alteration.

```

3390 \newcommand*{\normalfootgroup}[1]{%
3391   \csuse{Xbhookgroup@#1}%
3392   \unvbox\csname #1footins\endcsname%
3393   \hsize=\old@hsize%
3394   }%
3395 %
3396 %

```

**\mpnormalfootgroup** A somewhat different version for minipages. Note that, in this case, we do not make distinctions between the `\Xfootgroup` and `\Xfootstarts` macros.

```

3397 \unless\ifnoledgroup@
3398 \newcommand*{\mpnormalfootgroup}[1]{%
3399   \vskip\skip\@nameuse{mp#1footins}
3400   \ifl@dpairing\ifparledgroup%
3401     \leavevmode\marks\parledgroup@{begin}%
3402     \marks\parledgroup@series{#1}%
3403     \marks\parledgroup@type{Xfootnote}%
3404     \fi\fi\normalcolor%
3405     \ifparledgroup%
3406       \ifl@dpairing%
3407       \else%
3408         \setXnoteswidthliketwocolumns@{#1}%
3409         \setXnotespositionliketwocolumns@{#1}%
3410         \print@Xfootnoterule{#1}%
3411       \fi%
3412     \else%
3413       \setXnoteswidthliketwocolumns@{#1}%
3414       \setXnotespositionliketwocolumns@{#1}%
3415       \print@Xfootnoterule{#1}%
3416     \fi%
3417     \setlength{\parindent}{0pt}
3418     \csuse{Xbhookgroup@#1}%
3419     \unvbox\csname mp#1footins\endcsname}}
3420 \fi
3421 %

```

### XIII.6.3 Paragraphed footnotes

The paragraphed-footnote option reformats all the footnotes of one series for a page into a single paragraph; this is especially appropriate when the notes are numerous and brief. The code is based on *The TeXbook*, pp. 398–400, with alterations for our environment. This algorithm uses a considerable amount of save-stack space: a  $\TeX$  of ordinary size may not be able to handle more than about 100 notes of this kind on a page.

**\Xarrangement@paragraph** The `\Xarrangement@paragraph` macro sets up everything for one series of the footnotes so that they will be paragraphed; it takes the series letter as argument. We include



the setting of `\count\footins` to 1000 for the footnote series just in case user is switching to paragraphed footnotes after having columnar ones, since they change this value (see below).

The argument of `\Xarrangement@footparagraph` is the letter denoting the series of notes to be paragraphed.

```

3422 \newcommand*\Xarrangement@paragraph}[1]{%
3423   \csgdef{series@display#1}{paragraph}
3424   \expandafter\let\csname #1footstart\endcsname=\parafootstart
3425   \expandafter\let\csname v#1footnote\endcsname=\paravfootnote
3426   \expandafter\let\csname #1footfmt\endcsname=\parafootfmt
3427   \expandafter\let\csname #1footgroup\endcsname=\parafootgroup
3428   \count\csname #1footins\endcsname=1000
3429   \csxdef{default@#1footins}{1000}%Use this to confine the notes to one
side only
3430   \dimen\csname #1footins\endcsname=\csuse{Xmaxhnotes@#1}
3431   \skip\csname #1footins\endcsname=\csuse{Xbeforenotes@#1}%
3432   \advance\skip\csname #1footins\endcsname by\csuse{Xafterrule@#1}%
3433   \para@footsetup{#1}
3434 %

```

And the extra setup for minipages.

```

3435   \ifnoledgroup@else
3436     \expandafter\let\csname mpv#1footnote\endcsname=\mpparavfootnote
3437     \expandafter\let\csname mp#1footgroup\endcsname=\mpparavfootgroup
3438     \count\csname mp#1footins\endcsname=1000
3439     \dimen\csname mp#1footins\endcsname=\csuse{Xmaxhnotes@#1}
3440     \skip\csname mp#1footins\endcsname=\csuse{Xbeforenotes@#1}%
3441     \advance\skip\csname mp#1footins\endcsname by\csuse{Xafterrule@#1}%
3442     \fi
3443   }
3444 %

```

`\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.

```

3445 \providecommand{\footfudgefiddle}{64}
3446 %

```

`\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`.

```

3447 \newcommand*\para@footsetup}[1]{\csuse{Xhookgroup@#1}\csuse{
Xnotefontsize@#1}
3448   \setXnoteswidthliketwocolumns@{#1}%
3449   \ifcempty{Xwidth@#1}%
3450     {}%
3451     {\columnwidth=\expandafter\dimexpr\csuse{Xwidth@#1}\relax}%

```

```

3452 \dimen0=\baselineskip
3453 \multiply\dimen0 by 1024
3454 \divide \dimen0 by \columnwidth \multiply\dimen0 by \footfudgefiddle\
relax
3455 \csxdef{#1footfudgefactor}{%
3456   \expandafter\strip@pt\dimen0 }}
3457
3458 %

```

\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 \hsiz. So the paragraph of notes needs to be that wide.

The argument of the macro is again the note series letter.

```

3459 \newcommand*{\parafootstart}[1]{%
3460   \rightskip=0pt \leftskip=0pt%
3461   \Xsetparindent{#1}%
3462   \ifdimequal{0pt}{\Xprenotes@}{-}{%
3463     {%
3464       \iftoggle{Xprenotes@}{%
3465         \togglefalse{Xprenotes@}%
3466         \skip\csname #1footins\endcsname=%
3467         \glueexpr\csuse{Xprenotes@}+\csuse{Xafterrule@#1}\relax%
3468       }%
3469     }%
3470   }%
3471   \vskip\skip\csname #1footins\endcsname%
3472   \setXnoteswidthliketwocolumns@{#1}%
3473   \setXnotespositionliketwocolumns@{#1}%
3474   \print@Xfootnoterule{#1}%
3475   \let\old@bidi@RTL@everypar\old@bidi@RTL@everypar%
3476   \let\old@bidi@RTL@everypar\empty%
3477   \noindent\leavevmode%
3478   \let\old@bidi@RTL@everypar\old@bidi@RTL@everypar%
3479   \parfillskip=0pt plus 1fil%
3480 }%
3481 %

```

**\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. If you

later unbox these hboxes and stick them together, as the *TeXbook* macros do to make these footnotes, you lose the ability to hyphenate after an explicit hyphen. This can lead to overfull hboxes when you would not expect to find them, and to the uninitiated it might be very hard to see why the problem had arisen.<sup>32</sup>

Wayne Sullivan pointed out to us another subtle problem that arises from the same cause:  $\TeX$  also leaves the `\language` whatsit nodes out of the horizontal list.<sup>33</sup> So changes from one language to another will not invoke the proper hyphenation rules in such footnotes. Since critical editions often do deal with several languages, especially in a footnotes, we really ought to get this bit of code right.

To get around these problems, Wayne suggested emendations to the *TeXbook* versions of these macros which are broadly the same as those described by Michael: the central idea (also suggested by Donald Knuth in a letter to Michael) is to avoid collecting the text in an `\hbox` in the first place, but instead to collect it in a `\vbox` whose width is (virtually) infinite. The text is therefore typeset in unrestricted horizontal mode, as a paragraph consisting of a single long line. Later, there is an extra level of unboxing to be done: we have to unpack the `\vbox`, as well as the hboxes inside it, but that is not too hard. For details, we refer you to Michael's article, where the issues are clearly explained.<sup>34</sup> Michael's unboxing macro is called `\Xunvxh`: `unvbox`, extract the last line, and `unhbox` it.

Doing things this way has an important consequence: as Michael pointed out, you really can't put an explicit line-break into a note built in a `\vbox` the way we are doing.<sup>35</sup> In other words, be very careful not to use `\break`, or `\penalty-10000`, or any equivalent inside your para-footnote. If you do, most of the note will probably disappear. You *are* allowed to make strong suggestions; in fact `\penalty-9999` will be quite okay. Just do not make the break mandatory. We have not applied any of Michael's solutions here, since we feel that the problem is exiguous, and `reledmac` is quite baroque enough already. If you think you are having this problem, look up Michael's solutions.

One more thing; we set `\leftskip` and `\rightskip` to zero. This has the effect of neutralizing any such skips which may apply to the main text (cf. XIII.6.2 p. 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.

```

3482 \newcommand*{\paravfootnote}[2]{%
3483   \csuse{Xbeforeinserting@#1}%
3484   \insert\csname #1footins\endcsname
3485   \bgroup
3486   \csuse{Xnotefontsize@#1}
3487   \footsplitskips
3488   \setbox0=\vbox{\hsize=\maxdimen%
3489     \let\bid@RTL@everypar@\empty%
3490     \insert@Xtxtbeforenotes{#1}%
3491     \noindent\csuse{Xhooknote@#1}%
3492     \csname #1footfmt\endcsname #2{#1}}%
3493   \setbox0=\hbox{\Xunvxh{0}{#1}}%
3494   \dp0=0pt

```

<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).

<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.

```

3495 \ht0=\csname #1footfudgefactor\endcsname\wd0
3496 %

```

Here we produce the contents of the footnote from box 0, and add a penalty of 0 between boxes in this insert.

```

3497 \if@RTL\noindent \leavevmode\fi\box0%
3498 \penalty0
3499 \egroup}
3500 %
3501 %

```

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.

```

3502 \newcommand*{\mpparavfootnote}[2]{%
3503 \global\setbox\@nameuse{mp#1footins}\vbox{%
3504 \unvbox\@nameuse{mp#1footins}%
3505 \csuse{Xnotefontsize@#1}
3506 \footplitskips
3507 \setbox0=\vbox{\hsize=\maxdimen%
3508 \let\@RTL@everypar\@empty%
3509 \insert@Xtxtbeforenotes{#1}%
3510 \noindent\color@begingroup%
3511 \csuse{Xbhooknote@#1}%
3512 \csname #1footfmt\endcsname #2{#1}\color@endgroup}%
3513 \setbox0=\hbox{\Xunvxh{0}{#1}}%
3514 \dp0=\z@
3515 \ht0=\csname #1footfudgefactor\endcsname\wd0
3516 \box0
3517 \penalty0
3518 }}
3519 %
3520 %

```

`\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`.

```

3521 \newcommand*{\Xunvxh}[2]{%
3522 \setbox0=\vbox{\unvbox#1%
3523 \global\setbox1=\lastbox}%

```

```

3524 \unhbox1
3525 \unskip           % remove \rightskip,
3526 \unskip           % remove \parfillskip,
3527 \unpenalty        % remove \penalty of 10000,
3528 \hskip\csuse{Xafternote@#2}\relax}% add the glue to go between the notes
3529
3530 %

```

`\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).

```

3531 \newcommand*{\parafootfmt}[4]{%
3532   \Xstorelineinfo{#1}{#4}%
3533   \Xinsertparafootsep{#4}%
3534   \ledsetnormalparstuff@common%
3535   \printlinefootnote{#1}{#4}%
3536   \print@lemma{#1}{#2}{#4}%
3537   \csuse{Xwrapcontent@#4}{#3}%
3538   \penalty-10 }
3539 %

```

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.

```

3540 \newcommand*{\parafootgroup}[1]{%
3541   \hsize=\expandafter\dimexpr\csuse{Xwidth@#1}\relax%
3542   \unvbox\csname #1footins\endcsname
3543   \ifcsstring{Xragged@#1}{L}{\RaggedLeft}{}%
3544   \ifcsstring{Xragged@#1}{R}{\RaggedRight}{}%
3545   \makehboxofhboxes
3546   \setbox0=\hbox{\unhbox0 \removehboxes}%
3547   \csuse{Xbhookgroup@#1}%
3548   \csuse{Xnotefontsize@#1}%
3549   \unhbox0\par%
3550   \global\hsize=\old@hsize%
3551   }%
3552
3553 %

```

`\mpparafootgroup` The minipage version.

```

3554 \newcommand*{\mpparafootgroup}[1]{%
3555   \setXnoteswidthliketwocolumns@{#1}%

```

```

3556 \vskip\skip\@nameuse{mp#1footins}
3557 \ifl@dpairing\ifparledgroup%
3558 \leavevmode\marks\parledgroup@{begin}%
3559 \marks\parledgroup@series{#1}%
3560 \marks\parledgroup@type{Xfootnote}%
3561 \fi\fi\normalcolor
3562 \ifparledgroup%
3563 \ifl@dpairing%
3564 \else%
3565 \setXnoteswidthliketwocolumns@{#1}%
3566 \setXnotespositionliketwocolumns@{#1}%
3567 \print@Xfootnoterule{#1}%%
3568 \fi%
3569 \else%
3570 \setXnoteswidthliketwocolumns@{#1}%
3571 \setXnotespositionliketwocolumns@{#1}%
3572 \print@Xfootnoterule{#1}%
3573 \fi%
3574 \unvbox\cscname mp#1footins\endcscname
3575 \ifcsstring{Xragged@#1}{L}{\RaggedLeft}{}%
3576 \ifcsstring{Xragged@#1}{R}{\RaggedRight}{}%
3577 \makehboxofhboxes
3578 \setbox0=\hbox{\unhbox0 \removehboxes}%
3579 \csuse{Xbhookgroup@#1}%
3580 \csuse{Xnotefontsize@#1}%
3581 \Xsetparindent{#1}%
3582 \unhbox0\par}}
3583
3584 %

```

And finally, the two macros which are required to transform the long horizontal box stored in the insert' box to a printable text.

```

\makehboxofhboxes 85 \newcommand*{\makehboxofhboxes}{\setbox0=\hbox{}}%
\removehboxes 86 \loop
3587 \unpenalty
3588 \setbox2=\lastbox
3589 \ifhbox2
3590 \setbox0=\hbox{\box2\unhbox0}%
3591 \repeat}
3592
3593 \newcommand*{\removehboxes}{\setbox0=\lastbox
3594 \ifhbox0{\removehboxes}\unhbox0 \fi}
3595
3596 %

```

**Insertion of the footnotes separator** The command `\Xinsertparafootsep{<series>}` must be called at the beginning of `\parafootftm`.

```

\prevpage@num 97 \newcommand{\Xinsertparafootsep}[1]{%
\Xinsertparafootsep 98 \ifledRcol@%
3599 \ifnumequal{\csuse{#1prevpage@numR}}{\page@numR}%

```

```

3600     {\ifcsdef{prevline#1}% Be sur \prevline#1 exists.
3601       {\ifcsequal{prevline#1}{lineinfo@}%
3602         {\ifcsequal{Xsymlinenum@#1}{\csuse{Xparafootsep@#1}}{}}%
3603         {\csuse{Xparafootsep@#1}}%
3604       }%
3605     {\csuse{Xparafootsep@#1}}%
3606   }%
3607 }%
3608 \global\csname #1prevpage@numR\endcsname=\page@numR%
3609 \else%
3610   \ifnumequal{\csuse{#1prevpage@num}}{\page@num}%
3611     {\ifcsdef{prevline#1}% Be sur \prevline#1 exists.
3612       {\ifcsequal{prevline#1}{lineinfo@}%
3613         {\ifcsequal{Xsymlinenum@#1}{\csuse{Xparafootsep@#1}}{}}%
3614         {\csuse{Xparafootsep@#1}}%
3615       }%
3616     {\csuse{Xparafootsep@#1}}%
3617   }%
3618 }%
3619 \global\csname #1prevpage@num\endcsname=\page@num%
3620 \fi%
3621 }
3622 %

```

### XIII.6.4 Columnar footnotes

#### Common tools

`\rigidbalance` We will now define macros for three-column notes and two-column notes. Both sets of macros will use `\rigidbalance`, which splits a box (#1) into a number (#2) of columns, each with a space (#3) between the top baseline and the top of the `\vbox`. The `\Xrigidbalance` macro is taken from *The TeXbook*, p. 397, with a slight change to the syntax of the arguments so that they do not depend on white space. Note also the extra unboxing in `\splitoff`, which allows the new `\vbox` to have its natural height as it goes into the alignment.

The  $\TeX$  `\line` macro has no relationship to the TeX `\line`. The  $\TeX$  equivalent is `\@@line`.

We do not call directly `\rigidbalance`, but we call `\Xrigidbalance` for critical notes and `\rigidbalanceX` for familiar notes. Both of them call `\rigidbalance`.

```

3623 \newcount\@k \newdimen\@h
3624 \newcommand*\Xrigidbalance}[3]{%
3625   \hsize=\expandafter\dimexpr\csuse{Xwidth@\@currentseries}\relax%
3626   \rigidbalance{#1}{#2}{#3}%
3627 }%
3628
3629 \newcommand*\rigidbalanceX}[3]{%
3630   \hsize=\expandafter\dimexpr\csuse{widthX@\@currentseries}\relax%
3631   \rigidbalance{#1}{#2}{#3}%
3632 }%
3633
3634 \newcommand*\rigidbalance}[3]{%
3635   \setbox0=\box#1 \@k=#2 \@h=#3%

```

```

3636 \@@line{\splittopskip=\@h \vbadness=\@M \hfilneg
3637 \valign{##\vfil\cr\dosplits}}
3638
3639 \newcommand*{\dosplits}{\ifnum\@k>0 \noalign{\hfil}\splitoff
3640 \global\advance\@k-1\cr\dosplits\fi}
3641
3642 \newcommand*{\splitoff}{\dimen0=\ht0
3643 \divide\dimen0 by\@k \advance\dimen0 by\@h
3644 \setbox2 \vsplit0 to \dimen0
3645 \unvbox2 }
3646
3647 %

```

### Three columns

```

\Xarrangement@threecol148 \newcommand*{\Xarrangement@threecol}[1]{%
3649 \csgdef{series@display#1}{threecol}
3650 \expandafter\let\csname v#1footnote\endcsname=\threecolvfootnote
3651 \expandafter\let\csname #1footfmt\endcsname=\threecolfootfmt
3652 \expandafter\let\csname #1footgroup\endcsname=\threecolfootgroup
3653 \dimen\csname #1footins\endcsname=\csuse{Xmaxhnotes@#1}%
3654 \skip\csname #1footins\endcsname=\csuse{Xbeforenotes@#1}%
3655 \advance\skip\csname #1footins\endcsname by\csuse{Xafterrule@#1}%
3656 \threecolfootsetup{#1}
3657 %

```

The additional setup for minipages.

```

3658 \ifnoledgroup@else
3659 \expandafter\let\csname mpv#1footnote\endcsname=\mpnormalvfootnote
3660 \expandafter\let\csname mp#1footgroup\endcsname=\mpthreecolfootgroup
3661 \skip\csname mp#1footins\endcsname=\csuse{Xbeforenotes@#1}%
3662 \advance\skip\csname mp#1footins\endcsname by\csuse{Xafterrule@#1}%
3663 \mpthreecolfootsetup{#1}
3664 \fi
3665 }
3666
3667 %

```

The `\footstart` and `\footnoterule` macros for these notes assume the normal values (XIII.6.2 p. 183 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.

```

3668 \newcommand*\threecolfootsetup}[1]{%
3669   \count\csname #1footins\endcsname 333
3670   \csxdef{default@#1footins}{333}%Use this to confine the notes to one
side only
3671   \multiply\dimen\csname #1footins\endcsname \thr@@}
3672   %

```

`\mpthreecolfootsetup` The setup for minipages.

```

3673 \newcommand*\mpthreecolfootsetup}[1]{%
3674   \count\csname mp#1footins\endcsname 333
3675   \multiply\dimen\csname mp#1footins\endcsname \thr@@}
3676   %
3677   %

```

`\threecolvfootnote` `\threecolvfootnote` This is the `\vfootnote` command for three-column notes. However, most of the code is deported on `\threecolvfootnote@inserted`. The call to `\Xnotefontsize@<s>` ensures that the `\splittopskip` and `\splitmaxdepth` take their values from the right `\strutbox`: the one used in a footnotes. Note especially the importance of temporarily reducing the `\hsize` to 0.3 of its normal value. This determines the widths of the individual columns. So if the normal `\hsize` is (say) 10 cm, then each column will be  $0.3 \times 10 = 3$  cm wide, leaving a gap of 1 cm spread equally between columns (i.e., .5 cm between each).

The arguments are #1 the note series letter and #1 the full text of the note (including numbers, lemma and text).

```

3678 \notbool{parapparatus@}\newcommand*\newcommand*\threecolvfootnote}[2]{%
3679   \iftoggle{Xgroupbyline@#1}{%
3680     \prepare@Xgroupbyline{#1}{#2}\threecolvfootnote@inserted}%
3681   }%
3682   {%
3683     \let\bidirTL@everypar\relax%
3684     \X@beforeinsertion{#1}%
3685     \insert\csname #1footins\endcsname{%
3686       \threecolvfootnote@inserted{#1}{#2}%
3687     }%
3688   }%
3689 }%
3690 %

```

```

reecolvfootnote@inserted}
3691 \notbool{parapparatus@}\newcommand*\newcommand*\threecolvfootnote@inserted}[2]{%
3692   \hsize=\expandafter\dimexpr\csuse{Xwidth@#1}\relax%
3693   \noindent\csuse{Xhooknote@#1}%
3694   \csuse{Xnotefontsize@#1}%
3695   \footsplitskips%
3696   \csname #1footfmt\endcsname #2{#1}%
3697 }%
3698 %

```

`\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.

```

3699 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\threecolfootfmt}[4]{%
3700   \xstorelineinfo{#1}{#4}%
3701   \threecol@begin@insert{#4}%
3702   \hspace{\parindent}%
3703   \printlinefootnote{#1}{#4}%
3704   \print@lemma{#1}{#2}{#4}%
3705   \csuse{Xwrapcontent@#4}{#3}%
3706   \nottoggle{Xgroupbyline@#4}%
3707     {\strut\par\allowbreak}%
3708   }%
3709 }%
3710 %

```

`\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`.

```

3711 \newcommand{\threecol@begin@insert}[1]{%
3712   \normal@pars%
3713   \nottoggle{Xgroupbyline@#1}%
3714     {\hspace \csuse{Xsizethreecol@#1}}%
3715   }%
3716   \Xsetparindent{#1}%
3717   \tolerance=5000%
3718   \Xsethangindent{#1}%
3719   \@tempdima=\parindent%
3720   \csuse{Xcolalign@#1}%
3721   \parindent=\@tempdima%
3722   \strut%
3723 }%
3724 %

```

`\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.

```

3725 \newcommand*{\threecolfootgroup}[1]{%
3726   \begin@group%
3727     \csuse{Xhookgroup@#1}%
3728     \csuse{Xnotefontsize@#1}%
3729     \par%
3730     \splittopskip=\ht\strutbox%
3731     \expandafter%
3732     \Xrigidbalance\csname #1footins\endcsname \thr@@ \splittopskip%

```

```

3733 \endgroup%
3734 }%
3735 %

```

`\mpthreecolfootgroup` The setup for minipages.

```

3736 \newcommand*\mpthreecolfootgroup}[1]{%
3737   \vskip\skip\@nameuse{mp#1footins}
3738   \ifl@dpairing\ifparledgroup%
3739     \leavevmode\marks\parledgroup@{begin}%
3740     \marks\parledgroup@series{#1}%
3741     \marks\parledgroup@type{Xfootnote}%
3742   \fi\fi\normalcolor
3743   \ifparledgroup%
3744     \ifl@dpairing%
3745     \else%
3746       \setXnoteswidthliketwocolumns@{#1}%
3747       \setXnotespositionliketwocolumns@{#1}%
3748       \print@Xfootnoterule{#1}%
3749     \fi%
3750   \else%
3751     \setXnoteswidthliketwocolumns@{#1}%
3752     \setXnotespositionliketwocolumns@{#1}%
3753     \print@Xfootnoterule{#1}%
3754   \fi%
3755   \csuse{Xbhookgroup@#1}\par%
3756   \splittopskip=\ht\strutbox
3757   \expandafter
3758   \Xrigidbalance\csname mp#1footins\endcsname \thr@@ \splittopskip}}
3759
3760 %

```

## Two columns

```

\Xarrangement@twocol61 \newcommand*\Xarrangement@twocol}[1]{%
3762   \csgdef{series@display#1}{twocol}
3763   \expandafter\let\csname v#1footnote\endcsname=\twocolvfootnote
3764   \expandafter\let\csname #1footfmt\endcsname=\twocolfootfmt
3765   \expandafter\let\csname #1footgroup\endcsname=\twocolfootgroup
3766   \dimen\csname #1footins\endcsname=\csuse{Xmaxhnotes@#1}%
3767   \skip\csname #1footins\endcsname=\csuse{Xbeforenotes@#1}%
3768   \advance\skip\csname #1footins\endcsname by\csuse{Xafterrule@#1}%
3769   \twocolfootsetup{#1}
3770 %

```

The additional setup for minipages.

```

3771   \ifnoledgroup@else
3772     \expandafter\let\csname mpv#1footnote\endcsname=\mpnormalvfootnote
3773     \expandafter\let\csname mp#1footgroup\endcsname=\mptwocolfootgroup
3774     \skip\csname mp#1footins\endcsname=\csuse{Xbeforenotes@#1}%
3775     \advance\skip\csname mp#1footins\endcsname by\csuse{Xafterrule@#1}%
3776     \mptwocolfootsetup{#1}
3777   \fi

```

```

3778 }
3779
3780 %

\twocolfootsetup Here is a series of macros which are very similar to their three-column counterparts. In
\twocolvfootnote this case, each note is assumed to contribute only a half a line of text. And the notes are
\twocolvfootnote@inserted set in columns giving a gap between them of one tenth of the \hszize.
\twocolfootfmt
\twocolfootgroup
3781 \newcommand*{\twocolfootsetup}[1]{%
3782 \count\csname #1footins\endcsname 500
3783 \csxdef{default@#1footins}{500}%
3784 \multiply\dimen\csname #1footins\endcsname \tw@}
3785 %

3786 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\twocolvfootnote}[2]{%
3787 \iftoggle{Xgroupbyline@#1}{%
3788 \prepare@Xgroupbyline{#1}{#2}{\twocolvfootnote@inserted}%
3789 }{%
3790 \let\bidir@RTL@everypar\relax%
3791 \X@beforeinsertion{#1}%
3792 \insert\csname #1footins\endcsname{%
3793 \twocolvfootnote@inserted{#1}{#2}%
3794 }%
3795 }%
3796 }%
3797 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\
twocolvfootnote@inserted}[2]{%
3798 \hszize=\expandafter\dimexpr\csuse{Xwidth@#1}\relax%
3799 \noindent\csuse{Xbhooknote@#1}%
3800 \csuse{Xnotefontsize@#1}%
3801 \footsplitskips%
3802 \csname #1footfmt\endcsname #2{#1}%
3803 }%
3804 %

3805 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\twocolfootfmt}[4]{% 4th
arg is optional, for backward compatibility
3806 \Xstorelineinfo{#1}{#4}%
3807 \twocol@begin@insert{#4}%
3808 \hspace{\parindent}%
3809 \printlinefootnote{#1}{#4}%
3810 \print@lemma{#1}{#2}{#4}%
3811 \csuse{Xwrapcontent@#4}{#3}%
3812 \nottoggle{Xgroupbyline@#4}%
3813 {\strut\par\allowbreak}%
3814 }%
3815 }%
3816 %

3817 \newcommand{\twocol@begin@insert}[1]{%
3818 \normal@pars%
3819 \nottoggle{Xgroupbyline@#1}%
3820 {\hszize \csuse{Xhsizetwocol@#1}%
3821 }%

```

```

3822 \Xsetparindent{#1}%
3823 \tolerance=5000%
3824 \Xsethangindent{#1}%
3825 \@tempdima=\parindent%
3826 \csuse{Xcolalign@#1}%
3827 \parindent=\@tempdima%
3828 \strut%
3829 }%
3830
3831 \newcommand*\twocolfootgroup}[1]{%
3832 \begingroup%
3833 \csuse{Xbhookgroup@#1}%
3834 \csuse{Xnotefontsize@#1}%
3835 \par%
3836 \splittopskip=\ht\strutbox%
3837 \expandafter%
3838 \Xrigidbalance\csname #1footins\endcsname \tw@ \splittopskip%
3839 \endgroup%
3840 }%
3841
3842 %

```

`\mptwocolfootsetup` The versions for minipages.

`\mptwocolfootgroup`

```

3843 \newcommand*\mptwocolfootsetup}[1]{%
3844 \count\csname mp#1footins\endcsname 500
3845 \multiply\dimen\csname mp#1footins\endcsname \tw@}
3846 %
3847 \newcommand*\mptwocolfootgroup}[1]{%
3848 \vskip\skip\@nameuse{mp#1footins}
3849 \ifl@dpairing\ifparledgroup%
3850 \leavevmode\marks\parledgroup@{begin}%
3851 \marks\parledgroup@series{#1}%
3852 \marks\parledgroup@type{Xfootnote}%
3853 \fi\fi\normalcolor
3854 \ifparledgroup%
3855 \ifl@dpairing%
3856 \else%
3857 \setXnoteswidthliketwocolumns@{#1}%
3858 \setXnotespositionliketwocolumns@{#1}%
3859 \print@Xfootnoterule{#1}%
3860 \fi%
3861 \else%
3862 \setXnoteswidthliketwocolumns@{#1}%
3863 \setXnotespositionliketwocolumns@{#1}%
3864 \print@Xfootnoterule{#1}%
3865 \fi%
3866 \csuse{Xbhookgroup@#1}\par%
3867 \splittopskip=\ht\strutbox
3868 \expandafter
3869 \Xrigidbalance\csname mp#1footins\endcsname \tw@ \splittopskip}}
3870
3871 %

```

### XIII.7 Footnote paragraph indent

`\Xsetparindent` and `\setparindentX` These two commands set the paragraph indentation of the footnotes, depending on the settings of the user.

```

3872 \newcommand{\Xsetparindent}[1]{%
3873   \nottoggle{Xparindent@#1}{%
3874     \parindent=\z@%
3875   }%
3876   {%
3877     \ifdef{\parindent@beforestanza}{\parindent=\parindent@beforestanza}{}%
3878   }%
3879 }%
3880 \newcommand{\setparindentX}[1]{%
3881   \nottoggle{parindentX@#1}{%
3882     \parindent=\z@%
3883   }%
3884   {%
3885     \ifdef{\parindent@beforestanza}{\parindent=\parindent@beforestanza}{}%
3886   }%
3887 }%
3888 %

```

### XIII.8 Footnote hanging indent

`\Xsethangindent` and `\sethangindentX` `\Xsethangindent` sets the hangindent for a critical footnote, while `\sethangindentX` does it for a familiar footnote.

```

3889 \newcommand{\Xsethangindent}[1]{%
3890   \if@RTL%
3891     \hangindent=-\csuse{Xhangindent@#1}%
3892     \everypar{\hangindent=-\csuse{Xhangindent@#1}}%
3893   \else%
3894     \hangindent=\csuse{Xhangindent@#1}%
3895     \everypar{\hangindent=\csuse{Xhangindent@#1}}%
3896   \fi%
3897 }%
3898 \newcommand{\sethangindentX}[1]{%
3899   \if@RTL%
3900     \hangindent=-\csuse{hangindentX@#1}%
3901     \everypar{\hangindent=-\csuse{hangindentX@#1}}%
3902   \else%
3903     \hangindent=\csuse{hangindentX@#1}%
3904     \everypar{\hangindent=\csuse{hangindentX@#1}}%
3905   \fi%
3906 }%
3907 %

```

### XIII.9 Critical notes presentation

Here, we define some common macros which are used in order to print a critical note, 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 `\Xlinerrangeseparator`, `\Xendlinerrangeseparator`, `\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.

```

3908 \def\endashchar{\textnormal{--}}
3909
3910 \newcommand*\fullstop{\textnormal{.}}
3911 \def\Xsublinesep@side{\fullstop}
3912
3913 \newcommand*\rbracket{\textnormal{%
3914   \csuse{text}\csuse{footnote@lang}}{%
3915     \ifluatex%
3916       \ifdefstring{\footnote@luatextdir}{TRT}{\thinspace []{\thinspace
3917         \else%
3918         \thinspace]}%
3919       \fi}%
3920   }%
3921 }
3922
3923 %

```

### XIII.9.2 Pstart number in footnote

`\printpstart` The `\printpstart` macro prints the pstart number for a note.

```

3924 \newcommand{\printpstart}[0]{%
3925   \ifboolexpr{bool{l@dpairing} or bool{l@dprintingpages} or bool{
3926     l@dprintingcolumns}}{%
3927     \ifledRcol%
3928       \thepstartR%
3929     \else%
3930       \thepstartL%
3931     \fi%
3932   }%
3933   \thepstart%
3934 }
3935 %

```

### 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).

```

3936 %
3937 \newcommand{\print@lemma}[3]{%
3938   \bgroup%
3939     \nottoggle{Xlemmadisablefontselection@#3}%
3940     {\select@lemmafnt#1|}%
3941     {}%
3942   \bgroup%
3943     \csuse{Xlemmafnt@#3}%Deprecated
3944     \csuse{Xwraplemma@#3}{#2}%
3945   \egroup%
3946 \egroup%
3947 \iftoggle{nosep@}{%
3948   \hskip\csuse{Xinplaceoflemmaseparator@#3}%
3949   \relax%
3950 }%
3951 {\ifcsemt{Xlemmaseparator@#3}%
3952   {%
3953     \hskip\csuse{Xinplaceoflemmaseparator@#3}%
3954     \relax%
3955   }%
3956   {%
3957     \nobreak%
3958     \hskip\csuse{Xbeforelemmaseparator@#3}%
3959     \csuse{Xlemmaseparator@#3}%
3960     \hskip\csuse{Xafterlemmaseparator@#3}%
3961     \relax%
3962   }%
3963 }%
3964 }%
3965 %

```

### 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.

```

3966 \newcommand{\Xstorelineinfo}[2]{%
3967   \l@dp@rsefootspec#1|}%
3968   \iftoggle{Xnumberonlyfirstintwolines@#2}{%
3969     \xdef\lineinfo@{\l@dparsedstartline - \l@dparsedstartsub - \
3970     \l@dparsedendline - \l@dparsedendsub}%
3971     {}%
3972     {\xdef\lineinfo@{\l@dparsedstartline - \l@dparsedstartsub}%
3973     }%
3974 }%
3975 %

```



`\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`.

```

3976 \newcommand{\printlinefootnote}[2]{%
3977   \xdef\@currentseries{#2}%
3978   \iftoggle{nonum@}{%Try if the line number must printed for this specific
not (by default, yes)
3979   \hspace{\csuse{Xinplaceofnumber@#2}}%
3980   }%
3981   {%
3982     {%
3983       \iftoggle{Xnonumber@#2}%Try if the line number must printed (by
default, yes)
3984       {%
3985         \hspace{\csuse{Xinplaceofnumber@#2}}%
3986         }%
3987         {%
3988           {\iftoggle{Xnumberonlyfirstinline@#2}% If for this series the
line number must be printed only in the first time.
3989           {%
3990             \ifcsdef{prevline#2}%
3991             {%Be sure the \prevline exists.
3992             \ifcsequal{prevline#2}{\lineinfo@}%Try it
3993             {%
3994             \ifcsequal{Xsymlinenum@#2}%
3995             {%
3996             \hspace{\csuse{Xinplaceofnumber@#2}}%
3997             }%
3998             {\printsymlinefootnotearea{#2}}%
3999             }%
4000             {%
4001             \printlinefootnotearea{#1}{#2}%
4002             }%
4003             }%
4004             {%
4005             \printlinefootnotearea{#1}{#2}%
4006             }%
4007             }%
4008             {%
4009             \printlinefootnotearea{#1}{#2}%
4010             }%
4011             \csxdef{prevline#2}{\lineinfo@}%
4012             }%
4013             }%
4014             }%
4015             }%
4016   }
4017 %

```

`\printsymlinefootnotearea` This macro prints the space before the line symbol, changes the font, when prints the line symbol and the space after it.

```

4018 \newcommand{\printsymlinefootnotearea}[1]{%
4019   \hspace{\csuse{Xbeforesymlinewidth@#1}}%
4020   \csuse{Xnotenumfont@#1}%
4021   \ifdimequal{\csuse{Xboxsymlinewidth@#1}}{\z@}%
4022     {\csuse{Xsymlinewidth@#1}}%
4023     {\hbox to \csuse{Xboxsymlinewidth@#1}%
4024       {\csuse{Xsymlinewidth@#1}\hfill}}%
4025   }%
4026   \hspace{\csuse{Xaftersymlinewidth@#1}}%
4027 }%
4028 %

```

**\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.)

```

4029 \newcommand{\printlinefootnotearea}[2]{%
4030   \printXbeforenumber{#2}%
4031   \csuse{Xnotenumfont@#2}%
4032   \boxfootnotenumbers{#1}{#2}%
4033   \printXafternumber{#2}%
4034 }%
4035 %

```

**\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.

```

4036 \newcommand{\boxfootnotenumbers}[2]{%
4037   \ifdimequal{\csuse{Xboxlinewidth@#2}}{\Opt}{%
4038     \printlinefootnotenumbers{#1}{#2}%
4039   }%
4040   {%
4041     \hbox to \csuse{Xboxlinewidth@#2}%
4042       {%
4043         \IfSubStr{RC}{\csuse{Xboxlinenumalign@#2}}{\hfill}{}%
4044         \printlinefootnotenumbers{#1}{#2}%
4045         \IfSubStr{LC}{\csuse{Xboxlinenumalign@#2}}{\hfill}{}%
4046       }%
4047   }%
4048 }%
4049 %

```

**\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.

```

4050 \newcommand{\printlinefootnotenumbers}[2]{%
4051   \get@pstartcounter%
4052   \ifboolexpr{%
4053     (togl{Xpstart@#2} and bool{numberpstart})%
4054     or togl{Xpstarteverytime@#2}%
4055     and not (%

```

```

4056     togl{Xpstartonlyfirst@#2}%
4057     and test {\ifnumequal{\csuse{Xprevpstart@#2}}{\current@pstart}}%
4058   )%
4059 }%
4060 {%
4061   \printpstart%
4062   \csuse{Xpstartseparator@#2}%
4063 }{%
4064 \global\cslet{Xprevpstart@#2}{\current@pstart}%
4065 \get@stanzacounter%
4066 \ifboolexpr{%
4067   togl{Xstanza@#2}%
4068   and bool {numberstanza}%
4069   and not (%
4070     togl{Xstanzaonlyfirst@#2}%
4071     and test {\ifnumequal{\csuse{Xprevstanza@#2}}{\current@stanza}}%
4072   )%
4073 }{%
4074   \printstanza%
4075   \csuse{Xstanzaseparator@#2}%
4076 }{%
4077 \global\cslet{Xprevstanza@#2}{\current@stanza}%
4078 \iftoggle{Xonlypstart@#2}{-}{%
4079   \csuse{Xtxtbeforenumber@#2}%
4080   \printlines#1|\ifledRcol@{\@Rlineflag\fi}}%
4081 }%
4082 %

```

`\get@pstartcounter` A single macro to get the current pstart counter.

```

4083 \newcommand{\get@pstartcounter}[0]{%
4084   \ifboolexpr{bool{!@dprintingpages} or bool{!@dprintingcolumns}}%
4085   {%
4086     \ifledRcol%
4087       \global\numdef\current@pstart{\the\c@pstartR}%
4088     \else%
4089       \global\numdef\current@pstart{\the\c@pstartL}%
4090     \fi%
4091   }%
4092   {%
4093     \global\numdef\current@pstart{\the\c@pstart}%
4094   }%
4095 }%
4096 %

```

`\get@stanzacounter` The same to get the current stanza counter.

```

4097 \newcommand{\get@stanzacounter}[0]{%
4098   \ifboolexpr{bool{!@dprintingpages} or bool{!@dprintingcolumns}}%
4099   {%
4100     \ifledRcol%
4101       \global\numdef\current@stanza{\the\c@stanzaR}%
4102     \else%
4103       \global\numdef\current@stanza{\the\c@stanzaL}%

```

```

4104 \fi%
4105 }%
4106 {%
4107 \global\newdef\current@stanza{\the\c@stanza}%
4108 }%
4109 }%
4110 %

```

`\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.)

```

4111 \newcommand{\printXbeforenumber}[1]{%
4112 \hspace{\csuse{Xbeforenumber@#1}}%
4113 }%
4114 %

```

`\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

```

4115 \newcommand{\printXafternumber}[1]{%
4116 \iftoggle{Xnonbreakableafternumber@#1}{\nobreak}{}%
4117 \hspace{\csuse{Xafternumber@#1}}%
4118 }%
4119 %

```

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@pnum20 \newif\ifl@d@pnum
\ifl@d@ssub21 \newif\ifl@d@ssub
\ifl@d@elin22 \newif\ifl@d@elin
\ifl@d@esl23 \newif\ifl@d@esl
\ifl@d@dash24 \newif\ifl@d@dash
\ifl@d@Xtwolines25 \newif\ifl@d@Xtwolines%
\ifl@d@Xmorethantwolines26 \newif\ifl@d@Xmorethantwolines%
\ifl@d@Xlinenumber27 \let\@annot@start@print\relax%
\@annot@start@print28 \let\@annot@end@print\relax%
\@annot@end@print29 \newif\ifl@d@Xlinenumber%
\@annot@end@print30 %

```

```

\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
31 \newcommand*{\l@dp@rsefootsspec}[3]{\l@dp@rsefootsspec#1|}
32 \def\l@dp@rsefootsspec#1|#2|#3|#4|#5|#6|#7|{%
33 \gdef\l@dp@rsefootsspecstartpage{#1}%
34 \gdef\l@dp@rsefootsspecstartline{#2}%
35 \gdef\l@dp@rsefootsspecstartsub{#3}%
36 \gdef\l@dp@rsefootsspecendpage{#4}%
37 \gdef\l@dp@rsefootsspecendline{#5}%
38 \gdef\l@dp@rsefootsspecendsub{#6}%
39 }
40 %

```

Initialise the several number value macros.

```

41 \def\l@dp@rsefootsspecstartpage{0}%
42 \def\l@dp@rsefootsspecstartline{0}%
43 \def\l@dp@rsefootsspecstartsub{0}%
44 \def\l@dp@rsefootsspecendpage{0}%
45 \def\l@dp@rsefootsspecendline{0}%
46 \def\l@dp@rsefootsspecendsub{0}%
47
48 %

```

`\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`.

```

49 \newcommand*{\setprintlines}[6]{%
50 \let\@annot@start@print\relax%
51 \let\@annot@end@print\relax%
52 \l@d@pnumfalse%
53 \l@d@dashfalse%
54 \l@d@elinfalse%
55 %

```

Should we print the line number? It depends on hooks and annotations.

```

4156 \l@d@Xlinenumbertrue%
4157 \ifboolexpr{togl{Xnolinenumber@\@currentseries}%
4158   or (%
4159     togl{Xnolinenumberifannotation@\@currentseries}%
4160     and not test{\ifdefvoid{\annot@start}}}%
4161     and not test{\ifdefvoid{\annot@end}}}%
4162   )%
4163 }{%
4164   \l@d@Xlinenumberfalse%
4165 }{%
4166 %

```

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.

```

4167 \ifbypage@
4168   \ifnum#4=#1 \else
4169     \l@d@pnumtrue
4170     \l@d@dashtrue
4171   \fi
4172 \fi
4173 %

```

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.

```

4174 \ifboolexpr{%
4175   bool{\l@d@pnum}%
4176   or (%
4177     not test{\ifnumequal{#2}{#5}}%
4178     and bool{\l@d@Xlinenumber}%
4179   )%
4180 }{%
4181   \l@d@elintrue%
4182   \l@d@dashtrue%
4183   \unless\ifx\relax\annot@end%
4184     \def\@annot@end@print{%
4185       \l@wrappcs@ifnotemptybox{Xwraplinenumannotation@\@currentseries}{\
annot@end}%
4186     }%
4187   \fi%
4188 }{%
4189 %

```

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.

```

4190 \ifl@d@elin%
4191   \def\@annot@start@print{%

```

```

4192 \l@wrapcs@ifnotemptybox{Xwraplinenumannotation@\@currentseries}{\
annot@start}%
4193 }%
4194 \else%
4195 \ifx\annot@start\annot@end%
4196 \unless\ifx\@annot@start\relax%
4197 \def\@annot@start@print{%
4198 \l@wrapcs@ifnotemptybox{Xwraplinenumannotation@\@currentseries}{\
annot@start}%
4199 }%
4200 \fi%
4201 \else%
4202 \ifx\@annot@end@print\relax%
4203 \def\@annot@start@print{%
4204 \l@wrapcs@ifnotemptybox{Xwraplinenumannotation@\@currentseries}{%
4205 \ifx\annot@start\empty\else%
4206 \annot@start%
4207 \ifdefined\linangesep@%
4208 \linangesep@%
4209 \else%
4210 \csuse{Xlinangeseparator@\@currentseries}%
4211 \fi%
4212 \fi%
4213 \annot@end%
4214 }%
4215 }%
4216 \else%
4217 \let\@annot@start@print\@annot@end@print%
4218 \let\@annot@end@print\relax%
4219 \fi%
4220 \fi%
4221 \fi%
4222 %

```

We print the starting sub-line if it is nonzero.

```

4223 \l@d@ssubfalse
4224 \ifnum#3=0 \else
4225 \l@d@ssubtrue
4226 \fi
4227 %

```

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.

```

4228 \l@d@eslfalse
4229 \ifnum#6=0 \else
4230 \ifnum#6=#3
4231 \ifl@d@elin \l@d@esltrue \else \l@d@eslfalse \fi
4232 \else
4233 \l@d@esltrue
4234 \l@d@dashtrue
4235 \fi
4236 \fi%
4237 %

```

However, if the `\Xtwolines` is set for the current series, we do not print the last line number.

```

4238 \ifl@dash%
4239 \ifboolexpr{togl{fulllines@} or test{\ifcsempy{Xtwolines@
@currentseries}}}%
4240 {}%
4241 {%
4242 \setistwofollowinglines{#1}{#2}{#4}{#5}%
4243 \ifboolexpr{%
4244 (%
4245 togl {Xtwolinesbutnotmore@ \@currentseries}%
4246 and not%
4247 (%
4248 bool {istwofollowinglines@}%
4249 )%
4250 )%
4251 or%
4252 (%
4253 (not test{\ifnumequal{#1}{#4}})%
4254 and togl{Xtwolinesonlyinsamepage@ \@currentseries}%
4255 )%
4256 }%
4257 {}%
4258 {%
4259 \l@dashfalse%
4260 \l@Xtwolinestrue%
4261 \l@elinfalse%
4262 \l@eslfalse%
4263 \ifcsempy{Xmorethantwolines@ \@currentseries}%
4264 {}%
4265 {\ifistwofollowinglines@ \else%
4266 \l@Xmorethantwolinestrue%
4267 \fi%
4268 }%
4269 }%
4270 }%
4271 \fi%
4272 %

```

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

```

4273 \iftoggle{Xnoidenticallinenumannotation@ \@currentseries}{%
4274 \ifx\annot@start\annot@end%
4275 \let\@annot@end@print\relax%
4276 \ifboolexpr{%Do not print the dash if we do not print the line number
4277 test {\ifdefempty{\linenumrep}}%
4278 or not bool {l@d@Xlinenum}}%
4279 }{%
4280 \l@dashfalse%
4281 }{}%
4282 \fi%
4283 }{}%
4284 %

```



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.

```

4285 \iftoggle{Xlinenumannotationonlyfirst@\@currentseries}{%
4286   \ifboolexpr{%
4287     (%
4288       togl{Xlinenumannotationonlyfirstintwo@\@currentseries}%
4289       and test{\ifcsequal{annot@start}{prevannot@start@\@currentseries}}%
4290       and test{\ifcsequal{annot@end}{prevannot@end@\@currentseries}}%
4291     )%
4292     or%
4293     (%
4294       not togl{Xlinenumannotationonlyfirstintwo@\@currentseries}%
4295       test{\ifcsequal{annot@start}{prevannot@start@\@currentseries}}%
4296     )%
4297   }{%
4298     \def\@annot@start@print{%
4299       \l@wrapcs@ifnotemptybox{Xwraplinenumannotation@\@currentseries}{\
csuse{Xsymlinenumannotation@\@currentseries}}%
4300     }%
4301     \let\@annot@end@print\relax%
4302     \ifboolexpr{%Do not print the dash if we do not print the line number
4303       test {\ifdefempty{\linenumrep}}%
4304       or not bool {l@d@Xlinenum}}%
4305     }{%
4306       \l@d@dashedfalse%
4307     }{%
4308   }{%
4309     \global\cslet{prevannot@start@\@currentseries}{\annot@start}%
4310     \global\cslet{prevannot@end@\@currentseries}{\annot@end}%
4311   }%
4312   {}%
4313 %

```

End of `\setprintlines`.

```

4314 }%
4315 %

```

`\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.

```

4316 \newif\ifistwofollowinglines@%
4317 \newcommand{\setistwofollowinglines}[4]{%
4318   \ifcsdef{lastlinenumberon@#1}%
4319     {\numdef{\tmp}{\csuse{lastlinenumberon@#1}}}%
4320     {\numdef{\tmp}{0}}%
4321   \istwofollowinglines@false%
4322   \ifnumequal{#4-#2}{1}%
4323     {\istwofollowinglines@true}%%
4324     {\ifbypage@%
4325       \ifnumequal{#3-#1}{1}%
4326         {%
4327           \ifnumequal{#2}{\tmp}%
4328             {\ifnumequal{#4}{1}{\istwofollowinglines@true}{}}%
4329             {}%
4330           }%
4331         }%
4332       \fi%
4333     }%
4334 }%
4335 %

```

`\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

```

4336 \newcount\printlines@count%
4337 \def\printlines#1|#2|#3|#4|#5|#6|#7|#8|{%
4338   \global\advance\printlines@count by 1%
4339   \begingroup%
4340 %

```

Decide which part of line number components we will print.

```

4341   \setprintlines{#1}{#2}{#3}{#4}{#5}{#6}%
4342 %

```

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.

```

4343   \ifdimequal{\csuse{Xboxstartlinenum@\@currentseries}}{0pt}%
4344     {\bgroup}%
4345     {\leavevmode\hbox to \csuse{Xboxstartlinenum@\@currentseries}\bgroup\
4346 hfill}%
4347   \ifcsstring{Xlinenumannotationposition@\@currentseries}{before}%
4348     {\@annot@start@print}%
4349     {}%
4349 %

```

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.

```

4350 \ifl@d@pnum%
4351 \iftoggle{Xnopagenumberifcurrent@\@currentseries}{%
4352 \edlabel{printlines@\the\printlines@count}%
4353 }{ }%
4354 \ifboolexpr{%
4355 not test{\ifnumequal{\xpageref{printlines@\the\printlines@count
}}{#1}}%
4356 or not togl{Xnopagenumberifcurrent@\@currentseries}%
4357 }%
4358 {%
4359 \csuse{Xbeforepagenumber@\@currentseries}%
4360 \wrap@edcrossref{\@this@crossref@start}{%
4361 #1%
4362 }%
4363 \ifl@d@Xlinenumber%
4364 \csuse{Xpagelinesep@\@currentseries}%
4365 \fi%
4366 }{ }%
4367 \fi%
4368 \ifl@d@Xlinenumber%
4369 \wrap@edcrossref{\@this@crossref@start}{%
4370 \linenumrep{#2}%
4371 \iftoggle{Xlineflag@\@currentseries}{#8}{ }%
4372 }%
4373 \iftoggle{Xnotboxingsubline@\@currentseries}{\egroup}{ }%Close startline
box
4374 \ifl@d@ssub%
4375 \csuse{Xsublinesep@\@currentseries}%
4376 \wrap@edcrossref{\@this@crossref@start}{\sublinenumrep{#3}}%
4377 \fi%
4378 \fi%
4379 \ifcsstring{Xlinenumannotationposition@\@currentseries}{after}%
4380 {\@cannot@start@print}%
4381 }{ }%
4382 \iftoggle{Xnotboxingsubline@\@currentseries}{\egroup}{ }%Close startline
box
4383 %

```

Then print the dash + end line number, or the range symbol.

```

4384 \ifdimequal{\csuse{Xboxendlinenumber@\@currentseries}}{0pt}%
4385 {\bgroup}%
4386 {\hbox to \csuse{Xboxendlinenumber@\@currentseries}\bgroup}%
4387 \ifl@d@Xtwolines%
4388 \ifl@d@Xlinenumber%
4389 \ifl@d@Xmorethantwolines%
4390 \wrap@edcrossref{\@this@crossref@end}{\csuse{Xmorethantwolines@\@currentseries}}%
4391 \else%
4392 \wrap@edcrossref{\@this@crossref@end}{\csuse{Xtwolines@\@currentseries}}%

```

```

4393   \fi%
4394   \fi%
4395 \else%
4396   \ifl@d@dash%
4397   \ifdefined\linangesep%
4398     \linangesep%
4399   \else%
4400     \csuse{Xlinangeseparator@\@currentseries}%
4401   \fi%
4402 \fi%
4403 \ifcsstring{Xlinenumannotationposition@\@currentseries}{before}%
4404   {\@annot@end@print}%
4405   {}%
4406 \ifl@d@pnum%
4407   \ifboolexpr{%
4408     not test{\ifnumequal{\xpageref{printlines@\the\printlines@count
}}{#4}}%
4409     or not togl{Xnopagenumberifcurrent@\@currentseries}%
4410     }{%
4411       \csuse{Xbeforepagenumber@\@currentseries}%
4412       \wrap@edcrossref{\@this@crossref@end}{%
4413         #4%
4414       }%
4415       \csuse{Xpagelinesep@\@currentseries}%
4416     }{}%
4417 \fi%
4418 \ifl@d@Xlinenum%
4419   \ifl@d@elin%
4420     \wrap@edcrossref{\@this@crossref@end}{%
4421       \linenumrep{#5}%
4422       \iftoggle{Xlineflag@\@currentseries}{#8}{}%
4423     }%
4424   \fi%
4425   \ifl@d@esl%
4426     \ifl@d@elin%
4427       \csuse{Xsublinesep@\@currentseries}%
4428     \fi%
4429     \wrap@edcrossref{\@this@crossref@end}{\sublinenumrep{#6}}%
4430   \fi%
4431 \fi%
4432 \ifcsstring{Xlinenumannotationposition@\@currentseries}{after}%
4433   {\@annot@end@print}%
4434   {}%
4435 \fi%
4436 \ifdimequal{\csuse{Xboxendlinum@\@currentseries}}{0pt}%
4437   {}%
4438   {\hfill}%Prevent underfull hbox
4439 \egroup%
4440 \endgroup%
4441 }%
4442 %

```

## 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

```

4443 \newcommand{\prepare@Xgroupbyline}[3]{%
4444   \iftoggle{Xgroupbylineseparetwolines@#1}{%
4445     \l@dparsespec#2%
4446     \ifcsdef{#1@forinserting@l@dparsedendpage-l@dparsedendline-l@dparsedendsub}%
4447       {%
4448         \csgappto{#1@forinserting@l@dparsedendpage-l@dparsedendline-l@dparsedendsub}{%
4449           \hskip\csuse{Xafternote@#1}\relax%
4450         }%
4451       }%
4452     }%
4453     \add@hooktoggle@specific@to@cs{#1@forinserting@l@dparsedendpage-l@dparsedendline-l@dparsedendsub}%
4454     \add@hookarg@specific@to@cs{#1@forinserting@l@dparsedendpage-l@dparsedendline-l@dparsedendsub}%
4455     \csxappto%
4456       {#1@forinserting@l@dparsedendpage-l@dparsedendline-l@dparsedendsub}%
4457     {%
4458       \keep@this@crossref@forinserting%
4459       \unexpanded{%
4460         \ifcempty{Xsymlinenum@#1}%
4461           {\csuse{Xparafootsep@#1}}%
4462         }%
4463         #3{#1}{#2}%
4464       }%
4465     }%
4466     \listcsxadd{#1@forinserting}{l@dparsedendpage-l@dparsedendline-l@dparsedendsub}%
4467   }{%
4468     \ifcsdef{#1@forinserting@all}{%
4469       \csgappto%
4470         {#1@forinserting@all}%
4471         {\hskip\csuse{Xafternote@#1}\relax}%
4472       }{%
4473         \add@hooktoggle@specific@to@cs{#1@forinserting@all}%
4474         \add@hookarg@specific@to@cs{#1@forinserting@all}%
4475         \csxappto%
4476           {#1@forinserting@all}%
4477         {%
4478           \keep@this@crossref@forinserting%
4479           \unexpanded{%
4480             \ifcempty{Xsymlinenum@#1}%
4481               {\csuse{Xparafootsep@#1}}%
4482             }%
4483             #3{#1}{#2}%

```

```

4484 }%
4485 }%
4486 }%
4487 \listcsgadd{#1@forinserting}{all}%
4488 }%
4489 %

```

`\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.

```

4490 \newcommand{\keep@this@crossref@forinserting}{%
4491   \unexpanded{\gdef\@this@crossref@start}{\@this@crossref@start}%
4492   \unexpanded{\gdef\@this@crossref@end}{\@this@crossref@end}%
4493 }%
4494 %

```

## 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`.

```

4495 \providecommand*\multiplefootnotemarker}{3sp}
4496 \providecommand*\multfootsep}{\textsuperscript{\normalfont,}}
4497
4498 %

```

`\m@mmf@prepare` `\m@mmf@prepare` A pair of self-cancelling kerns. This may have been defined in the memoir class.

```

4499 \providecommand*\m@mmf@prepare}{%
4500   \kern-\multiplefootnotemarker
4501   \kern\multiplefootnotemarker\relax}
4502 %

```

`\m@mmf@check` `\m@mmf@check` This may have been defined in the memoir class. If it recognises the last kern as `\multiplefootnotemarker` it typesets `\multfootsep`.

```

4503 \providecommand*\m@mmf@check}{%
4504   \ifdim\lastkern=\multiplefootnotemark\relax
4505     \edef\@x@sf{\the\spacefactor}%
4506     \unkern
4507     \multfootsep
4508     \spacefactor\@x@sf\relax
4509   \fi}
4510
4511 %

```

We have to modify `\@footnotetext` and `\@footnotemark`. However, if `memoir` is used the modifications have already been made.

```

4512 \@ifclassloaded{memoir}{}{%
4513 %

```

`\@footnotetext` Add `\m@mmf@prepare` at the end of `\@footnotetext`.

```

4514 \apptocmd{\@footnotetext}{\m@mmf@prepare}{}{}
4515 %

```

`\@footnotemark` Modify `\@footnotemark` to cater for adjacent footnotes.

```

4516
4517 \patchcmd{\@footnotemark}
4518   {\nobreak}
4519   {\m@mmf@check
4520    \nobreak
4521   }
4522   {}{}
4523 \patchcmd{\@footnotemark}
4524   {\@makefnmark}
4525   {\@makefnmark
4526    \m@mmf@prepare
4527   }
4528   {}{}
4529 %

```

Finished the modifications for the non-memoir case.

```

4530 }
4531
4532 %

```

## 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.

```

4533 \pretocmd{\@footnotetext}{%
4534   \ifnumberedpar@
4535     \edtext{}{}%

```

```

4536     \advance \edtext@notes by 1%
4537     \l@dbfnote{#1}%
4538   }%
4539   \else
4540   {}{}
4541 \apptocmd{\@footnotetext}{\fi}{}{}%
4542 %

```

`\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
4543
4544 \patchcmd%
4545   {\footnote}%
4546   {\stepcounter\@mpfn}%
4547   {%
4548   \ifboolexpr{bool{l@dpairing} or bool{l@dprintingpages} or bool{
4549   l@dprintingcolumns}}{%
4550     \global\advance\footnote@reading by \@ne%
4551     \get@thisfootnote%
4552     \get@fnmark{\thisc@footnote}%
4553     \ifcsdef{footnotereading\the\footnote@reading=typeset}%
4554       {\setcounter{\@mpfn}{\csuse{footnotereading\the\footnote@reading=
4555       typeset}}}%
4556       {\setcounter{\@mpfn}{\footnote@reading}}}%
4557   }-%
4558   \stepcounter\@mpfn%
4559   }%
4560   {}
4561   {}
4562 \newcommand{\get@thisfootnote}{%
4563   \ifboolexpr{bool{l@dpairing} or bool{l@dprintingpages} or bool{
4564   l@dprintingcolumns}}{%
4565     \protected@xdef\thisc@footnote{\the\footnote@reading}%
4566     }-%
4567     \protected@xdef\thisc@footnote{\the\c@footnote}%
4568   }%
4569
4570 \newcommand{\l@dbfnote}[1]{%
4571   \get@thisfootnote%
4572   \gdef\@tag{#1\relax}%
4573   \ifledRcol%
4574     \xright@appenditem{%
4575       \ifdefined\Hy@footnote@currentHref%
4576       \noexpand\def\noexpand\Hy@footnote@currentHref{\
4577       Hy@footnote@currentHref}%
4578       \fi%
4579       \noexpand\vl@dbfnote{{\expandonce\@tag}}{\thisc@footnote}%

```



```

4580         \to\inserts@listR
4581         \global\advance\insert@countR \@ne%
4582     \else%
4583         \xright@appenditem{%
4584             \ifdefined\Hy@footnote@currentHref%
4585                 \noexpand\def\noexpand\Hy@footnote@currentHref{\
Hy@footnote@currentHref}%
4586             \fi%
4587             \noexpand\vl@dbfnote{\expandonce\@tag}{\thisc@footnote}%
4588         }%
4589         \to\inserts@list
4590         \global\advance\insert@count \@ne%
4591     \fi
4592     \ignorespaces%
4593 }%
4594
4595 \newcommand{\get@fnmark}[1]{%
4596     \ifboolexpr{bool{!@dpairing} or bool{!@dprintingpages} or bool{
!@dprintingcolumns}}%
4597     {%
4598         \stepcounter{footnote@typeset}%
4599         \setcounter{footnote}{\c@footnote@typeset}%
4600         \immediate\write\@mainaux{%
4601             \csgdef{footnotereading#1=typeset}{\the\c@footnote@typeset}%
4602         }%
4603         \def\@thefnmark{\thefootnote}%
4604     }%
4605     {%
4606         \setcounter{footnote}{#1}%
4607         \def\@thefnmark{\thefootnote}%
4608     }%
4609 }%
4610
4611 \newcommand{\vl@dbfnote}[2]{%
4612     \get@fnmark{#2}%
4613     \@footnotetext{#1}%
4614 }%
4615 %

```

### XIV.3 Footnote formats

Some of the code for the various formats is remarkably similar to that in section ??.

The following macros generally set things up for the “standard” footnote format.

`\prebodyfootmark` Two convenience macros for use by `\...@footnotemark...` macros.  
`\postbodyfootmark`

```

4616 \newcommand*\prebodyfootmark{%
4617     \leavevmode
4618     \ifhmode
4619         \edef\@x@sf{\the\spacefactor}%
4620         \m@mmf@check
4621         \nobreak
4622     \fi}
4623 \newcommand*\postbodyfootmark{%

```

```

4624 \m@mmf@prepare
4625 \ifhmode\spacefactor\@x@sf\fi\relax}
4626
4627 %

```

## XIV.4 Footnote arrangement

### XIV.4.1 User level macro

`\arrangementX` `\arrangementX[⟨s⟩]{⟨arrangement⟩}` command calls, for each series, a specific command which set many counters and commands in order to define specific arrangement.

```

4628 \newcommandx{\arrangementX}[2][1,usedefault]{%
4629   \def\do##1{%
4630     \csname arrangementX@#2\endcsname{##1}%
4631   }%
4632   \ifstrempy{#1}%
4633     {%
4634     \dolistloop{\@series}%
4635     }%
4636     {
4637     \docsvlist{#1}%
4638     }%
4639 }%
4640 %

```

### XIV.4.2 Normal footnotes

`\normal@footnotemarkX` `\normal@footnotemarkX{⟨series⟩}` sets up the typesetting of the marker at the point where the footnote is called for.

```

4641 \newcommand*{\normal@footnotemarkX}[1]{%
4642   \prebodyfootmark
4643   \wrapped@bodyfootmarkX{#1}%
4644   \postbodyfootmark}
4645
4646 %

```

`\normalbodyfootmarkX` The `\normalbodyfootmarkX{⟨series⟩}` *really* typesets the in-text marker. The style is the normal superscript.

```

4647 \newcommand*{\normalbodyfootmarkX}[1]{%
4648   \hbox{\textsuperscript{\normalfont\@nameuse{@thefnmark#1}}}}
4649 %

```

`\normalvfootnoteX` `\normalvfootnoteX{⟨series⟩}{⟨text⟩}` does the `\insert` for the `⟨series⟩` and calls the series' `\footfmt...` to format the `⟨text⟩`.

```

4650 \notbool{parapparatus@}{\newcommand*{\newcommand}{\normalvfootnoteX}[2]{%
4651   \beforeinsertion@X{#1}%
4652   \insert\@nameuse{footins#1}\bgroup
4653     \fontseries{\seriesdefault}%
4654     \fontshape{\shapedefault}%
4655     \hsize=\expandafter\dimexpr\csuse{widthX@#1}\relax%

```

```

4656 \noindent\csuse{bhooknoteX@#1}%
4657 \csuse{notefontsizeX@#1}%
4658 \footsplitskips
4659 \ifl@dpairing\ifl@dpaging\else%
4660 \setnoteswidthliketwocolumnsX@{#1}%
4661 \fi\fi%
4662 \setnotesXpositionliketwocolumns@{#1}%
4663 \spaceskip=\z@skip \xspaceskip=\z@skip
4664 \csuse{\csuse{footnote@dir}}\@nameuse{footfmt#1}{#1}{#2}\egroup}
4665
4666 %

```

`\mpnormalvfootnoteX` The minipage version.

```

4667 \newcommand*\mpnormalvfootnoteX[3]{%
4668 \get@thisfootnoteX{#1}%
4669 \get@fnmarkX{#1}{\thisc@footnote}{#3}%
4670 \ifstrempy{#3}{%
4671 \edef\this@footnoteX@reading{\the\csname footnote#1@reading\endcsname}%
4672 }{%
4673 \edef\this@footnoteX@reading{###3}%
4674 }%
4675 \global\setbox\@nameuse{mpfootins#1}\vbox{%
4676 \unvbox\@nameuse{mpfootins#1}
4677 \noindent\csuse{bhooknoteX@#1}%
4678 \csuse{notefontsizeX@#1}%
4679 \hsize\columnwidth
4680 \@parboxrestore
4681 \color@begingroup
4682 \@nameuse{footfmt#1}{#1}{#2}\color@endgroup}}
4683
4684 %

```

`\normalfootfmtX` `\normalfootfmtX{<series>}{<text>}` typesets the footnote text, prepended by the marker.

```

4685 \notbool{parapparatus@}{\newcommand*{\newcommand}{\normalfootfmtX}[2]{%
4686 \ifluatex%
4687 \textdir\footnote@luatextextdir%
4688 \pardir\footnote@luatexpardir%
4689 \fi%
4690 \protected@edef\@currentlabel{%
4691 \@nameuse{@thefnmark#1}%
4692 }%
4693 \ledsetnormalparstuffX{#1}%
4694 \sethangindentX{#1}%
4695 \rule\z@\splittopskip%
4696 {{\csuse{notenumfontX@#1}\wrapped@footfootmarkX{#1}}%
4697 \csuse{wrapcontentX@#1}{#2}%
4698 \strut\par}}
4699
4700 %

```

`\normalfootfootmarkX` `\normalfootfootmarkX{<series>}` is called by `\normalfootfmtX` to typeset the footnote marker in the footer before the footnote text.

```

4701 \newcommand*{\normalfootfootmarkX}[1]{%
4702   \textsuperscript{@nameuse{@thefnmark#1}}
4703 }
4704 %

```

`\normalfootstartX` `\normalfootstartX{<series>}` is the `<series>` footnote starting macro used in the output routine.

```

4705 \newcommand*{\normalfootstartX}[1]{%
4706   \ifdimequal{Opt}{\prenotesX@}{%
4707     {%
4708       \iftoggle{prenotesX@}{%
4709         \togglefalse{prenotesX@}%
4710         \skip\csname footins#1\endcsname=%
4711         \glueexpr\csuse{prenotesX@}+\csuse{afterruleX@#1}\relax%
4712       }%
4713     }%
4714   }%
4715   \vskip\skip\csname footins#1\endcsname%
4716   \leftskip=\z@
4717   \rightskip=\z@
4718   \ifl@dpairing\else%
4719     \hsize=\old@hsize%
4720   \fi%
4721   \setnoteswidthliketwocolumnsX@{#1}%
4722   \setnotesXpositionliketwocolumns@{#1}%
4723   \print@footnoteXrule{#1}%
4724 }%
4725 %
4726 %

```

`\normalfootnoteruleX` The rule drawn before the footnote series group.

```

4727 \let\normalfootnoteruleX=\footnoterule
4728 %
4729 %

```

`\normalfootgroupX` `\normalfootgroupX{<series>}` sends the contents of the `<series>` insert box to the output page without alteration.

```

4730 \newcommand*{\normalfootgroupX}[1]{%
4731   \csuse{bhookgroupX@#1}%
4732   \unvbox\@nameuse{footins#1}%
4733   \hsize=\old@hsize%
4734 }%
4735 %
4736 %

```

`\mpnormalfootgroupX` The minipage version.

```

4737 \newcommand*{\mpnormalfootgroupX}[1]{%
4738   \vskip\skip\@nameuse{mpfootins#1}
4739   \ifl@dpairing\ifparledgroup%
4740     \leavevmode\marks\parledgroup@{begin}%

```

```

4741 \marks\parledgroup@series{#1}%
4742 \marks\parledgroup@type{footnoteX}%
4743 \fi\fi\normalcolor
4744 \ifparledgroup%
4745 \ifl@dpairing%
4746 \else%
4747 \setnoteswidthliketwocolumnsX@{#1}%
4748 \setnotesXpositionliketwocolumns@{#1}%
4749 \print@footnoteXrule{#1}%
4750 \fi%
4751 \else%
4752 \setnoteswidthliketwocolumnsX@{#1}%
4753 \setnotesXpositionliketwocolumns@{#1}%
4754 \print@footnoteXrule{#1}%
4755 \fi%
4756 \csuse{bhookgroupX@#1}%
4757 \unvbox\@nameuse{mpfootins#1}}
4758
4759 %

```

`\normalbfnoteX` #1 = footnote series, #2 = footnote content, #3 manual footnote number

```

4760
4761 \newcommand{\normalbfnoteX}[3]{%
4762 \get@thisfootnoteX{#1}%
4763 \ifledRcol%
4764 \ifluatex
4765 \footnotelang@lua[R]%
4766 \fi
4767 \@ifundefined{xpg@main@language}%if polyglossia
4768 {}%
4769 {\footnotelang@poly[R]}%
4770 \xright@appenditem{%
4771 \noexpand\led@set@index@fornote{#1}%
4772 \noexpand\prepare@edindex@fornote{\led@nums}%
4773 \unexpanded{\gdef\@currentnotetype{familiar}}%
4774 \unexpanded{\def\this@footnoteX@reading}{\the\csname footnote#1
@reading\endcsname}%
4775 \noexpand\vbfnoteX{#1}{#2}{\thisc@footnote}{#3}%
4776 \noexpand\led@reinit@index@fornote%
4777 \unexpanded{\advance\@edindex@fornote@\m@ne}%
4778 }%
4779 \to\inserts@listR
4780 \global\advance\insert@countR \@ne%
4781 \else%
4782 \ifluatex
4783 \footnotelang@lua%
4784 \fi
4785 \@ifundefined{xpg@main@language}%if polyglossia
4786 {}%
4787 {\footnotelang@poly}%
4788 \xright@appenditem{%
4789 \noexpand\led@set@index@fornote{#1}%
4790 \noexpand\prepare@edindex@fornote{\led@nums}%

```

```

4791     \unexpanded{\gdef\@currentnotetype{familiar}}%
4792     \unexpanded{\def\this@footnoteX@reading{\the\csname footnote#1
@reading\endcsname}}%
4793     \noexpand\vbfnoteX{#1}{#2}{\this@footnote}{#3}%
4794     \noexpand\led@reinit@index@fornote%
4795     \unexpanded{\advance\@edindex@fornote@\m@ne}%
4796     }%
4797         \to\inserts@list
4798     \global\advance\insert@count \@ne%
4799     \fi
4800 \ignorespaces}
4801
4802 %

```

`\get@thisfootnoteX` The macro `\get@thisfootnote` command just saves the footnote number in the `\thisfootnote` macro, depending on the use of pairing environments.

```

4803 \newcommand{\get@thisfootnoteX}[1]{%
4804     \ifboolexpr{bool{l@dpairing} or bool{l@dprintingpages} or bool{
l@dprintingcolumns}}{%
4805     \protected@xdef\this@footnote{\the\csname footnote#1@reading\
endcsname}}%
4806     }{%
4807     \protected@xdef\this@footnote{\the\csname c@footnote#1\endcsname}}%
4808     }%
4809 }%
4810 %

```

`\vbfnoteX` This command calls the correct footnote-inserting commands. #1 footnote series, #2 footnote content, #3 footnote counter, #4 manual footnote mark

```

4811 \newcommand{\vbfnoteX}[4]{%
4812     \get@fnmarkX{#1}{#3}{#4}\relax%
4813     \@nameuse{regvfootnote#1}{#1}{#2}%
4814 }%
4815 %

```

`\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.

```

4816 \newcommand{\get@fnmarkX}[3]{%
4817     \ifstrempy{#3}{%
4818     \ifboolexpr{bool{l@dpairing} or bool{l@dprintingpages} or bool{
l@dprintingcolumns}}{%
4819     {%
4820     \stepcounter{footnote#1@typeset}%
4821     \setcounter{footnote#1}{\value{footnote#1@typeset}}%
4822     \immediate\write\@mainaux{%
4823     \csgdef{footnote#1reading#2=typeset}{\the\csname c@footnote#1
@typeset\endcsname}}%
4824     }%
4825     }%
4826     }%

```

```

4827     \setcounter{footnote#1}{#2}%
4828   }%
4829   \csdef{@thefnmark#1}{\csuse{thefootnote#1}}%
4830   \csxdef{hr@thefnmark#1}{\the\value{footnote#1}@the\csname footnote#1
@series\endcsname}%
4831 }%
4832 {%
4833   \csdef{@thefnmark#1}{#3}%
4834   \csxdef{hr@thefnmark#1}{#3@the\csname footnote#1@series\endcsname}%
4835 }%
4836 }%
4837 %
4838 %

```

`\vnumfootnoteX` #1 footnote series, #2 footnote content, #3 manual footnote mark

```

4839
4840 \newcommand{\vnumfootnoteX}[3]{%
4841   \ifnumberedpar@
4842     \edtext{}{%
4843       \advance\edtext@notes by 1%
4844       \normalbfnoteX{#1}{#2}{#3}%
4845     }%
4846   \else
4847     \def\this@footnoteX@reading{\the\csname footnote#1@reading\endcsname}%
4848     \get@thisfootnoteX{#1}%
4849     \get@fnmarkX{#1}{\expandonce\thisc@footnote}{#3}%
4850     \@nameuse{regvfootnote#1}{#1}{#2}%
4851   \fi}
4852 %

```

`arrangementX@normal` `\arrangementX@normal{<series>}` initialises the settings for the `<series>` footnotes. This should always be called for each series.

```

4853 \newcommand*{\arrangementX@normal}[1]{%
4854   \csgdef{series@displayX#1}{normal}
4855   \expandafter\let\csname footstart#1\endcsname=\normalfootstartX
4856   \@namedef{@footnotemark#1}{\normal@footnotemarkX{#1}}
4857   \@namedef{bodyfootmark#1}{\normalbodyfootmarkX{#1}}
4858   \expandafter\let\csname regvfootnote#1\endcsname=\normalvfootnoteX
4859   \expandafter\let\csname vfootnote#1\endcsname=\vnumfootnoteX
4860   \expandafter\let\csname footfmt#1\endcsname=\normalfootfmtX
4861   \@namedef{footfootmark#1}{\normalfootfootmarkX{#1}}
4862   \expandafter\let\csname footgroup#1\endcsname=\normalfootgroupX
4863   \expandafter\let\csname footnoterule#1\endcsname=\normalfootnoteruleX
4864   \count\csname footins#1\endcsname=1000
4865   \csxdef{default@footins#1}{1000}%Use to have note only for one side
4866   \dimen\csname footins#1\endcsname=\csuse{maxhnotesX@#1}
4867   \skip\csname footins#1\endcsname=\csuse{beforenotesX@#1}%
4868   \advance\skip\csname footins#1\endcsname by\csuse{afterruleX@#1}%
4869 %

```

Additions for minipages.

```

4870   \ifnoledgroup@ \else%

```

```

4871 \expandafter\let\csname mpvfootnote#1\endcsname=\mpnormalvfootnoteX
4872 \expandafter\let\csname mpfootgroup#1\endcsname=\mpnormalfootgroupX
4873 \count\csname mpfootins#1\endcsname=1000
4874 \dimen\csname mpfootins#1\endcsname=\csuse{maxhnotesX@#1}
4875 \skip\csname mpfootins#1\endcsname=\csuse{beforenotesX@#1}%
4876 \advance\skip\csname mpfootins#1\endcsname by\csuse{afterruleX@#1}%
4877 \fi
4878 }
4879
4880 %

```

#### 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 4881 \newcommand*{\arrangementX@twocol}[1]{%
4882 \csgdef{series@displayX#1}{twocol}
4883 \expandafter\let\csname regvfootnote#1\endcsname=\twocolvfootnoteX
4884 \expandafter\let\csname footfmt#1\endcsname=\twocolfootfmtX
4885 \expandafter\let\csname footgroup#1\endcsname=\twocolfootgroupX
4886 \dimen\csname footins#1\endcsname=\csuse{maxhnotesX@#1}%
4887 \skip\csname footins#1\endcsname=\csuse{beforenotesX@#1}%
4888 \advance\skip\csname footins#1\endcsname by \csuse{afterruleX@#1}\relax%
4889 \twocolfootsetupX{#1}
4890 \ifnoledgroup@else%
4891 \expandafter\let\csname mpvfootnote#1\endcsname=\mpnormalvfootnoteX
4892 \expandafter\let\csname mpfootgroup#1\endcsname=\mptwocolfootgroupX
4893 \skip\csname mpfootins#1\endcsname=\csuse{beforenotesX@#1}%
4894 \advance\skip\csname mpfootins#1\endcsname by\csuse{afterruleX@#1}
4895 \mptwocolfootsetupX{#1}
4896 \fi%
4897 }
4898
4899 %

```

```

\twocolfootsetupX \twocolfootsetupX{<series>}
\mptwocolfootsetupX
4900 \newcommand*{\twocolfootsetupX}[1]{%
4901 \count\csname footins#1\endcsname 500
4902 \csxdef{default@footins#1}{500}%Use this to confine the notes to one
side only
4903 \multiply\dimen\csname footins#1\endcsname by \tw@}
4904 \newcommand*{\mptwocolfootsetupX}[1]{%
4905 \count\csname mpfootins#1\endcsname 500
4906 \multiply\dimen\csname mpfootins#1\endcsname by \tw@}
4907
4908 %

```

```

\twocolvfootnoteX \twocolvfootnoteX{<series>}
4909 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\twocolvfootnoteX}[2]{%
4910 \beforeinsertion@X{#1}%

```



```

4911 \let\ bidi@RTL@everypar\relax%
4912 \insert\csname footins#1\endcsname\bgroup%
4913 \hsize=\expandafter\dimexpr\csuse{widthX@#1}\relax%
4914 \noindent\csuse{bhooknoteX@#1}%
4915 \csuse{notefontsizeX@#1}%
4916 \footsplitskips%
4917 \spaceskip=\z@skip \xspaceskip=\z@skip%
4918 \@nameuse{footfmt#1}{#1}{#2}\egroup}
4919
4920 %

```

`\twocolfootfmtX` `\twocolfootfmtX{<series>}`

```

4921 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\twocolfootfmtX}[2]{%
4922 \protected@edef\@currentlabel{%
4923 \@nameuse{@thefnmark#1}%
4924 }%
4925 \normal@pars%
4926 \sethangindentX{#1}%
4927 \hsize \csuse{hsizetwocolX@#1}%
4928 \setparindentX{#1}%
4929 \tolerance=5000\relax%
4930 \par%
4931 \@tempdima=\parindent%
4932 \csuse{colalignX@#1}%
4933 \parindent=\@tempdima%
4934 {\hspace{\parindent}%
4935 \csuse{notenumentX@#1}\wrapped@footfootmarkX{#1}\strut%
4936 \csuse{wrapcontentX@#1}{#2}%
4937 \strut\par}%
4938 \allowbreak%
4939 }%
4940
4941 %

```

`\twocolfootgroupX` `\twocolfootgroupX{<series>}`

`\mptwocolfootgroupX`

```

4942 \newcommand*{\twocolfootgroupX}[1]{\csuse{bhookgroupX@#1}\csuse{
notefontsizeX@#1}
4943 \splittopskip=\ht\strutbox
4944 \expandafter
4945 \rigidbalanceX\csname footins#1\endcsname \tw@ \splittopskip}}
4946
4947 \newcommand*{\mptwocolfootgroupX}[1]{%
4948 \vskip\skip\@nameuse{mpfootins#1}
4949 \ifl@dpairing\ifparledgroup%
4950 \leavevmode\marks\parledgroup@{begin}%
4951 \marks\parledgroup@series{#1}%
4952 \marks\parledgroup@type{footnoteX}%
4953 \fi\fi\normalcolor
4954 \ifparledgroup%
4955 \ifl@dpairing%
4956 \else%
4957 \setnoteswidthliketwocolumnsX@{#1}%

```

```

4958 \setnotesXpositionliketwocolumns@{#1}%
4959 \print@footnoteXrule{#1}%
4960 \fi%
4961 \else%
4962 \setnoteswidthliketwocolumnsX@{#1}%
4963 \setnotesXpositionliketwocolumns@{#1}%
4964 \print@footnoteXrule{#1}%
4965 \fi%
4966 \csuse{bhookgroupX@#1}%
4967 \splittopskip=\ht\strutbox
4968 \expandafter
4969 \rigidbalanceX\csname mpfootins#1\endcsname \tw@ \splittopskip}}
4970
4971 %

```

#### 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]{%
4972
4973 \csgdef{series@displayX#1}{threecol}
4974 \expandafter\let\csname regvfootnote#1\endcsname=\threecolvfootnoteX
4975 \expandafter\let\csname footfmt#1\endcsname=\threecolfootfmtX
4976 \expandafter\let\csname footgroup#1\endcsname=\threecolfootgroupX
4977 \dimen\csname footins#1\endcsname=\csuse{maxhnotesX@#1}%
4978 \skip\csname footins#1\endcsname=\csuse{beforenotesX@#1}%
4979 \advance\skip\csname footins#1\endcsname by \csuse{afterruleX@#1}\relax%
4980 \threecolfootsetupX{#1}
4981 \ifnoledgroup@\else%
4982 \expandafter\let\csname mpvfootnote#1\endcsname=\mpnormalvfootnoteX
4983 \expandafter\let\csname mpfootgroup#1\endcsname=\mpthreecolfootgroupX
4984 \skip\csname mpfootins#1\endcsname=\csuse{beforenotesX@#1}%
4985 \advance\skip\csname mpfootins#1\endcsname by \csuse{afterruleX@#1}
4986 \mpthreecolfootsetupX{#1}
4987 \fi%
4988 }
4989
4990 %

```

`\threecolfootsetupX` `\threecolfootsetupX{<series>}`  
`\mpthreecolfootsetupX`

```

4991 \newcommand*{\threecolfootsetupX}[1]{%
4992 \count\csname footins#1\endcsname 333
4993 \csxdef{default@footins#1}{333}%Use this to confine the notes to one
side only
4994 \multiply\dimen\csname footins#1\endcsname by \thr@@}
4995 \newcommand*{\mpthreecolfootsetupX}[1]{%
4996 \count\csname mpfootins#1\endcsname 333
4997 \multiply\dimen\csname mpfootins#1\endcsname by \thr@@}
4998
4999 %

```

```

\threecolvfootnoteX \threecolvfootnoteX{<series>}{<text>}
5000 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\threecolvfootnoteX}[2]{
%
5001 \let\bidirTL@everypar\relax%
5002 \beforeinsertion@X{#1}%
5003 \insert\csname footins#1\endcsname\bgroup%
5004 \hsize=\expandafter\dimexpr\csuse{widthX@#1}\relax%
5005 \noindent\csuse{bhooknoteX@#1}%
5006 \csuse{notefontsizeX@#1}%
5007 \footsplitskips%
5008 \@nameuse{footfmt#1}{#1}{#2}\egroup}
5009
5010 %

```

```

\threecolfootfmtX \threecolfootfmtX{<series>}
5011 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\threecolfootfmtX}[2]{%
5012 \protected@edef\@currentlabel{%
5013 \nameuse{@thefnmark#1}%
5014 }%
5015 \sethangindentX{#1}%
5016 \normal@pars%
5017 \hsize \csuse{hsizethreecolX@#1}%
5018 \setparindentX{#1}%
5019 \tolerance=5000\relax%
5020 \@tempdima=\parindent%
5021 \csuse{colalignX@#1}%
5022 \parindent=\@tempdima%
5023 {\hspace{\parindent}%
5024 \csuse{notenumfontX@#1}\wrapped@footfootmarkX{#1}\strut%
5025 \csuse{wrapcontentX@#1}{#2}%
5026 \strut\par}\allowbreak}
5027
5028 %

```

```

\threecolfootgroupX \threecolfootgroupX{<series>}
\mpthreecolfootgroupX
5029 \newcommand*{\threecolfootgroupX}[1]{\csuse{bhookgroupX@#1}\csuse{
notefontsizeX@#1}
5030 \splittopskip=\ht\strutbox
5031 \expandafter
5032 \rigidbalanceX\csname footins#1\endcsname \thr@@ \splittopskip}}
5033
5034 \newcommand*{\mpthreecolfootgroupX}[1]{%
5035 \vskip\skip\@nameuse{mpfootins#1}
5036 \ifl@dpairing\ifparledgroup
5037 \leavevmode\marks\parledgroup@{begin}%
5038 \marks\parledgroup@series{#1}%
5039 \marks\parledgroup@type{footnoteX}%
5040 \fi\fi\normalcolor
5041 \ifparledgroup%
5042 \ifl@dpairing%
5043 \else%
5044 \setnoteswidthliketwocolumnsX@{#1}%

```

```

5045 \setnotesXpositionliketwocolumns@{#1}%
5046 \print@footnoteXrule{#1}%
5047 \fi%
5048 \else%
5049 \setnoteswidthliketwocolumnsX@{#1}%
5050 \setnotesXpositionliketwocolumns@{#1}%
5051 \print@footnoteXrule{#1}%
5052 \fi%
5053 \csuse{bhookgroupX@#1}%
5054 \splittopskip=\ht\strutbox
5055 \expandafter
5056 \rigidbalanceX\csname mpfootins#1\endcsname \thr@@ \splittopskip}}
5057
5058 %

```

#### XIV.4.5 Paragraphed footnotes

The following macros set footnotes as one paragraph.

`\arrangementX@paragraph` `\footparagraphX{series}`

```

5059 \newcommand*{\arrangementX@paragraph}[1]{%
5060 \csgdef{series@displayX#1}{paragraph}%
5061 \expandafter\let\csname footstart#1\endcsname=\parafootstartX
5062 \expandafter\let\csname regvfootnote#1\endcsname=\para@vfootnoteX
5063 \expandafter\let\csname footfmt#1\endcsname=\parafootfmtX
5064 \expandafter\let\csname footgroup#1\endcsname=\para@footgroupX
5065 \expandafter\let\csname footnoterule#1\endcsname=\normalfootnoteruleX
5066 \count\csname footins#1\endcsname=1000
5067 \csxdef{default@footins#1}{1000}%Use this to confine the notes to one
side only
5068 \dimen\csname footins#1\endcsname=\csuse{maxhnotesX@#1}
5069 \skip\csname footins#1\endcsname=\csuse{beforenotesX@#1}%
5070 \advance\skip\csname footins#1\endcsname by\csuse{afterruleX@#1}%
5071 \para@footsetupX{#1}
5072 \ifnoledgroup@\else
5073 \expandafter\let\csname mpvfootnote#1\endcsname=\mppara@vfootnoteX
5074 \expandafter\let\csname mpfootgroup#1\endcsname=\mppara@footgroupX
5075 \count\csname mpfootins#1\endcsname=1000
5076 \dimen\csname mpfootins#1\endcsname=\csuse{maxhnotesX@#1}
5077 \skip\csname mpfootins#1\endcsname=\csuse{beforenotesX@#1}%
5078 \advance\skip\csname mpfootins#1\endcsname by\csuse{afterruleX@#1}%
5079 \fi
5080 }
5081
5082 %

```

`\para@footsetupX` `\para@footsetupX{series}`

```

5083 \newcommand*{\para@footsetupX}[1]{\csuse{bhookgroupX@#1}\csuse{
notefontsizeX@#1}
5084 \setnoteswidthliketwocolumnsX@{#1}%
5085 \ifcsemtyp{widthX@#1}%
5086 {}%

```

```

5087   {\columnwidth=\expandafter\dimexpr\csuse{widthX@#1}\relax}%
5088   \dimen0=\baselineskip
5089   \multiply\dimen0 by 1024
5090   \divide\dimen0 by \columnwidth \multiply\dimen0 by \footfudgefiddle\relax
5091   %
5092   \expandafter
5093   \xdef\csname footfudgefactor#1\endcsname{%
5094     \expandafter\strip@pt\dimen0 }}
5095   %

```

`\parafootstartX` `\parafootstartX{<series>}`

```

5096 \newcommand*\parafootstartX[1]{%
5097   \ifdimequal{Opt}{\prenotesX@}{}%
5098   {%
5099     \iftoggle{prenotesX@}{%
5100       \togglefalse{prenotesX@}%
5101       \skip\csname footins#1\endcsname=%
5102       \glueexpr\csuse{prenotesX@}+\csuse{afterruleX@#1}\relax%
5103     }%
5104   }%
5105 }%
5106 \leftskip=\z@
5107 \rightskip=\z@
5108 \parfillskip=Opt plus 1fil
5109 \setparindentX{#1}%
5110 \vskip\skip\@nameuse{footins#1}%
5111 \setnoteswidthliketwocolumnsX@{#1}%
5112 \setnotesXpositionliketwocolumns@{#1}%
5113 \print@footnoteXrule{#1}%
5114 }
5115 %
5116 %

```

`\para@vfootnoteX` `\para@vfootnoteX{<series>}{<text>}`

`\mppara@vfootnoteX`

```

5117 \newcommand*\para@vfootnoteX[2]{%
5118   \csuse{beforeinsertingX@#1}%
5119   \insert\csname footins#1\endcsname%
5120   \bgroup
5121     \csuse{notefontsizeX@#1}
5122     \footsplitskips
5123     \setbox0=\vbox{\hsize=\maxdimen%
5124       \let\bidir@RTL@everypar\@empty%
5125       \insert@txtbeforenotesX{#1}%
5126       \noindent\csuse{bhooknoteX@#1}%
5127       \@nameuse{footfmt#1}{#1}{#2}}%
5128     \setbox0=\hbox{\unvvhX{0}{#1}}%
5129     \dp0=\z@
5130     \ht0=\csname footfudgefactor#1\endcsname\wd0
5131     \box0
5132     \penalty0
5133   \egroup}

```

```

5134 \newcommand*{\mppara@vfootnoteX}[3]{%
5135   \get@thisfootnoteX{#1}%
5136   \get@fnmarkX{#1}{\thisc@footnote}{#3}%
5137   \edef\this@footnoteX@reading{\the\csname footnote#1@reading\endcsname}%
5138   \global\setbox\@nameuse{mpfootins#1}\vbox{%
5139     \unvbox\@nameuse{mpfootins#1}
5140     \csuse{notefontsizeX@#1}
5141     \footsplitskips
5142     \setbox0=\vbox{\hsize=\maxdimen%
5143       \let\bidirectional@everypar\empty%
5144       \noindent\color@begingroup%
5145       \csuse{bhooknoteX@#1}%
5146       \@nameuse{footfmt#1}{#1}{#2}\color@endgroup}%
5147     \setbox0=\hbox{\unvxhX{0}{#1}}%
5148     \dp0=\z@
5149     \ht0=\csname footfudgefactor#1\endcsname\wd0
5150     \box0
5151     \penalty0}}
5152 %
5153 %

```

```

\unvxhX 5154 \newcommand*{\unvxhX}[2]{% 2th is optional for retro-compatibility
5155   \setbox0=\vbox{\unvbox#1%
5156     \global\setbox1=\lastbox}%
5157   \unhbox1
5158   \unskip           % remove \rightskip,
5159   \unskip           % remove \parfillskip,
5160   \unpenalty       % remove \penalty of 10000,
5161   \hskip\csuse{afternoteX@#2}%
5162   \relax}% but add the glue to go between the notes
5163 %
5164 %

```

`\parafootfmtX` `\parafootfmtX{<series>}`

```

5165 \newcommand*{\parafootfmtX}[2]{%
5166   \protected@edef\@currentlabel{%
5167     \@nameuse{@thefnmark#1}%
5168   }%
5169   \insertparafootsepX{#1}%
5170   \ledsetnormalparstuff@common%
5171   {\csuse{notenumfontX@#1}%
5172    \csuse{notenumfontX@#1}%
5173    \wrapped@footfootmarkX{#1}%
5174    \strut%
5175    \csuse{wrapcontentX@#1}{#2}%
5176    \penalty-10}}
5177 %
5178 %

```

`\para@footgroupX` `\para@footgroupX{<series>}`  
`\mppara@footgroupX`

```

5179 \newcommand*{\para@footgroupX}[1]{%

```

```

5180 \hspace=\expandafter\dimexpr\csuse{widthX@#1}\relax%
5181 \unvbox\csname footins#1\endcsname
5182 \ifcsstring{raggedX@#1}{L}{\RaggedLeft}{}%
5183 \ifcsstring{raggedX@#1}{R}{\RaggedRight}{}%
5184 \makehboxofhboxes
5185 \setbox0=\hbox{\unhbox0 \removehboxes}%
5186 \csuse{bhookgroupX@#1}
5187 \csuse{notefontsizeX@#1}
5188 \unhbox0\par}
5189
5190 \newcommand*\mppara@footgroupX}[1]{%
5191 \setnoteswidthliketwocolumnsX@{#1}%
5192 \vskip\skip\@nameuse{mpfootins#1}
5193 \ifl@dpairing\ifparledgroup
5194 \leavevmode%
5195 \leavevmode\marks\parledgroup@{begin}%
5196 \marks\parledgroup@series{#1}%
5197 \marks\parledgroup@type{footnoteX}%
5198 \fi\fi\normalcolor
5199 \ifparledgroup%
5200 \ifl@dpairing%
5201 \else%
5202 \setnoteswidthliketwocolumnsX@{#1}%
5203 \setnotesXpositionliketwocolumns@{#1}%
5204 \print@footnoteXrule{#1}%
5205 \fi%
5206 \else%
5207 \setnoteswidthliketwocolumnsX@{#1}%
5208 \setnotesXpositionliketwocolumns@{#1}%
5209 \print@footnoteXrule{#1}%
5210 \fi%
5211 \unvbox\csname mpfootins#1\endcsname
5212 \ifcsstring{raggedX@#1}{L}{\RaggedLeft}{}%
5213 \ifcsstring{raggedX@#1}{R}{\RaggedRight}{}%
5214 \makehboxofhboxes
5215 \setbox0=\hbox{\unhbox0 \removehboxes}%
5216 \csuse{bhookgroupX@#1}%
5217 \csuse{notefontsizeX@#1}%
5218 \setparindentX{#1}%
5219 \unhbox0\par}}
5220
5221 %

```

**Insertion of the footnotes separator** The command `\insertparafootsepX{<series>}` must be called at the beginning of `\parafootftmX`.

```

\insertparafootsepX22 \newcommand{\insertparafootsepX}[1]{%
5223 \ifledRcol%
5224 \ifnumequal{\csuse{prevpage#1@numR}}{\page@numR}%
5225 {\csuse{parafootsepX@#1}}%
5226 {}%
5227 \global\csname prevpage#1@numR\endcsname=\page@numR%
5228 \else%

```

```

5229 \ifboolexpr{bool{!@dprintingpages} or bool{!@dprintingcolumns} or bool
{numbering}}{%
5230 \ifnumequal{\csuse{prevpage#1@num}}{\page@num}%
5231 {\csuse{parafootsepX@#1}}%
5232 {}%
5233 \global\csname prevpage#1@num\endcsname=\page@num%
5234 }{%We can't use directly the page counter, see https://tex.stackexchange.com/a/239457/7712
5235 \global\expandafter\advance\csname footnote#1@parafootsepX\endcsname
by 1%
5236 \edef\@tmpa{\parafootsepX@#1\the\csname footnote#1@parafootsepX\endcsname}%
5237 \expandafter\edlabel\expandafter{\@tmpa}%
5238 \edef\@tmpb{\expandafter\xpageref\expandafter{\@tmpa}}%
5239 \IfStrEq{\csuse{prevpage#1@num@outnumbering}}{\@tmpb}%
5240 {\csuse{parafootsepX@#1}}%
5241 {}%
5242 \global\cslet{prevpage#1@num@outnumbering}{\@tmpb}%
5243 }%
5244 \fi%
5245 }
5246 %

```

#### 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.

```

5247 \newcommand{\wrapped@footfootmarkX}[1]{%
5248 \ifdefined\hypertarget%
5249 \hyperlink%
5250 {@bodyfootmark#1\csuse{hr@thefnmark#1}}%
5251 {\@nameuse{footfootmark#1}}%
5252 \Hy@raisedlink{%
5253 \hypertarget%
5254 {@footnotemark#1\csuse{hr@thefnmark#1}}%
5255 {}%
5256 }%
5257 \else%
5258 \@nameuse{footfootmark#1}%
5259 \fi%
5260 }%
5261 %

```

`\wrapped@bodyfootmarkX` `\wrapped@bodyfootmarkX` prints the footnote mark of the text body, wrapped in `hyperref` package's commands, if needed.

```

5262 \newcommand{\wrapped@bodyfootmarkX}[1]{%
5263 \ifdefined\hypertarget%
5264 \hyperlink%
5265 {@footnotemark#1\csuse{hr@thefnmark#1}}%
5266 {\@nameuse{bodyfootmark#1}}%
5267 \Hy@raisedlink{%
5268 \hypertarget%

```



```

5269     {@bodyfootmark#1@\csuse{hr@thefnmark#1}}%
5270     {}%
5271   }%
5272   \else%
5273     \@nameuse{bodyfootmark#1}%
5274   \fi%
5275 }%
5276 %

```

## 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 77 \newcommand*\ledsetnormalparstuff@common{%
\Xledsetnormalparstuff 78   \ifluatex%
\ledsetnormalparstuffX 79   \ifdefstring{\footnote@luatextextdir}{TLT}{}%
5280     {\textdir\footnote@luatextextdir}%
5281   \pardir\footnote@luatexpardir%
5282   \fi%
5283   \csuse{\csuse{footnote@dir}}%
5284   \normal@pars%
5285   \parfillskip \z@ \@plus 1fil}%
5286
5287 \newcommand*\Xledsetnormalparstuff}[1]{%
5288   \ledsetnormalparstuff@common%
5289   \Xsetparindent{#1}%
5290   \iftoggle{Xparindent@#1}{\hspace{\parindent}}{}%
5291 }%
5292
5293 \newcommand*\ledsetnormalparstuffX}[1]{%
5294   \ledsetnormalparstuff@common%
5295   \setparindentX{#1}%
5296   \iftoggle{parindentX@#1}{\hspace{\parindent}}{}%
5297 }%
5298 %

```

## 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 `\setXnoteswidthliketwocolumns@` and `\setnoteswidthliketwocolumnsX@` for paragraphed notes.

```

5299
5300 \newdimen\old@hsize%
5301 \AtBeginDocument{\old@hsize=\hsize}%
5302
5303 \newcommand{\setXnoteswidthliketwocolumns@[1]}{%
5304   \global\let\hsize@fornote=\hsize%
5305   \global\old@hsize=\hsize%
5306   \let\old@columnwidth=\columnwidth%
5307   \iftoggle{Xnoteswidthliketwocolumns@#1}%
5308     {%
5309       \setwidthliketwocolumns%
5310       \global\let\hsize@fornote=\hsize%
5311     }%
5312   {%
5313     \let\hsize=\hsize@fornote%
5314     \let\columnwidth=\old@columnwidth%
5315   }%
5316
5317 \newcommand{\setnoteswidthliketwocolumnsX@[1]}{%
5318   \global\let\hsize@fornote=\hsize%
5319   \global\old@hsize=\hsize%
5320   \let\old@columnwidth=\columnwidth%
5321   \iftoggle{noteswidthliketwocolumnsX@#1}%
5322     {%
5323       \setwidthliketwocolumns%
5324       \global\let\hsize@fornote=\hsize%
5325     }%
5326   {%
5327     \let\hsize=\hsize@fornote%
5328     \let\columnwidth=\old@columnwidth%
5329   }%
5330
5331 %

```

`\setXnotespositionliketwocolumns@` and `\setnotesXpositionliketwocolumns@` These two commands set the position of the critical / familiar footnotes, depending on the hooks `Xnoteswidthliketwocolumns` and `noteswidthliketwocolumnsX`. They call commands which are defined only in `reledpar`, because this feature has no sense without `reledpar`.

```

5332 \newcommand{\setXnotespositionliketwocolumns@[1]}{%
5333   \iftoggle{Xnoteswidthliketwocolumns@#1}{%
5334     \csuse{setnotespositionliketwocolumns@\columns@position}%
5335   }{}%
5336 }%
5337
5338 \newcommand{\setnotesXpositionliketwocolumns@[1]}{%
5339   \iftoggle{noteswidthliketwocolumnsX@#1}{%
5340     \csuse{setnotespositionliketwocolumns@\columns@position}%
5341   }{}%
5342 }%

```

```
5343
5344 %
```

## 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
5345 \def\@fnpos{familiar-critical}
5346 \def\@mpfnpos{critical-familiar}
5347 \newcommand{\fnpos}[1]{\xdef\@fnpos{#1}}
5348 \newcommand{\mpfnpos}[1]{\xdef\@mpfnpos{#1}}
5349 %
```

## XVIII Footnotes' rule

### XVIII.1 Printing only one footnote rule

The `\Xonlyonerule` and `\onlyoneruleX` macros allow to print only one footnote rule in each page for, respectively, critical and familiar footnotes series. They just switch the low level `\Xonlyonerule@` and `\onlyoneruleX@` toggles.

```
Xonlyonerule@50 \unless\ifnocritical@%
Xonlyonerule@51 \newtoggle{Xonlyonerule@}%
onlyoneruleX@52 \newcommandx{\Xonlyonerule}[1][1=true]{%
onlyoneruleX@53 \settoggle{Xonlyonerule@}{#1}%
5354 }%
5355 \fi%
5356 \unless\ifnofamiliar@%
5357 \newtoggle{onlyoneruleX@}%
5358 \newcommandx{\onlyoneruleX}[1][1=true]{%
5359 \settoggle{onlyoneruleX@}{#1}%
5360 }%
5361 \fi%
5362 %
```

### XVIII.2 Dealing with multiple columns

Because the footnotes' rules can be shifted to the right when footnotes are set like two columns, we do not print them directly, but we put them in a `\vbox`.

```
\print@Xfootnoterule@63 \newtoggle{Xfirstfootnoterule@printed}%
\print@footnoteruleX@64 \newcommand{\print@Xfootnoterule}[1]{%
rstfootnoterule@printed@65 \ifboolexpr{%
rstfootnoteruleX@printed@66 (not togl{Xonlyonerule@} or not togl{Xfirstfootnoterule@printed})%
5367 and%
5368 not (togl{onlyonerule@} and (togl{firstfootnoteruleX@printed} or togl{
Xfirstfootnoterule@printed}))%
5369 }{%
5370 \vskip-\csuse{Xafterrule@#1}%Because count in \dimen\csuse{#1footins}%
```

```

5371 \nointerlineskip%
5372 \moveleft-\leftskip\ vbox{\csuse{#1footnoterule}}%
5373 \nointerlineskip%
5374 \vskip\csuse{Xafterrule@#1}%
5375 }{}%
5376 \global\toggletrue{Xfirstfootnoterule@printed}%
5377 }%
5378
5379 \newtoggle{firstfootnoteruleX@printed}%
5380 \newcommand{\print@footnoterule}[1]{%
5381 \ifboolexpr{%
5382 (not togl{onlyoneruleX@} or not togl{firstfootnoteruleX@printed})%
5383 and%
5384 not (togl{onlyonerule@} and (togl{firstfootnoteruleX@printed} or togl{
Xfirstfootnoterule@printed}))}%
5385 }{}%
5386 \vskip-\csuse{afterruleX@#1}%Because count in \dimen\csuse{footins#1}%
5387 \nointerlineskip%
5388 \moveleft-\leftskip\ vbox{\csuse{footnoterule#1}}%
5389 \nointerlineskip%
5390 \vskip\csuse{afterruleX@#1}%
5391 }{}%
5392 \global\toggletrue{firstfootnoteruleX@printed}%
5393 }%
5394
5395 %

```

## 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.

```
5396 \newtoggle{Xprenotes@}%
5397 \toggletrue{Xprenotes@}%
5398 \newcommand{Xprenotes@}{Opt}%
5399 \newcommand*{Xprenotes}[1]{\renewcommand{Xprenotes@}{#1}}%
5400 \newcommand{preXnotes}[1]{\led@warning@preXnotes@deprecatedXprenotes{#1}}
%For compatibility
5401 %
```

The same, but for familiar footnotes.

```
\Xprenotes@02 \newtoggle{prenotesX@}
\Xprenotes@03 \toggletrue{prenotesX@}
5404 \newcommand{prenotesX@}{Opt}
5405 \newcommand*{prenotesX}[1]{\renewcommand{prenotesX@}{#1}}
5406 %
```

## XIX.3 Internal commands

```
firstXseries@07 \gdef\firstXseries@{}
prepare@Xprenotes08 \newcommand{prepare@Xprenotes}[1]{%
5409 \ifdimequal{Opt}{Xprenotes@}%
5410 {}%
5411 {%
5412 \IfStrEq{\firstXseries@}{}%
5413 \global\skip\csuse{#1footins}=Xprenotes@%
5414 \global\advance\skip\csname #1footins\endcsname by\csuse{Xafterrule@
#1}%
5415 \gdef\firstXseries@{#1}%
5416 }%
5417 {%
5418 \ifseriesbefore{#1}{\firstXseries@}%
5419 {%
5420 \global\skip\csuse{#1footins}=\csuse{Xbeforenotes@\firstXseries@}%
5421 \global\advance\skip\csname #1footins\endcsname by\csuse{Xafterrule@
#1}%
5422 \gdef\firstXseries@{#1}%
5423 }%
5424 {%
```

```

5425 }%
5426 }%
5427 }
5428 %

```

The same thing is required for familiar notes and `\prenotesX`.

```

firstseriesX@29 \gdef\firstseriesX@{}
prepare@prenotesX30 \newcommand{\prepare@prenotesX}[1]{%
5431 \ifdimequal{Opt}{\prenotesX@}%
5432 {}%
5433 {%
5434 \IfStrEq{\firstseriesX@}{}%
5435 \global\skip\csuse{footins#1}=\prenotesX@%
5436 \global\advance\skip\csname footins#1\endcsname by\csuse{afterruleX@
#1}%
5437 \gdef\firstseriesX@{#1}%
5438 }%
5439 {%
5440 \ifseriesbefore{#1}{\firstseriesX@}%
5441 {%
5442 \global\skip\csuse{footins#1}=\csuse{beforenotesX@\firstseriesX@}%
5443 \global\advance\skip\csname footins#1\endcsname by\csuse{afterruleX@
#1}%
5444 \gdef\firstseriesX@{#1}%
5445 }%
5446 {}%
5447 }%
5448 }%
5449 }
5450 %

```

## XX Endnotes

First, check the `noend` option.

```

5451 \ifbool{noend@}{%Used instead of \ifnoend@ to prevent expansion problem
5452 %

```

### 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.

```

5453 \newcommand{\l@dend@open}[1]{%
5454 \global\booltrue{l@dend@#1}%
5455 \expandafter\immediate%
5456 \expandafter\openout%
5457 \csname l@d@#1end\endcsname%

```

```

5458     =\l@auxdir\jobname.#1end\relax%
5459 }%
5460 \newcommand{\l@dend@close}[1]{%
5461   \global\boolfalse{l@dend@#1}%
5462   \expandafter\immediate%
5463   \expandafter\closeout\csname l@d@#1end\endcsname%
5464 }%
5465 %
5466 %

```

**\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.

```

5467 \newcommand{\l@dend@stuff}{%
5468   \def\do##1{%
5469     \ifbool{l@dend@##1}{%
5470       {\l@dend@open{##1}}%
5471       \expandafter\immediate\expandafter\write\csname l@d@##1end\endcsname{\
string\l@d@section{\the\section@num}\@percentchar}%
5472     }%
5473     \dolistloop{\@series}%
5474   }%
5475 %
5476 %

```

**\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.

```

5477 \global\newbool{parapparatus@}{\long}\def\endprint#1#2#3#4#5#6#7{%
5478   \iftoggle{Xendfirstnote@#4}{%
5479     \global\togglefalse{Xendfirstnote@#4}%
5480     \csuse{Xendtxtbeforenotes@#4}%
5481   }{%
5482     \csuse{Xendbhooknote@#4}%
5483     \csuse{Xendnotefontsize@#4}%
5484     \hangindent=\csuse{Xendhangindent@#4}%

```

```

5485 \Xendstorelineinfo{#1}{#4}%
5486 \ifXendinsertsep%
5487   \hskip\csuse{Xendafternote@#4}\relax%
5488   \ifcsdef{prevedline#4}%
5489     {\ifcsequal{prevedline#4}{lineinfo@}%
5490       {\ifcsequal{Xendsymnum@#4}{\csuse{Xendsep@#4}}{}}%
5491       {\csuse{Xendsep@#4}}%
5492     }%
5493     {\csuse{Xendsep@#4}}%
5494 \else%
5495   \iftoggle{Xendparagraph@#4}%
5496     {\global\Xendinsertsep@true}%
5497     {}%
5498 \fi%
5499 \xdef\@currentseries{#4}%
5500 \def\do##1{%
5501   \setkeys[mac]{truefootnoteoption}{##1}%
5502 }%
5503 \notblank{#5}{\docsvlist{#5}}{}%
5504 \IfStrEq{#6}{R}{\ledRcol@true}{}%
5505 \def\@this@crossref@start{#7:start}%
5506 \def\@this@crossref@end{#7:end}%
5507 \printlineendnote{#1}{#4}%
5508 \IfStrEq{#6}{R}{\ledRcol@false}{}%
5509 \nottoggle{Xendlemmadisablefontselection@#4}%
5510   {\select@lemmafонт#1|}%
5511   {}%
5512 \bgroup%
5513   \csuse{Xendlemmafонт@#4}%
5514   \csuse{Xendwraplemma@#4}{#2}%
5515 \egroup%
5516 \ifboolexpr{%
5517   togl {nosep@}%
5518   or test{\ifcsequal{Xendlemmaseparator@#4}}%
5519 }%
5520 {\hskip\csuse{Xendinplaceoflemmaseparator@#4}\relax}%
5521 {\nobreak%
5522   \hskip\csuse{Xendbeforelemmaseparator@#4}%
5523   \csuse{Xendlemmaseparator@#4}%
5524   \hskip\csuse{Xendafterlemmaseparator@#4}%
5525   \relax%
5526 }%
5527 \csuse{Xendwrapcontent@#4}{#3}%
5528 \nottoggle{Xendparagraph@#4}{\par}{}%
5529 \def\do##1{%
5530   \setkeys[mac]{falsefootnoteoption}{##1}%
5531 }%
5532 \notblank{#5}{\docsvlist{#5}}{}%
5533 }}%
5534
5535 \let\l@d@section=\@gobble
5536
5537 %

```



`\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.

```

5538 \newcommand{\Xendstorelineinfo}[2]{%
5539   \l@dp@rsefootspec#1}%
5540   \iftoggle{Xendnumberonlyfirstintwolines@#2}{%
5541     \xdef\lineinfo@{\l@dparsedstartpage - \l@dparsedstartline - \
5542     \l@dparsedstartsub - \l@dparsedendpage - \l@dparsedendline - \
5543     \l@dparsedendsub}%
5544     }%
5545     {%
5546     \xdef\lineinfo@{\l@dparsedstartpage - \l@dparsedstartline - \
5547     \l@dparsedstartsub}%
5548     }%
5549   }%
5550   %

```

`\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.

```

5548 \newcommand{\printlineendnote}[2]{%
5549   \ifboolexpr{%
5550     togl {nonum@}%
5551     or togl {Xendnonumber@#2}%
5552   }%
5553   {\hspace{\csuse{Xendinplaceofnumber@#2}}}%
5554   {%
5555     \iftoggle{Xendnumberonlyfirstinline@#2}{%
5556       {\ifcsdef{prevendline#2}%
5557         {\ifcsequal{prevendline#2}{lineinfo@}%
5558           {%
5559             \csuse{Xendbhookinplaceofnumber@#2}%
5560             \ifcsempty{Xendsymmlinenum@#2}{%
5561               {\hspace{\csuse{Xendinplaceofnumber@#2}}}%
5562               {\printsymmlineendnotearea{#2}}}%
5563             \csuse{Xendahookinplaceofnumber@#2}%
5564             }%
5565             {\printlineendnotearea{#1}{#2}}}%
5566             {\printlineendnotearea{#1}{#2}}}%
5567           }%
5568           {\printlineendnotearea{#1}{#2}}%We keep every time line
5569     \csxdef{prevendline#2}{\lineinfo@}%
5570     }%
5571   }%
5572   %

```

```

73 \newcommand{\printsymmlineendnotearea}[1]{%
5574   \hspace{\csuse{Xendbeforemsymmlinenum@#1}}%
5575   \csuse{Xendnotenumfont@#1}%
5576   \ifdimequal{\csuse{Xendboxsymmlinenum@#1}}{\z@}%

```

```

5577 {\csuse{Xendsymmlinenumber@#1}}}%
5578 {\hbox to \csuse{Xendboxsymmlinenumber@#1}%
5579   {\csuse{Xendsymmlinenumber@#1}\hfill}%
5580 }%
5581 \hspace{\csuse{Xendaftersymmlinenumber@#1}}}%
5582 }%
5583 %

```

`\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.)

```

5584 \newcommand{\printlineendnotearea}[2]{%
5585   \csuse{Xendhooklinenumber@#2}}%
5586   \hspace{\csuse{Xendbeforenumber@#2}}}%
5587   \bgroup%
5588     \csuse{Xendnotenumfont@#2}}%
5589     \ifdimequal{\csuse{Xendboxlinenumber@#2}}{0pt}%
5590       {\printendlines#1||\ifledRcol@ \@Rlineflag\fi}%
5591       {\leavevmode%
5592         \hbox to \csuse{Xendboxlinenumber@#2}}%
5593         {%
5594           \IfSubStr{RC}{\csuse{Xendboxlinenumberalign@#2}}{\hfill}}}%
5595       \printendlines#1||\ifledRcol@ \@Rlineflag\fi%
5596       \IfSubStr{LC}{\csuse{Xendboxlinenumberalign@#2}}{\hfill}}}%
5597     }%
5598   \egroup%
5599   \hspace{\csuse{Xendaftersymmlinenumber@#2}}}%
5600   \csuse{Xendahooklinenumber@#2}}}%
5601 }%
5602 %

```

## 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*03 \newcommandx{\toendnotes}[2][1,usedefault]{%
\toendnotes*04   \ifboolexpr{bool{numbering} or bool{numberingR}}{%
5605     \def\do##1{%
5606       \expandafter\immediate\expandafter\write\csname l@d@##1end\endcsname%
5607       {\unexpanded{##1}\@percentchar}%
5608     }%
5609     \ifstrempy{##1}%
5610       {\dolistloop{\@series}}%
5611       {\docsvlist{##1}}%
5612     }{\led@err@toendnotes@outsidenumbering}%
5613   }%

```

```

5614 \WithSuffix\newcommandx\toendnotes*[2][1,usedefault]{%
5615   \ifboolexpr{bool{numbering} or bool{numberingR}}{%
5616     \def\do##1{%
5617       \expandafter\immediate\expandafter\write\csname l@d@##1end\endcsname%
5618       {#2\@percentchar}%
5619     }%
5620     \ifstrempy{#1}%
5621       {\dolistloop{\@series}}%
5622       {\docsvlist{#1}}%
5623   }{\led@err@toendnotes@outsidenumbering}%
5624 }%
5625 %

```

### 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.

```

5626 \newif\ifXendinsertsep@%
5627 \newcommand*\doendnotes}[1]{%
5628   \l@dend@close{#1}%
5629   \begingroup
5630     \csxdef{prevpagenum@#1}{}%
5631     \csxdef{prevpagerange@#1}{}%
5632     \global\toggletrue{Xendfirstnote@#1}%
5633     \makeatletter
5634     \expandafter\let\csname #1end\endcsname=\endprint
5635     \input\l@auxdir\jobname.#1end%
5636     \global\Xendinsertsep@false%
5637   \endgroup}
5638 %

```

`\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.

```

5639 \newcommand*\doendnotesbysection}[1]{%
5640   \l@dend@close{#1}%
5641   \csxdef{prevpagenum@#1}{}%
5642   \csxdef{prevpagerange@#1}{}%
5643   \global\expandafter\advance\csname #1end@bysection\endcsname by 1%
5644   \begingroup%
5645     \makeatletter%
5646     \def\l@d@section##1{%
5647       \global\toggletrue{Xendfirstnote@#1}%
5648       \ifnumequal{##1}{\csname #1end@bysection\endcsname}%
5649         {\cslet{#1end}{\endprint}}%
5650         {\cslet{#1end}{\@gobbleseven}}%
5651     }%
5652     \input\l@auxdir\jobname.#1end%
5653   \global\Xendinsertsep@false%

```

```

5654 \endgroup%
5655 }%
5656 %

```

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.

```

5657 }%
5658 %

```

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.

```

\setprintendlines59 \newif\ifl@d@Xendlinenumber%
\ifl@d@Xendlinenumber60 \newcommand*{\setprintendlines}[6]{%
5661 \let\@annot@start@print\relax%
5662 \let\@annot@end@print\relax%
5663 \l@d@pnumfalse%
5664 \l@d@dashfalse%
5665 \l@d@elinfalse%
5666 %

```

Should we print the line number? It depends on hooks and annotations.

```

5667 \l@d@Xendlinenumbertrue%
5668 \ifboolexpr{togl{Xendnolinenumber@\@currentseries}}%
5669 or (%
5670 togl{Xendnolinenumberifannotation@\@currentseries}}%
5671 and not test{\ifdefvoid{\@annot@start}}%
5672 and not test{\ifdefvoid{\@annot@end}}%
5673 )%
5674 }{%
5675 \l@d@Xendlinenumberfalse%
5676 }{%
5677 %

```

We print the second page number only if the ending page number is different

```

5678 \ifnum#4=#1 \else
5679 \l@d@pnumtrue
5680 \l@d@dashtrue
5681 \fi
5682 %

```

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.

```

5683 \ifboolexpr{%
5684   bool{!@d@pnum}%
5685   or not test{\ifnumequal{#2}{#5}}%
5686   and bool{!@d@Xendlinenumber}%
5687 }{%
5688   \!@d@elintrue%
5689   \!@d@dashtrue%
5690   \!@d@elintrue
5691   \!@d@dashtrue
5692   \unless\ifx\relax\annot@end%
5693   \def\@annot@end@print{%
5694     \!@wrapcs@ifnotemptybox{Xendwraplinenumannotation@\@currentseries
5695 }{\@annot@end}%
5696   }%
5697   \fi%
5698 }%
5699 %

```

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.

```

5700 \if!@d@elin%
5701   \def\@annot@start@print{%
5702     \!@wrapcs@ifnotemptybox{Xendwraplinenumannotation@\@currentseries}{\
5703   annot@start}%
5704   }%
5705   \else%
5706     \ifx\annot@start\annot@end%
5707       \unless\ifx\@annot@start\relax%
5708         \def\@annot@start@print{%
5709           \!@wrapcs@ifnotemptybox{Xendwraplinenumannotation@\@currentseries
5710 }{\@annot@start}%
5711         }%
5712         \fi%
5713       \else%
5714         \ifx\@annot@end@print\relax%
5715         \def\@annot@start@print{%
5716           \!@wrapcs@ifnotemptybox{Xendwraplinenumannotation@\@currentseries
5717 }{%
5718           \ifx\annot@start\empty\else%
5719             \annot@start%
5720             \ifdefined\linerangesep%
5721               \linerangesep%
5722             \else%
5723               \csuse{Xendlinrangeseparator@\@currentseries}%
5724             \fi%
5725           \fi%
5726           \annot@end%
5727         }%
5728       }%
5729     }%

```

```

5726     \else%
5727     \let\@annot@start@print\@annot@end@print%
5728     \let\@annot@end@print\relax%
5729     \fi%
5730     \fi%
5731 \fi%
5732 %

```

We print the starting sub-line if it is nonzero.

```

5733 \l@d@ssubfalse
5734 \ifnum#3=0 \else
5735     \l@d@ssubtrue
5736 \fi
5737 %

```

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.

```

5738 \l@d@eslfalse
5739 \ifnum#6=0 \else
5740     \ifnum#6=#3
5741         \ifl@d@elin \l@d@esltrue \else \l@d@eslfalse \fi
5742     \else
5743         \l@d@esltrue
5744         \l@d@dashtrue
5745     \fi
5746 \fi%
5747 %

```

```

5748 \ifl@d@dash%
5749     \ifboolexpr{togl{fulllines@} or test{\ifcsempy{Xendtwolines@
@currentseries}}}%
5750     {%
5751     {%
5752     \setistwofollowinglines{#1}{#2}{#4}{#5}%
5753     \ifboolexpr{%
5754     (%
5755         togl {Xendtwolinesbutnotmore@\@currentseries}%
5756         and not%
5757         (%
5758             bool {istwofollowinglines@}%
5759         )%
5760     )%
5761     or%
5762     (%
5763         (not test{\ifnumequal{#1}{#4}})%
5764         and togl{Xendtwolinesonlyinsamepage@\@currentseries}%
5765     )%
5766     }%
5767     }%
5768     {%
5769     \l@d@dashfalse%
5770     \l@d@xtwolinestruetrue%
5771     \l@d@elinfalse%
5772     \l@d@eslfalse%

```

```

5773     \ifcempty{Xendmorethantwolines@\@currentseries}%
5774     {}%
5775     {\ifistwofollowinglines\else%
5776     \l@d@Xmorethantwolinestrue%
5777     \fi%
5778     }%
5779     }%
5780   }%
5781 \fi%
5782 %

```

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

```

5783 \iftoggle{Xendnoidenticallinenumannotation@\@currentseries}{%
5784 \ifx\annot@start\annot@end%
5785 \let\@annot@end@print\relax%
5786 \ifboolexpr{%Do not print the dash if we do not print the line number
5787 test{\ifdefempty{\linenumrep}}%
5788 or not bool {l@d@Xendlinenum}}%
5789 }{%
5790 \l@d@dashfalse%
5791 }{%
5792 \fi%
5793 }{%
5794 %

```

Finally, we check for `\Xendlinenumannotationonlyfirst` and `\Xendlinenumannotationonlyfirstintwo`, and we redefine, if required, `\@annot@start@print` and `\annot@end@print`. We also store the current line number annotations.

```

5795 \iftoggle{Xendlinenumannotationonlyfirst@\@currentseries}{%
5796 \ifboolexpr{%
5797 (%
5798 togl{Xendlinenumannotationonlyfirstintwo@\@currentseries}%
5799 and test{\ifcsequal{annot@start}{prevannot@start@\@currentseries}}%
5800 and test{\ifcsequal{annot@end}{prevannot@end@\@currentseries}}%
5801 )%
5802 or%
5803 (%
5804 not togl{Xendlinenumannotationonlyfirstintwo@\@currentseries}%
5805 test{\ifcsequal{annot@start}{prevannot@start@\@currentseries}}%
5806 )%
5807 }{%
5808 \def\@annot@start@print{%
5809 \l@wrapcs@ifnotemptybox{Xendwraplinenumannotation@\@currentseries
}{\csuse{Xendsymmlinenumannotation@\@currentseries}}%
5810 }%
5811 \let\@annot@end@print\relax%
5812 \ifboolexpr{%Do not print the dash if do not print the line number
5813 test{\ifdefempty{\linenumrep}}%
5814 or not bool {l@d@Xendlinenum}}%
5815 }{%
5816 \l@d@dashfalse%
5817 }{%

```

```

5818 }{}%
5819 \global\cslet{prevannot@start@\@currentseries}{\annot@start}%
5820 \global\cslet{prevannot@end@\@currentseries}{\annot@end}%
5821 }%
5822 {}%
5823 %

End of \setprintendlines.

5824 }%
5825 %

```

`\printendlines` Now we are ready to print it all.

```

5826 \def\printendlines#1|#2|#3|#4|#5|#6|#7|#8|{%
5827   \begingroup
5828   \setprintendlines{#1}{#2}{#3}{#4}{#5}{#6}%
5829 %

```

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.

```

5830 \ifdimequal{\csuse{Xendboxstartlinenum@\@currentseries}}{Opt}%
5831   {\bgroup}%
5832   {\leavevmode\hbox to \csuse{Xendboxstartlinenum@\@currentseries}\bgroup
\hfill}%
5833 %

```

Then, print the starting page number-

```

5834 \ifboolexpr{%
5835   (%
5836     test{\ifcsstring{prevpagenum@\@currentseries}{#1}}%
5837     and not%
5838     (togl{Xendpagenumberonlyfirstifsingle@\@currentseries} and bool{
1@d@pnum})%
5839   )%
5840   or%
5841   (%
5842     test {\ifcsstring{prevpagerange@\@currentseries}{#1-#4}}%
5843   )%
5844 }%
5845 {%
5846   \ifcsempy{Xendsympagenum@\@currentseries}%
5847     {\hspace{\csuse{Xendinplaceofpagenumber@\@currentseries}}}%
5848     {\csuse{Xendsympagenum@\@currentseries}}%
5849   }%
5850   {%
5851     \wrap@edcrossref{\@this@crossref@start}{\printnpnum{#1}}%
5852   }%
5853 %

```

Then, determine what must be printed before the start line.



```

5854 \ifl@d@dash%
5855 \ifl@d@pnum%
5856 \csuse{Xendlineprefixsingle@\@currentseries}%
5857 \else%
5858 \ifcempty{Xendlineprefixmore@\@currentseries}%
5859 {\csuse{Xendlineprefixsingle@\@currentseries}}%
5860 {\csuse{Xendlineprefixmore@\@currentseries}}%
5861 \fi%
5862 \else%
5863 \csuse{Xendlineprefixsingle@\@currentseries}%
5864 \fi%
5865 %

```

Then print the starting line, followed, if needed, by the side flag and the starting sub line number, then the line number annotation.

```

5866 \ifcsstring{Xendlinenumberannotationposition@\@currentseries}{before}%
5867 {\@annot@start@print}%
5868 {}%
5869 \ifl@d@Xendlinenumber%
5870 \wrap@edcrossref{\@this@crossref@start}{%
5871 \ifledRcol@%
5872 \linenumrepR{#2}%
5873 \else%
5874 \linenumrep{#2}%
5875 \fi%
5876 }%
5877 \fi%
5878 \iftoggle{Xendlineflag@\@currentseries}{\ifledRcol@\@Rlineflag\fi}{}%
5879 \iftoggle{Xendnotboxingsubline@\@currentseries}{\egroup}{}%Close the box
5880 \ifl@d@Xendlinenumber%
5881 \ifl@d@ssub%
5882 \csuse{Xendsublinesep@\@currentseries}%
5883 \wrap@edcrossref{\@this@crossref@start}{%
5884 \ifledRcol@%
5885 \sublinenumrepR{#3}%
5886 \else%
5887 \sublinenumrep{#3}%
5888 \fi%
5889 }%
5890 \fi%
5891 \fi%
5892 \ifcsstring{Xendlinenumberannotationposition@\@currentseries}{after}%
5893 {\@annot@start@print}%
5894 {}%
5895 %

```

Close the box.

```

5896 \iftoggle{Xendnotboxingsubline@\@currentseries}{\egroup}%
5897 %

```

Open the box for the ending line number.

```

5898 \ifdimequal{\csuse{Xendboxendlinenum@\@currentseries}}{0pt}%
5899 {\bgroup}%
5900 {\hbox to \csuse{Xendboxendlinenum@\@currentseries}\bgroup}%

```

```

5901 %
Print the dash + the ending line number, or the line number range symbol.
5902 \ifl@d@Xtwolines%
5903 \ifl@d@Xendlinenum%
5904 \ifl@d@Xmorethantwolines%
5905 \wrap@edcrossref{\@this@crossref@end}{\csuse{Xendmorethantwolines@
@currentseries}}%
5906 \else%
5907 \wrap@edcrossref{\@this@crossref@end}{\csuse{Xendtwolines@
@currentseries}}%
5908 \fi%
5909 \fi%
5910 \else%
5911 \ifl@d@dash%
5912 \ifdefined\linerangesep%
5913 \linerangesep%
5914 \else%
5915 \csuse{Xendlinerangeseparator@ \@currentseries}%
5916 \fi%
5917 \fi%
5918 %

```

Print the ending page number.

```

5919 \ifl@d@pnum%
5920 \ifcsstring{prevpagerange@ \@currentseries}{#1-#4}%
5921 {%
5922 \ifcsemtyp{Xendsympagenum@ \@currentseries}%
5923 {\hspace{\csuse{Xendinplaceofpagenumber@ \@currentseries}}}%
5924 {\csuse{Xendsympagenum@ \@currentseries}}%
5925 }%
5926 {%
5927 \wrap@edcrossref{\@this@crossref@end}{\printnnum{#4}}%
5928 }%
5929 \fi%
5930 %

```

Print the ending line number, with, if needed, the line prefix, and followed by the side flag, the subtitle number, and line number annotation.

```

5931 \ifcsstring{Xendlinenumannotationposition@ \@currentseries}{before}%
5932 {\@annot@end@print}%
5933 }%
5934 \ifl@d@Xendlinenum%
5935 \ifl@d@elin%
5936 \ifl@d@pnum\csuse{Xendlineprefixsingle@ \@currentseries}\fi%
5937 \wrap@edcrossref{\@this@crossref@end}{%
5938 \ifledRcol%
5939 \linenumrepR{#5}%
5940 \else%
5941 \linenumrep{#5}%
5942 \fi%
5943 }%
5944 \iftoggle{Xendlineflag@ \@currentseries}{\ifledRcol@ \@Rlineflag\fi
}%

```

```

5945     \fi%
5946     \ifl@d@esl%
5947         \ifl@d@elin%
5948             \csuse{Xendsublinesep@}\@currentseries}%
5949     \fi%
5950     \wrap@edcrossref{\@this@crossref@end}{%
5951         \ifledRcol@%
5952             \sublinenumrepR{#6}%
5953         \else%
5954             \sublinenumrep{#6}%
5955         \fi%
5956     }%
5957 \fi%
5958 \fi%
5959 \ifcsstring{Xendlinenumannotationposition@}\@currentseries}{after}%
5960     {\@annot@end@print}%
5961     {}%
5962 \fi%
5963 %

```

Close the ending line box.

```

5964     \ifdimequal{\csuse{Xendboxendlinenum@}\@currentseries}}{0pt}%
5965     {}%
5966     {\hfill}%Prevent underfull hbox
5967 \egroup%
5968 %

```

And, finally, save, if needed, the current page number for the Xendpagenumberonlyfirst hooks.

```

5969     \iftoggle{Xendpagenumberonlyfirst@}\@currentseries}%
5970     {\iftoggle{Xendpagenumberonlyfirstintwo@}\@currentseries}%
5971     {\csxdef{prevpagerange@}\@currentseries}{#1-#4}}%
5972     {\csxdef{prevpagenum@}\@currentseries}{#4}}%
5973     }%
5974     {}%
5975 %

```

Now, the end of \printendlines macro.

```

5976     \endgroup%
5977 }%
5978
5979 %

```

**\printnpnum** A macro to print a page number in an endnote. Should not be override anymore

```

5980 \newcommand*{\printnpnum}[1]{\csuse{Xendbeforepagenumber@}\@currentseries
5981     }#1\csuse{Xendafterpagenumber@}\@currentseries}
5982 %

```

## XXI Generate series of notes

In this section, X means the name of the series (A, B etc.)

**\series** `\series\series` creates one more new series. It is a public command, which just loops on the private command `\newseries@`.

```
5983 \newcommand{\newseries}[1]{%
5984     \def\do##1{\newseries@{##1}}%
5985     \docsvlist{#1}
5986 }
5987 %
```

**\@series** The `\series@` macro is an etoolbox list, which contains the name of all series.

```
5988 \newcommand{\@series}{}
5989 %
```

The command `\newseries@\series` creates a new series of the footnote.

```
\newseries@90 \newcommand{\newseries@}[1]{
5991 %
```

### XXI.1 Test if series is still existing

```
5992 \xifinlist{#1}{\@series}{\led@warn@SeriesStillExist{#1}}%
5993 {%
5994 %
```

### XXI.2 Init specific to reledpar

When calling `\newseries@` after having loaded `reledpar`, we need to load specific setting.

```
5995 \ifdefined\newseries@par%
5996     \newseries@par{#1}%
5997 \fi%
5998 %
```

### XXI.3 For critical footnotes

Critical footnotes are those which start with letters. We look for the `\nocritical` option of `reledmac`.

```
5999 \unless\ifnocritical@
6000 %
```

#### XXI.3.1 Options

```
6001 \newtoggle{Xlineflag@#1}
6002 \newtoggle{Xparindent@#1}
6003 \newtoggle{Xlemmadisablefontselection@#1}
6004 \csgdef{Xwrapcontent@#1}{}%
6005 \csgdef{Xbeforeinserting@#1}{}%
6006 \csgdef{Xhangindent@#1}{Opt}%
6007 \csgdef{Xragged@#1}{}%
```

```

6008 \csgdef{Xhsizetwocol@#1}{0.45 \hsize}%
6009 \csgdef{Xhsizethreecol@#1}{.3 \hsize}%
6010 \csgdef{Xcolalign@#1}{\raggedright}%
6011 \csgdef{Xnotenumfont@#1}{\normalfont}%
6012 \csgdef{Xnotefontsize@#1}{\footnotesize}%
6013 \csgdef{Xbhooknote@#1}{}%
6014 \csgdef{Xbhookgroup@#1}{}%
6015
6016 \csgdef{Xboxlinenum@#1}{Opt}%
6017 \csgdef{Xboxlinenumalign@#1}{L}%
6018
6019 \csgdef{Xboxstartlinenum@#1}{Opt}%
6020 \newtoggle{Xnotboxingsubline@#1}%
6021 \csgdef{Xboxendlinenum@#1}{Opt}%
6022
6023 \csgdef{Xboxsymlinenum@#1}{Opt}%
6024 \newtoggle{Xgroupbyline@#1}%
6025 \newtoggle{Xgroupbylineseparetwolines@#1}%
6026
6027 \newtoggle{Xnumberonlyfirstinline@#1}%
6028 \newtoggle{Xnumberonlyfirstintwolines@#1}%
6029
6030 \newtoggle{Xlinenumannotationonlyfirst@#1}%
6031 \newtoggle{Xlinenumannotationonlyfirstintwo@#1}%
6032
6033 \csgdef{Xtwolines@#1}{}%
6034 \csgdef{Xmorethantwolines@#1}{}%
6035 \csgdef{Xsublinesep@#1}{\fullstop}%
6036 \csgdef{Xpagelinesep@#1}{\csname Xsublinesep@#1\endcsname}%for
backward compatibility, call Xsublinesep@#1
6037 \newtoggle{Xtwolinesbutnotmore@#1}%
6038 \newtoggle{Xtwolinesonlyinsamepage@#1}%
6039 \newtoggle{Xonlypstart@#1}%
6040 \newtoggle{Xpstarteverytime@#1}%
6041
6042 \newtoggle{Xpstart@#1}%
6043 \newtoggle{Xpstartonlyfirst@#1}%
6044 \csdef{Xprevpstart@#1}{0}%Not a hook, but a tool for the hook
Xpstartonlyfirst
6045 \csgdef{Xpstartseparator@#1}{}%
6046
6047 \newtoggle{Xstanza@#1}%
6048 \newtoggle{Xstanzaonlyfirst@#1}%
6049 \csdef{Xprevstanza@#1}{0}%Not a hook, but a tool for the hook
Xstanzaonlyfirst
6050
6051 \csgdef{Xstanzaseparator@#1}{}%
6052
6053 \csgdef{Xsymlinenum@#1}{}%
6054 \csgdef{Xsymlinenumannotation@#1}{}%
6055
6056 \newtoggle{Xnonote@#1}%
6057
6058 \newtoggle{Xnonumber@#1}%

```

```

6059 \newtoggle{Xnolinenumber@#1}%
6060 \newtoggle{Xnolinenumberifannotation@#1}%
6061 \csgdef{Xbeforenumber@#1}{0pt}%
6062 \csgdef{Xtxtbeforenumber@#1}{}%
6063 \newtoggle{Xnopagenumberifcurrent@#1}%
6064 \csgdef{Xbeforepagenumber@#1}{}%
6065 \csgdef{Xafternumber@#1}{0.5em}%
6066 \newtoggle{Xnonbreakableafternumber@#1}%
6067 \csgdef{Xbeforesymmlinum@#1}{\csuse{Xbeforenumber@#1}}%
6068 \csgdef{Xaftersymmlinum@#1}{\csuse{Xafternumber@#1}}%
6069 \csgdef{Xinplaceofnumber@#1}{1em}%
6070 \global\cslet{Xlemmaseparator@#1}{\rbracket}%
6071 \csgdef{Xbeforelemmaseparator@#1}{0em}%
6072 \csgdef{Xafterlemmaseparator@#1}{0.5em}%
6073 \csgdef{Xinplaceoflemmaseparator@#1}{1em}%
6074 \csgdef{Xbeforenotes@#1}{1.2em \@plus .6em \@minus .6em}%
6075 \csgdef{Xafterrule@#1}{0pt}
6076
6077 \csgdef{Xtxtbeforenotes@#1}{%
6078 \newtoggle{Xtxtbeforenotes@#1@typeset}}%Not directly used by user,
but internal
6079 \newtoggle{Xtxtbeforenotesonlyonce@#1}%
6080
6081 \csgdef{Xmaxhnotes@#1}{0.8\vsizer}
6082 \newtoggle{Xnoteswidthliketwocolumns@#1}%
6083 \csgdef{Xparafootsep@#1}{}%
6084 \csgdef{Xafternote@#1}{1em plus .4em minus .4em}%
6085 \csgdef{Xlinerangeseparator@#1}{\endashchar}%
6086
6087 \csgdef{Xlemmafont@#1}{}%
6088 \csgdef{Xwraplemma@#1}{%
6089 \csgdef{Xwidth@#1}{\hsizer}%
6090 \csgdef{Xwraplinenumannotation@#1}{\textsuperscript}%
6091 \csgdef{Xlinenumannotationposition@#1}{after}%
6092 \expandafter\csgdef{Xinnotemark@#1}##1{\ledinnotemark{##1}}%
6093 \newtoggle{Xnoidenticallinenumannotation@#1}%
6094 %

```

### XXI.3.2 Create inserts, needed to add notes in foot

As regards inserts, see chapter 15 of *The TeXbook* by D. Knuth.

```

6095 \expandafter\newinsert\csname #1footins\endcsname%
6096 \unless\ifnoledgroup%
6097 \expandafter\newinsert\csname mp#1footins\endcsname%
6098 \fi%
6099 %

```

### 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.

```

6100 \global\notbool{parapparatus@}{\expandafter\newcommand\expandafter
*}{\expandafter\newcommand}\csname #1footnote\endcsname [2] [] {%

```

```

6101     \xdef\@currentseries{#1}%
6102     \nottoggle{Xnonote@#1}{%
6103       \advance \edtext@notes by 1%
6104       \if@edtext@secondarg{%
6105         \ifledRcol%
6106         \ifcsstring{Xonlyside@#1}{L}{\
led@error@note@called@onrightside{#1footnote}}{}}%
6107       \else%
6108         \ifcsstring{Xonlyside@#1}{R}{\
led@error@note@called@onleftside{#1footnote}}{}}%
6109       \fi%
6110       \begingroup%
6111       \newcommand{\content}{##2}%
6112       \ifcsstring{series@display#1}{paragraph}{%
6113         \patchcmd{\@tag}{\par}{\relax}{}}%
6114       }{}%
6115       \ifnumberedpar{%
6116         \ifledRcol%
6117         \ifluatex%
6118           \footnotelang@lua[R]%
6119         \fi%
6120         \@ifundefined{xpg@main@language}{if polyglossia
6121           }{}%
6122           {\footnotelang@poly[R]}%
6123         \footnoteoptions@{R}{#1}{true}%
6124         \xright@appenditem{%
6125           \ifbool{indtl@innote}%
6126             {\unexpanded{\let\index\nindex}}%
6127             }{}%
6128           \ifbool{indtl@notenumber}%
6129             {\unexpanded{\let\index\nindex}}%There is no note
number... so
6130           }{}%
6131           \noexpand\Xnote@true%
6132           \unexpanded{\gdef\@currentnotetype{critical}}%
6133           \noexpand\prepare@Xprenotes{#1}%
6134           \noexpand\prepare@edindex@fornote{\l@d@nums}%
6135           \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.
6136           \noexpand\setcounter{stanzaR}{\the\c@stanzaR}%Save
stanzaR counter for footnote
6137           \unexpanded{\def\@this@crossref@start}{\theedtext:
start}%
6138           \unexpanded{\def\@this@crossref@end}{\theedtext:end}%
6139           \expandonce{\@beforeinsertofthisedtext}% Internal for
now, no reason to make it public
6140           \noexpand\parse@annot\l@current@annotR|%
6141           \noexpand\csuse{v#1footnote}{#1}%
6142           {{\l@d@nums}{\expandonce\@tag}{\expandonce\content}}
%
6143           \noexpand\Xnote@false%
6144           \unexpanded{\advance\@edindex@fornote@m@ne}%

```

```

6145     \ifbool{indtl@innote}%
6146       {\unexpanded{\let\index\orig@@index}}%
6147       {}%
6148     \ifbool{indtl@notenumber}%
6149       {\unexpanded{\let\index\orig@@index}}%
6150       {}%
6151     }\to\inserts@listR
6152     \footnoteoptions@{R}{##1}{false}%
6153     \global\advance\insert@countR \@one%
6154   \else%
6155     \ifluatex%
6156       \footnotelang@lua%
6157     \fi%
6158     \@ifundefined{xpg@main@language}%if polyglossia
6159       {}%
6160       {\footnotelang@poly}%
6161     \footnoteoptions@{L}{##1}{true}%
6162     \xright@appenditem{%
6163       \ifbool{indtl@innote}%
6164         {\unexpanded{\let\index\nindex}}%
6165         {}%
6166       \ifbool{indtl@notenumber}%
6167         {\unexpanded{\let\index\nindex}}%There is no note
number... so
6168         {}%
6169       \noexpand\Xnote@true%
6170       \unexpanded{\gdef\@currentnotetype{critical}}%
6171       \noexpand\prepare@Xprenotes{##1}%
6172       \noexpand\prepare@edindex@fornote{\l@d@nums}%
6173       \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.
6174       \ifl@dpairing%
6175         \noexpand\setcounter{stanzaL}{\the\c@stanzaL}%Save
stanzaR counter for footnote
6176       \fi%
6177       \unexpanded{\def\@this@crossref@start}{\theedtext :
start}%
6178       \unexpanded{\def\@this@crossref@end}{\theedtext :end}%
6179       \expandonce{\@beforeinsertofthisedtext}%Internal for
now, no reason to make it public
6180       \noexpand\parse@annot\l@current@annot|%
6181       \noexpand\csuse{v#1footnote}%
6182         {##1}%
6183         {\l@d@nums}{\expandonce\@tag}{\expandonce\content
}}%
6184       \noexpand\Xnote@false%
6185       \unexpanded{\advance\@edindex@fornote@m@one}%
6186       \ifbool{indtl@innote}%
6187         {\unexpanded{\let\index\orig@@index}}%
6188         {}%
6189       \ifbool{indtl@notenumber}%
6190         {\unexpanded{\let\index\orig@@index}}%

```



```

6191         {}}%
6192         }\to\inserts@list
6193         \global\advance\insert@count \@one%
6194         \footnoteoptions@{L}{##1}{false}%
6195     \fi
6196 \else
6197     \csuse{v#1footnote}{#1}{0|0|0|0|0|0|0|0}{##1}%
6198 \fi%
6199 \endgroup%
6200 \else%
6201     \led@err@FootnoteNotInSecondArgEdtext{#1}%
6202 \fi%
6203 }\@noneed@notestruer}% End of \nottoggle{Xnonote@#1}{%
6204 \ignorespaces%
6205 }
6206 %

```

Create counter used to determine on which page the previous note was called.

```

6207 \expandafter\newcount\csname #1prevpage@num\endcsname%
6208 \expandafter\newcount\csname #1prevpage@numR\endcsname%
6209 %

```

We need to be able to modify reledmac's footnote macros and restore their

```

6210 \global\csletcs{#1@@footnote}{#1footnote}
6211 %

```

### XXI.3.4 Set standard display

```

6212 \Xarrangement@normal{#1}%
6213 %

```

End of for critical footnotes.

```

6214 \fi
6215 %

```

## XXI.4 For familiar footnotes

Familiar footnotes are those which end with letters. We look for the `nofamiliar` option of `reledmac`.

```

6216 \unless\ifnofamiliar@
6217 %

```

### XXI.4.1 Options

```

6218 \newtoggle{parindentX@#1}
6219 \csgdef{wrapcontentX@#1}{}%
6220 \csgdef{hangindentX@#1}{Opt}%
6221 \csgdef{beforeinsertingX@#1}{}%
6222 \csgdef{raggedX@#1}{}%
6223 \csgdef{hsizetwocolX@#1}{0.45 \hsize}%
6224 \csgdef{hsizethreecolX@#1}{.3 \hsize}%
6225 \csgdef{colalignX@#1}{\raggedright}%

```

```

6226 \csgdef{notenunfontX@#1}{\normalfont}%
6227 \csgdef{notefontsizeX@#1}{\footnotesize}%
6228 \csgdef{bhooknoteX@#1}{}%
6229 \csgdef{bhookgroupX@#1}{}%
6230 \csgdef{afterruleX@#1}{Opt}
6231 \csgdef{beforenotesX@#1}{1.2em \@plus .6em \@minus .6em}
6232 \csgdef{maxhnotesX@#1}{0.8\vsizex}%
6233 \newtoggle{noteswidthliketwocolumnsX@#1}%
6234 \csgdef{parafootsepX@#1}{}%
6235 \csgdef{afternoteX@#1}{1em plus.4em minus.4em}
6236 \csgdef{widthX@#1}{\hsizex}%
6237 \expandafter\csgdef{innotemarkX@#1}##1{\ledinnotemark{##1}}%
6238 \csgdef{txtbeforenotesX@#1}{}%
6239 \newtoggle{nonoteX@#1}%
6240 \newtoggle{txtbeforenotesX@#1@typeset}%Not directly used by user,
but internal
6241 \newtoggle{txtbeforenotesonlyonceX@#1}%
6242 % End of for familiar footnotes.
6243 % \subsubsection{Create inserts, needed to add notes in foot}
6244 % As regards inserts, see chapter 15 of the TeXBook by D. Knuth.
6245 % \begin{macrocode}
6246 \expandafter\newinsert\csname footins#1\endcsname%
6247 \unless\ifnoledgroup@%
6248 \expandafter\newinsert\csname mpfootins#1\endcsname%
6249 \fi%
6250 %

```

#### 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.

```

6251 \global\expandafter\newcommand\csname footnote#1\endcsname [2] [] {%
6252 \xdef\@currentseries{#1}%
6253 \nottoggle{nonoteX@#1}{%
6254 \beginingroup%
6255 \prepare@prenotesX{#1}%
6256 \newcommand{\content}{##2}%
6257 %
6258 %

```

If we are using the `\csquotes` package, we reset the quotation level.

```

6259 \ifdefined\csqqlevel%
6260 \csqqlevel=0\relax%
6261 \fi%
6262 %

```

If we prepare parallel typesetting, we cannot just increase the footnote counter. Read `reledpar`'s handbook about that (V.1.2 p. 50). If we have a manual footnote mark, use it.

```

6263 \global\expandafter\advance\csname footnote#1@reading\
endcsname by \@ne%
6264 \ifstrempy{##1}{%
6265 \ifboolexpr{bool{!@dpairing} or bool{!@dprintingpages} or
bool{!@dprintingcolumns}}{%

```

```

6266     \ifcsdef{footnote#1reading\the\csname footnote#1@reading
\endcsname=typeset}%
6267     {\setcounter{footnote#1}{\csuse{footnote#1reading\the\
csname footnote#1@reading\endcsname=typeset}}}%
6268     {\setcounter{footnote#1}{\the\csname footnote#1@reading\
endcsname}}}%
6269     }{%
6270     \stepcounter{footnote#1}%
6271     }%
6272     }{}%
6273 %

```

We also have to check consistency with `\onlysideX` setting.

```

6274     \ifledRcol%
6275     \ifcsstring{onlysideX@#1}{L}{\
led@error@note@called@onrightside{footnote#1}}{}%
6276     \else%
6277     \ifcsstring{onlysideX@#1}{R}{\
led@error@note@called@onleftside{footnote#1}}{}%
6278     \fi%
6279 %

```

And now, the feature not depending on whether we prepare parallel typesetting

```

6280     \ifstrempy{##1}%
6281     {\protected@csxdef{@thefnmark#1}{\csuse{thefootnote#1}}}%
6282     {\protected@csxdef{@thefnmark#1}{##1}}%
6283     \protected@csxdef{hr@thefnmark#1}{\the\value{footnote#1}@the
\csname footnote#1@series\endcsname}%
6284     \nottoggle{nomk@}%Nomk is set to true when using \
footnoteXnomk with \parpackage
6285     {\csuse{@footnotemark#1}}%
6286     {}%
6287     \ifluatex%
6288     \xdef\footnote@luatextextdir{\the\textdir}%
6289     \xdef\footnote@luatexpardir{\the\pardir}%
6290     \fi%
6291     \if@ledgroup%
6292     \led@set@index@fornote{#1}%
6293     \fi%
6294     \csuse{vfootnote#1}{#1}{\expandonce\content}{##1}\
m@mmf@prepare%
6295     \ifbool{indtl@innote}%
6296     {\let\index\orig@@index}%
6297     {}%
6298     \ifbool{indtl@notenumber}%
6299     {\let\index\orig@@index}%
6300     {}%
6301     \endgroup%
6302     }{}%End of the notoggleX@#1
6303 }
6304 %

```

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. 50).

The  $\TeX$  counter `\footnoteX@series` is increased each time we reset the  $\TeX$  counter `footnoteX`, either using `+setcounter{footnoteX}{0}+` or because of `cs@addtoreset`.

```

6305     \newcounter{footnote#1}
6306     \global\expandafter\renewcommand\csname thefootnote#1\endcsname{\
arabic{footnote#1}}
6307     \expandafter\newcount\csname footnote#1@reading\endcsname%
6308     \expandafter\newcount\csname footnote#1@series\endcsname%
6309     \global\expandafter\csname footnote#1@series\endcsname=1%
6310 %

```

Create counter used to determine on which page the previous note was called.

```

6311     \expandafter\newcount\csname prevpage#1@num\endcsname%
6312     \expandafter\newcount\csname prevpage#1@numR\endcsname%
6313     \expandafter\newcount\csname footnote#1@parafootsepX\endcsname%For
the case we use \parafootsepX outside of numbered section
6314 %

```

Add `\let\footnoteX@gobble to \no@expands`.

```

6315     \expandafter\gappto\expandafter\no@expands\expandafter{\expandafter\
let\csname footnote#1\endcsname@gobble}%
6316 %

```

And now, define `\footnoteXmark` and `\footnoteXtext`, equivalent to classical `\footnotemark` and `\footnotetext`.

```

6317     \expandafter\newcommand\csname footnote#1mark\endcsname [1] [] {%
6318     \begingroup%
6319     \prepare@prenotesX{#1}%
6320     \ifstrempy{##1}{%
6321     \stepcounter{footnote#1}%
6322     }{%
6323     \setcounter{footnote#1}{##1}%
6324     }%
6325     \protected@csxdef{@thefnmark#1}{\csuse{thefootnote#1}}%
6326     \protected@csxdef{hr@thefnmark#1}{\the\value{footnote#1}@the\
csname footnote#1@series\endcsname}%
6327     \csuse{@footnotemark#1}%
6328     \m@mmf@prepare%
6329     \endgroup%
6330     }%
6331     \expandafter\newcommand\csname footnote#1text\endcsname [2] [] {%
6332     \begingroup%
6333     \ifluatex%
6334     \xdef\footnote@luatextextdir{\the\textdir}%
6335     \xdef\footnote@luatexpardir{\the\pardir}%
6336     \fi%
6337     \csuse{vfootnote#1}{#1}{\expandonce{##2}}{##1}%
6338     \endgroup%
6339     }%
6340 %

```

Do not forget to initialize the series.

```

6341 \arrangementX@normal{#1}%
6342 \fi
6343 %

```

## XXI.5 The endnotes

Endnotes are commands like `\Xendnote`, where `X` is a series letter. First, we check for the `noend` options.

```

6344 \unless\ifnoend@
6345 %

```

### XXI.5.1 The auxiliary file

```

\l@d@Xend Endnotes of all varieties are saved up in a file, one by series, typically named  $\langle jobname \rangle$ .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
6346 \expandafter\newwrite\csname l@d@#1end\endcsname%
6347 \expandafter\newif\csname ifl@dend@#1\endcsname%
6348 %

```

### 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.

```

6349
6350 \global\expandafter\newcommand\csname #1endnote\endcsname[2][1,
usedefault]{%
6351 \advance \edtext@notes by 1%
6352 \bgroup%
6353 \newlinechar='40%
6354 \newcommand{\content}{##2}%
6355 \stepcounter{labidx}%
6356 \expandafter\immediate\expandafter\write\csname l@d@#1end\
endcsname{%
6357 \unexpanded{\def\sw@list@inedtext}%
6358 {\expandafter\unexpanded\expandafter{\sw@inthisedtext}}%
6359 \@percentchar\space%Explicit space, to add a linebreak in the
output file
6360 \noexpand\parse@annot\l@current@annot|\@percentchar\space%
6361 \expandafter\string\csname #1end\endcsname%
6362 {\ifnumberedpar@l@d@nums\fi}%
6363 {\ifnumberedpar@\expandonce@tag\fi}%
6364 {\expandonce\content}%
6365 {#1}%
6366 {\unexpanded{##1}}%
6367 {\ifledRcol R\else L\fi}%
6368 {\theedtext}%
6369 \@percentchar%

```

```

6370     }%
6371     \egroup%
6372     \ignorespaces%
6373 }%
6374 %

```

### XXI.5.3 Tools

The `\Xtoendnotes` command inserts any arbitrary content into the endnote file. It is an alias of the more generalist `\addtoendnotes`

```

6375
6376 \global\expandafter\newcommand\csname #1toendnotes\endcsname[1]{%
6377 \toendnotes[#1]{##1}%
6378 }%
6379
6380 \expandafter\WithSuffix\expandafter\newcommand\csname #1toendnotes\
6381 endcsname*[1]{%
6382 \toendnotes*[#1]{##1}%
6383 }%
6384 %

```

### 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.

```

6385
6386 \global\cslet{#1end}{\@gobbleseven}
6387 %

```

We need to store the number of times `\doendnotesbysection` is called for one series.

```

6388 \global\expandafter\newcount\csname #1end@bysection\endcsname%
6389 %

```

### XXI.5.5 The options

```

6390 \csgdef{Xendwraplemma@#1}{%
6391 \csgdef{Xendwrapcontent@#1}{}%
6392 \csgdef{Xendtwolines@#1}{}%
6393 \csgdef{Xendmorethantwolines@#1}{}%
6394 \newtoggle{Xendtwolinesbutnotmore@#1}{}%
6395 \newtoggle{Xendtwolinesonlyinsamepage@#1}{}%
6396 \newtoggle{Xendlemmadisablefontselection@#1}{}%
6397 \csgdef{Xendnotenumberfont@#1}{\normalfont}%
6398 \csgdef{Xendnotefontsize@#1}{\footnotesize}%
6399 \csgdef{Xendbhooknote@#1}{}%
6400

```

```

6401 \csgdef{Xendsublinesep@#1}{\fullstop}%
6402
6403 \csgdef{Xendbeforenumber@#1}{Opt}
6404 \csgdef{Xendafternumber@#1}{0.5em}
6405
6406 \csgdef{Xendboxlinenum@#1}{Opt}%
6407 \csgdef{Xendboxlinenumalign@#1}{L}%
6408
6409 \csgdef{Xendboxstartlinenum@#1}{Opt}%
6410 \newtoggle{Xendnotboxingsubline@#1}%
6411 \csgdef{Xendboxendlinenum@#1}{Opt}%
6412
6413 \csgdef{Xendlemmaseparator@#1}{}%
6414 \csgdef{Xendbeforelemmaseparator@#1}{0em}%
6415 \csgdef{Xendafterlemmaseparator@#1}{0.5em}%
6416 \csgdef{Xendinplaceoflemmaseparator@#1}{0.5em}%
6417
6418 \newtoggle{Xendparagraph@#1}%
6419 \csgdef{Xendafternote@#1}{1em plus.4em minus.4em}%
6420 \csgdef{Xendsep@#1}{}%
6421
6422 \csgdef{Xendinplaceofnumber@#1}{Opt}%
6423 \newtoggle{Xendnonumber@#1}%
6424 \newtoggle{Xendnolinenum@#1}%
6425 \newtoggle{Xendnolinenumannotation@#1}%
6426
6427 \csgdef{Xendhangindent@#1}{Opt}%
6428
6429 \newtoggle{Xendnumberonlyfirstinline@#1}%
6430 \newtoggle{Xendnumberonlyfirstintwolines@#1}%
6431
6432 \newtoggle{Xendlinenumannotationonlyfirst@#1}%
6433 \newtoggle{Xendlinenumannotationonlyfirstintwo@#1}%
6434
6435 \csgdef{Xendbeforesymlinenum@#1}{\csuse{Xendbeforenumber@#1}}%
6436 \csgdef{Xendaftersymlinenum@#1}{\csuse{Xendafternumber@#1}}%
6437
6438 \csgdef{Xendsymlinenum@#1}{}%
6439 \csgdef{Xendsymlinenumannotation@#1}{}%
6440
6441 \csgdef{Xendboxsymlinenum@#1}{Opt}%
6442
6443 \csgdef{Xendbhooklinenum@#1}{}%
6444 \csgdef{Xendehooklinenum@#1}{}%
6445 \csgdef{Xendbhookinplaceofnumber@#1}{}%
6446 \csgdef{Xendehookinplaceofnumber@#1}{}%
6447
6448 \csgdef{Xendlinerangeseparator@#1}{\endashchar}%
6449
6450 \csgdef{Xendbeforepagenumber@#1}{p.}%
6451 \csgdef{Xendafterpagenumber@#1}{) }%
6452 \csgdef{Xendlineprefixsingle@#1}{}%
6453 \csgdef{Xendlineprefixmore@#1}{}%
6454

```

```

6455 \newtoggle{Xendlineflag@#1}
6456
6457 \csgdef{Xendlemmafont@#1}{}%
6458
6459 \csgdef{Xendlinenumannotationposition@#1}{after}%
6460 \csgdef{Xendwraplinenumannotation@#1}{\textsuperscript}%
6461 \newtoggle{Xendnoidenticallinenumannotation@#1}%
6462
6463 \newtoggle{Xendpagenumberonlyfirst@#1}%
6464 \newtoggle{Xendpagenumberonlyfirstiftsingle@#1}%
6465 \newtoggle{Xendpagenumberonlyfirstinttwo@#1}%
6466 \csgdef{Xendsympagenum@#1}{}%
6467 \csgdef{Xendinplaceofpagenumber@#1}{Opt}%
6468
6469 \csgdef{Xendtxtbeforenotes@#1}{}%
6470 \newtoggle{Xendfirstnote@#1}%Not a hook, but used to apply
Xendtxtbeforenotes
6471
6472 %

```

End of endnotes declaration

```

6473 \fi%
6474 %

```

Dump series in \@series

```

6475 \listxadd{\@series}{#1}
6476 }
6477 }% End of \newseries
6478 %

```

## XXI.6 Init standards series (A,B,C,D,E)

```

6479 \expandafter\newseries\expandafter{\default@series}
6480 %

```

## 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.

```

6481 \newcommand{\seriesatbegin}[1]{%
6482 \StrDel{\@series}{#1}[\@series]%
6483 \edef\@new{}%
6484 \listead{\@new}{#1}%
6485 \listead{\@new}{\@series}%
6486 \xdef\@series{\@new}%
6487 }
6488 %

```

`\seriesatend` And `\seriesatend` moves the series to the end of the list.



```

6489 \newcommand{\seriesatend}[1]{%
6490   \StrDel{\@series}{#1}[\@series]%
6491   \edef\@new{%
6492     \listead{\@new}{\@series}%
6493     \listead{\@new}{#1}%
6494     \xdef\@series{\@new}%
6495   }
6496 %

```

## XXII.2 Test series order

`\ifseriesbefore` `\ifseriesbefore{<seriesA>}{<seriesB>}{<true>}{<false>}` expands to `<true>` if `<seriesA>` is printed before `<seriesB>`, or to `<false>` otherwise.

```

6497 \newcommand{\ifseriesbefore}[4]{%
6498   \StrPosition{\@series}{#1}[\@first]%
6499   \StrPosition{\@series}{#2}[\@second]%
6500   \ifnumgreater{\@second}{\@first}{#3}{#4}%
6501 }
6502 %

```

### XXII.2.1 Get the first series

In some specific case, we need to know the first series of the list of series.

```

\@getfirstseries03 \newcommand{\@getfirstseries}{%
6504   \ifdefempty{\@series}%
6505     {\xdef\@firstseries{}}%
6506     {\StrChar{\@series}{1}[\@firstseries]}%
6507 }%
6508 %

```

## 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`.

```

6509 \newcommandx{\settoggle@series}[5][4,5,usedefault]{%
6510   \def\do##1{%
6511     \ifcsdef{etb@tgl@#2@##1}{%
6512       \global\settoggle{#2@##1}{#3}%
6513     }{%
6514       \led@warn@BadHookSeries{##1}{#2}%
6515     }%
6516     \ifstrequal{#4}{critical}{
6517       \cuse{Xarrangement@\cuse{series@display##1}}{##1}%
6518     }{}
6519     \ifstrequal{#4}{familiar}{%
6520       \cuse{arrangementX@\cuse{series@displayX##1}}{##1}%
6521     }{}%
6522   }%
6523   \ifstrempy{#1}{%
6524     \dolistloop{\@series}%
6525     \ifstrempy{#5}{%
6526       \docsvlist{#5}%
6527     }
6528   }%
6529   {%
6530     \docsvlist{#1}%
6531   }%
6532 }
6533 %

```

`\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.

```

6534 \newcommandx{\setcommand@series}[6][4,5,6=0,usedefault]{%
6535   \def\do##1{
6536     \ifcsdef{#2@##1}{%
6537       \expandafter\global\expandafter\renewcommand\csname#2@##1\endcsname
6538       [#6]{#3}%
6539     }{%
6540       \led@warn@BadHookSeries{##1}{#2}%
6541     }%
6542     \ifstrequal{#4}{critical}{%
6543       \csuse{Xarrangement@\csuse{series@display##1}}{##1}%
6544     }{}
6545     \ifstrequal{#4}{familiar}{%
6546       \csuse{arrangementX@\csuse{series@displayX##1}}{##1}%
6547     }{}%
6548     \ifstreempty{#1}{%
6549       \dolistloop{\@series}%
6550       \ifstreempty{#5}{%
6551         \docsvlist{#5}
6552       }
6553     }%
6554     {%
6555       \docsvlist{#1}%
6556     }%
6557 }%
6558 %

```

### XXII.3.3 Tools to generate options commands

`\newhookcommand@series` `\newhookcommand@series\command` names is a generic command to add new commands for hooks, like `\Xhsizetwocol`. The first argument is the name of the hook, the second a comma-separated list of pseudo-series where the hook can be used, like `appref` in the case of `\Xtwolines`. The second argument is also used to create commands named `\<hookname><pseudoserries>`, like `\Xtwolinesappref`. The third argument

```

6559 \newcommandx{\newhookcommand@series}[3][2,3=0,usedefault]{%
6560   \global\expandafter\newcommand\expandafter*\csname #1\endcsname[2][%
6561     \setcommand@series{##1}{#1}{##2}[][#2][#3]%
6562   }%
6563   \ifstreempty{#2}{%
6564     \def\do##1{%
6565       \global\expandafter\newcommand\expandafter*\csname #1##1\endcsname
6566       [1]{%
6567         \csuse{#1}[##1]{###1}%
6568       }%
6569     }%
6570     \docsvlist{#2}%
6571   }%

```

```

6571 }
6572 %

\newhooktoggle@series \newhooktoggle@series\command names is a generic command to add new com-
mands for a new toggle hook, like \Xnumberonlyfirstinline. The second argu-
ment is also used to create commands named \<hookname><pseudoserries>, like
\Xtwolinesbutnotmoreappref.

6573 \newcommandx{\newhooktoggle@series}[2][2,usedefault]{%
6574   \global\expandafter\newcommandx\expandafter*\csname #1\endcsname[2][1,2={
true},usedefault]{%
6575     \settoggle@series{##1}{#1}{##2}[][#2]%
6576   }%
6577   \ifstrempy{#2}{}{%
6578     \def\do##1{%
6579       \global\expandafter\newcommand\expandafter*\csname #1##1\endcsname{%
6580         \cuse{#1}[##1]%
6581       }%
6582     }%
6583     \docsvlist{#2}%
6584   }%
6585 }
6586 %

```

`\newhooktoggle@series@reload` `\newhookcommand@toggle@reload` does the same thing as `\newhooktoggle@series` but the commands created by this macro also reload the series arrangement, depending on the type of notes

```

6587 \newcommand{\newhooktoggle@series@reload}[2]{%
6588   \global\expandafter\newcommandx\expandafter*\csname #1\endcsname[2][1,2={
true},usedefault]{%
6589     \settoggle@series{##1}{#1}{##2}[#2]%
6590   }%
6591 }%
6592 %

```

`\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.

```

6593 \newcommand{\newhookcommand@series@reload}[2]{%
6594   \global\expandafter\newcommand\expandafter*\csname #1\endcsname[2][1]{%
6595     \setcommand@series{##1}{#1}{##2}[#2]%
6596   }%
6597 }
6598 %

```

### 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.

```

6599 \unless\ifnocritical@
6600 \newhookcommand@series{Xwrapcontent}%
6601 \newhookcommand@series{Xbeforeinserting}%
6602 \newhookcommand@series{Xlemmafонт}%
6603 \newhookcommand@series{Xwraplemma}%
6604 \newhooktoggle@series{Xparindent}
6605 \newhookcommand@series{Xhangindent}
6606 \newhookcommand@series{Xragged}
6607 \newhookcommand@series{Xhsizetwocol}
6608 \newhookcommand@series{Xhsizethreecol}
6609 \newhookcommand@series{Xcolalign}%
6610 \newhookcommand@series{Xnotenumberfont}
6611 \newhookcommand@series{Xbhooknote}
6612 \newhookcommand@series@reload{Xbhookgroup}{critical}
6613 \newhookcommand@series{Xboxsymlinenum}%
6614 \newhookcommand@series{Xsymlinenum}
6615 \newhookcommand@series{Xsymlinenumannotation}%
6616 \newhookcommand@series{Xbeforenumber}
6617 \newhookcommand@series{Xtxtbeforenumber}
6618 \newhooktoggle@series{Xnopagenumberifcurrent}%
6619 \newhookcommand@series{Xbeforepagenumber}%
6620 \newhookcommand@series{Xafternumber}
6621 \newhookcommand@series{Xbeforesymlinenum}
6622 \newhookcommand@series{Xaftersymlinenum}
6623 \newhookcommand@series{Xinplaceofnumber}
6624 \newhookcommand@series{Xlemmaseparator}
6625 \newhookcommand@series{Xbeforelemmaseparator}
6626 \newhookcommand@series{Xafterlemmaseparator}
6627 \newhookcommand@series{Xinplaceoflemmaseparator}
6628 \newhookcommand@series{Xtxtbeforenotes}
6629 \newhooktoggle@series{Xtxtbeforenotesonlyonce}%
6630 \newhookcommand@series@reload{Xafterrule}{critical}
6631 \newhooktoggle@series{Xnumberonlyfirstinline}
6632 \newhooktoggle@series{Xnumberonlyfirstintwolines}
6633 \newhooktoggle@series{Xlinenumannotationonlyfirst}%
6634 \newhooktoggle@series{Xlinenumannotationonlyfirstintwo}%
6635 \newhooktoggle@series{Xgroupbyline}%
6636 \newhooktoggle@series{Xgroupbylineseparetwolines}%
6637 \newhooktoggle@series{Xnonumber}
6638 \newhooktoggle@series{Xnolinenum}%
6639 \newhooktoggle@series{Xnolinenumifannotation}%
6640
6641 \newhooktoggle@series{Xpstart}
6642 \newhooktoggle@series{Xpstartonlyfirst}%
6643 \newhooktoggle@series{Xpstarteverytime}%
6644 \newhookcommand@series{Xpstartseparator}%
6645
6646 \newhooktoggle@series{Xstanza}%
6647 \newhooktoggle@series{Xstanzaonlyfirst}%
6648 \newhookcommand@series{Xstanzaseparator}%
6649
6650 \newhooktoggle@series{Xnonote}%
6651
6652 \newhooktoggle@series{Xonlypstart}

```

```

6653 \newhooktoggle@series{Xnonbreakableafternumber}
6654 \newhooktoggle@series{Xlemmadisablefontselection}
6655 \newhookcommand@series@reload{Xmaxhnotes}{critical}
6656 \newhookcommand@series@reload{Xbeforenotes}{critical}
6657 \newhooktoggle@series@reload{Xnoteswidthliketwocolumns}{critical}%
6658 \newhookcommand@series@reload{Xnotefontsize}{critical}
6659
6660 \newhookcommand@series{Xboxlinenum}%
6661 \newhookcommand@series{Xboxlinenumalign}%
6662
6663 \newhookcommand@series{Xboxstartlinenum}%
6664 \newhooktoggle@series{Xnotboxingsubline}%
6665 \newhookcommand@series{Xboxendlinenum}%
6666
6667 \newhookcommand@series{Xafternote}%
6668 \newhookcommand@series{Xparafootsep}
6669 \newhookcommand@series@reload{Xwidth}{critical}%
6670
6671 \newhookcommand@series{Xinnotemark}[] [1]%
6672
6673 \ifundef{\Xhsize}%
6674 {%
6675   \newcommandx{\Xhsize}[2][1,usedefault]{%
6676     \led@warning@Xhsize@deprecated%
6677     \Xwidth[#1]{#2}%
6678   }%
6679 }%
6680 {}%
6681 \fi
6682 %

```

Now, hooks for critical notes which also apply to crossreferencing and line numbering at the sides of the page.

```

6683 \newhooktoggle@series{Xlineflag}[appref,SEref]
6684 \newhookcommand@series{Xtwolines}[appref,SEref]
6685 \newhookcommand@series{Xmorethantwolines}[appref,SEref]
6686 \newhookcommand@series{Xsublinesep}[appref,SEref,side]%
6687 \newhookcommand@series{Xpagelinesep}[appref,SEref,side]%
6688 \newhooktoggle@series{Xtwolinesbutnotmore}[appref,SEref]
6689 \newhooktoggle@series{Xtwolinesonlyinsamepage}[appref,SEref]
6690 \newhookcommand@series{Xlinerangeseparator}[appref,SEref]
6691 \newhookcommand@series{Xlinenumannotationposition}[side,appref,SEref]%
6692 \newhookcommand@series{Xwraplinenumannotation}[side,ref,appref,SEref]%
6693 \newhooktoggle@series{Xnoidenticallinenumannotation}[side,ref,appref,SEref]
6694 %

```

### XXII.3.5 Options for familiar notes

Before generating the optional commands for familiar notes, we check the `\nofamiliar` option.

```

6695 \unless\ifnofamiliar@
6696 \newhookcommand@series{wrapcontentX}%

```

```

6697 \newhookcommand@series{beforeinsertingX}%
6698 \newhooktoggle@series{parindentX}
6699 \newhookcommand@series{hangindentX}
6700 \newhookcommand@series{raggedX}
6701 \newhookcommand@series{hsizetwocolX}
6702 \newhookcommand@series{hsizethreecolX}
6703 \newhookcommand@series{colalignX}%
6704 \newhookcommand@series{notenumfontX}
6705 \newhookcommand@series{bhooknoteX}
6706 \newhookcommand@series@reload{bhookgroupX}{familiar}
6707 \newhookcommand@series@reload{beforenotesX}{familiar}
6708 \newhookcommand@series@reload{maxhnotesX}{familiar}
6709 \newhooktoggle@series@reload{noteswidthliketwocolumnsX}{familiar}%
6710 \newhookcommand@series@reload{afterruleX}{familiar}
6711 \newhookcommand@series@reload{notefontsizeX}{familiar}
6712 \newhookcommand@series{afternoteX}
6713 \newhookcommand@series{parafootsepX}
6714 \newhookcommand@series{txtbeforenotesX}%
6715 \newhooktoggle@series{txtbeforenotesonlyonceX}%
6716 \newhookcommand@series@reload{widthX}{familiar}%
6717 \newhookcommand@series{innotemarkX}[] [1]%
6718 \newhooktoggle@series{nonoteX}%
6719 \ifundef{\hsizeX}%
6720   {%
6721     \newcommandx{\hsizeX}[2] [1,usedefault]{%
6722       \led@warning@hsizeX@deprecated%
6723       \widthX[#1]{#2}%
6724     }%
6725   }%
6726 {}%
6727 \fi
6728 %

```

### 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.

```

6729 \unless\ifnoend@
6730 \newhookcommand@series{Xendwraplemma}
6731 \newhookcommand@series{Xendwrapcontent}
6732 \newhookcommand@series{Xendnotenumfont}
6733 \newhookcommand@series{Xendlemmafont}%
6734 \newhookcommand@series{Xendbhooknote}
6735
6736 \newhookcommand@series{Xendboxlinenum}%
6737 \newhookcommand@series{Xendboxlinenumalign}%
6738
6739 \newhookcommand@series{Xendboxstartlinenum}%
6740 \newhooktoggle@series{Xendnotboxingsubline}%
6741 \newhookcommand@series{Xendboxendlinenum}%
6742
6743 \newhookcommand@series{Xendnotefontsize}
6744 \newhooktoggle@series{Xendlemmadisablefontselection}

```

```

6745 \newhookcommand@series{Xendlemmaseparator}
6746 \newhookcommand@series{Xendbeforelemmaseparator}
6747 \newhookcommand@series{Xendafterlemmaseparator}
6748 \newhookcommand@series{Xendinplaceoflemmaseparator}
6749
6750 \newhookcommand@series{Xendbeforenumber}%
6751 \newhookcommand@series{Xendafternumber}%
6752
6753 \newhooktoggle@series{Xendparagraph}
6754 \newhookcommand@series{Xendafternote}
6755 \newhookcommand@series{Xendsep}
6756
6757 \newhookcommand@series{Xendinplaceofnumber}%
6758 \newhooktoggle@series{Xendnonumber}%
6759 \newhooktoggle@series{Xendnolinenumber}%
6760 \newhooktoggle@series{Xendnolinenumberifannotation}%
6761
6762 \newhooktoggle@series{Xendnumberonlyfirstinline}%
6763 \newhooktoggle@series{Xendnumberonlyfirstintwolines}%
6764
6765 \newhooktoggle@series{Xendlinenumannotationonlyfirst}%
6766 \newhooktoggle@series{Xendlinenumannotationonlyfirstintwo}%
6767
6768 \newhookcommand@series{Xendsymmlinenum}%
6769 \newhookcommand@series{Xendbeforemsymmlinenum}%
6770 \newhookcommand@series{Xendaftersymmlinenum}%
6771 \newhookcommand@series{Xendboxsymmlinenum}%
6772
6773 \newhookcommand@series{Xendsymmlinenumannotation}%
6774
6775 \newhookcommand@series{Xendbhooklinenumber}%
6776 \newhookcommand@series{Xendahooklinenumber}%
6777 \newhookcommand@series{Xendbhookinplaceofnumber}%
6778 \newhookcommand@series{Xendahookinplaceofnumber}%
6779
6780 \newhookcommand@series{Xendhangindent}%
6781
6782 \newhooktoggle@series{Xendpagenumberonlyfirst}%
6783 \newhooktoggle@series{Xendpagenumberonlyfirstifsingle}%
6784 \newhooktoggle@series{Xendpagenumberonlyfirstintwo}%
6785 \newhookcommand@series{Xendsympagenum}%
6786 \newhookcommand@series{Xendinplaceofpagenumber}%
6787
6788 \newhookcommand@series{Xendtxtbeforenotes}%
6789
6790 \fi
6791 \newhooktoggle@series{Xendlineflag}[apprefwithpage,SErefwithpage]
6792 \newhookcommand@series{Xendtwolines}[apprefwithpage,SErefwithpage]
6793 \newhookcommand@series{Xendmoreethantwolines}[apprefwithpage,SErefwithpage]
6794 \newhooktoggle@series{Xendtwolinesbutnotmore}[apprefwithpage,SErefwithpage]
6795 \newhooktoggle@series{Xendtwolinesonlyinsamepage}[apprefwithpage,
SErefwithpage]
6796 \newhookcommand@series{Xendlinerangeseparator}[apprefwithpage,SErefwithpage]
]

```



```

6797 \newhookcommand@series{Xendbeforepagenumber}[apprefwithpage,SErefwithpage,
SErefonlypage]
6798 \newhookcommand@series{Xendafterpagenumber}[apprefwithpage,SErefwithpage]
6799 \newhookcommand@series{Xendlineprefixsingle}[apprefwithpage,SErefwithpage]
6800 \newhookcommand@series{Xendlineprefixmore}[apprefwithpage,SErefwithpage]
6801 \newhookcommand@series{Xendsublinesep}[apprefwithpage,SErefwithpage]
6802
6803 \newhookcommand@series{Xendlinenumannotationposition}[apprefwithpage,
SErefwithpage]%
6804 \newhookcommand@series{Xendwraplinenumannotation}[apprefwithpage,
SErefwithpage]%
6805 \newhooktoggle@series{Xendnoidenticallinenumannotation}[apprefwithpage,
SErefwithpage]%
6806 %

```

## XXII.4 Hooks for a particular footnote

`\newhooktoggle@specific` `\newhooktoggle@specific` is a generic command to create boolean hook specific to a note.

```

6807 \newcommand{\newhooktoggle@specific}[1]{%
6808   \newtoggle{#1@}%
6809   \listgadd{\hooktoggle@specific}{#1}%
6810   \define@key[mac]{truefootnoteoption}{#1}[]{\global\settoggle{#1@}{true}}%
When enabling footnote option
6811   \define@key[mac]{falsefootnoteoption}{#1}[]{\global\settoggle{#1@}{false
}}
6812 }
6813 %

```

`\newhookarg@specific` `\newhookarg@specific` is a generic command to create argumen hook specific to a note.

```

6814 \newcommand{\newhookarg@specific}[1]{%
6815   \listgadd{\hookarg@specific}{#1}%
6816   \define@key[mac]{truefootnoteoption}{#1}{\csgdef{#1@}{##1}}%When enabling
footnote option
6817   \define@key[mac]{falsefootnoteoption}{#1}{\global\csundef{#1@}}%When
disabling footnote option
6818 }
6819 %

```

`\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.

```

6820 \def\hooktoggle@specific{}%
6821 \newcommand{\add@hooktoggle@specific@to@cs}[1]{%
6822   \def\do##1{%
6823     \iftoggle{##1@}{%
6824       \ifcsdef{#1}{%
6825         \csgappto{#1}{\toggletrue{##1@}}%
6826       }{%

```

```

6827 \csgdef{#1}{\toggletrue{##1@}}%
6828 }%
6829 }{%
6830 \ifcsdef{#1}{%
6831 \csgappto{#1}{\togglefalse{##1@}}%
6832 }{%
6833 \csgdef{#1}{\togglefalse{##1@}}%
6834 }%
6835 }%
6836 }%
6837 \dolistloop{\hooktoggle@specific}%
6838 }%
6839 %

```

`\hookarg@specific` The same, but for optional argument of critical footnotes with assigned value.  
`\add@hookarg@specific@to@cs`

```

6840 \def\hookarg@specific{}%
6841 \newcommand{\add@hookarg@specific@to@cs}[1]{%
6842 \def\do##1{%
6843 \ifcsvoid{##1@}{%
6844 \ifcsdef{#1}{%
6845 \csxappto{#1}{%
6846 \noexpand\csdef%
6847 {##1@}{\csname##1@\endcsname}%
6848 }%
6849 }{%
6850 \csxdef{#1}{%
6851 \noexpand\csdef%
6852 {##1@}{\csname##1@\endcsname}%
6853 }%
6854 }%
6855 }%
6856 }%
6857 \dolistloop{\hookarg@specific}%
6858 }%
6859 %

```

And now, we define some hooks specific to a note.

```

6860 \newhooktoggle@specific{fulllines}%
6861 \newhooktoggle@specific{nonum}
6862 \newhooktoggle@specific{nosep}
6863 \newhooktoggle@specific{noprefix}%
6864 \newhooktoggle@specific{prefixmore}%
6865 \newhookarg@specific{linrangesep}
6866 %

```

`linrangesep@` `\linrangesep@` is defined by the option `linrangesep` of critical notes to change temporarily the line range separator for a specific line. As we have to define it before typesetting the line and undefine it after, we use the family of `xkeyval` package's key.

```

6867 %

```

`\nomk@` `\nomk@` toggle is used by `reledpar` to remove the footnote mark in the text when using `\footnoteXmk`. Read `reledpar` handbook.

```
6868 \newtoggle{nomk@}%
6869 %
```

## XXII.5 Alias

`\Xnolemmaseparator` `\Xnolemmaseparator` [*series*] is just an alias for `\Xlemmaseparator` [*series*] {}.

```
6870 \newcommand*{\Xnolemmaseparator}[1][1]{\Xlemmaseparator[#1]{} }
6871 %
```

## 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  $\text{\LaTeX}$ '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  $\text{\LaTeX}$  are first, then familiar familiar footnotes and finally the critical footnotes.

```
6872 \newcommand*{\l@ddoxtrafeet}{%
6873   \global\togglefalse{firstfootnoteruleX@printed}%
6874   \global\togglefalse{Xfirstfootnoterule@printed}%
6875   \IfStrEq{familiar-critical}{\@fnpos}
6876     {\do@feetX\do@Xfeet}%
6877     {%
6878     \IfStrEq{critical-familiar}{\@fnpos}%
6879       {\do@Xfeet\do@feetX}%
6880       {%
6881         \setbox\@outputbox \vbox{%
6882           \unvbox\@outputbox%
6883           \do@feet@custom@order}{\@fnpos}%
6884         }%
6885       }%
6886     }%
6887 }%
6888
6889 %
```

`\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,  $\text{\LaTeX}$  could perform an infinite run.

```

6890 \newcommand{\do@feet@custom@order}[2]{%
6891   \def\do##1{%
6892     \edef\@notesseries{\@firstoftwo##1}%
6893     \edef\@notetype{\@secondoftwo##1}%
6894     \ifdefstring{\@notetype}{critical}%
6895       {\csuse{#1append@Xnotes}{\@notesseries}}%
6896       {\ifdefstring{\@notetype}{familiar}%
6897         {\csuse{#1append@notesX}{\@notesseries}}%
6898         {}}%
6899     }%
6900   }%
6901   \expandafter\docsvlist\expandafter{#2}%
6902 }%
6903 %

```

`\do@Xfeet` `\do@Xfeet` is the code extending `\@makecol` to cater to the extra critical feet.

```

6904 \newcommand*{\do@Xfeet}{%
6905   \setbox\@outputbox \vbox{%
6906     \unvbox\@outputbox
6907     \@opXfeet}}
6908 %

```

`\@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
6909 \newcommand{\append@Xnotes}[1]{%
6910   \ifvoid\csuse{#1footins}\else%
6911     \global\skip\csuse{#1footins}=\csuse{Xbeforenotes@#1}%
6912     \global\advance\skip\csuse{#1footins} by\csuse{Xafterrule@#1}%
6913     \print@Xnotes{#1}%
6914     \fi%
6915 }%
6916 %

```

The normal way to add one series, `\print@Xnotes`, is replaced by `reledpar` when using `\Pages`.

```

6917 \newcommand\print@Xnotes[1]{%
6918   \xdef\@currentseries{#1}%
6919   \csuse{#1footstart}{#1}%
6920   \csuse{#1footgroup}{#1}%%
6921 }%
6922 %

```

We print all series of notes by looping on them. We check before printing them that they are not voided.

```

6923 \newcommand*{\@opXfeet}{%
6924   \unless\ifnocritical@%
6925     \gdef\firstXseries@{}%
6926     \def\do##1{%
6927       \append@Xnotes{##1}%
6928     }%
6929     \dolistloop{\@series}%
6930   \fi%

```

```
6931 }%
6932 %
```

`\l@ddodoreintrafeet` `\l@ddodoreintrafeet` is the code catering for the extra footnotes within `\@reinserts`. We use the same category and ordering as in `\l@ddoxtrafeet`.

```
6933 \newcommand*\l@ddodoreintrafeet}{%
6934   \IfStrEq{familiar-critical}{\@fnpos}
6935     {\@doreinfeetX\X@doreinfeet}%
6936     {%
6937     \IfStrEq{critical-familiar}{\@fnpos}%
6938       {\X@doreinfeet\@doreinfeetX}%
6939       {\@doreinfeetX\X@doreinfeet}%
6940     }%
6941 }
6942
6943 %
```

`\X@doreinfeet` `\X@doreinfeet` is the code for catering for the extra critical footnotes within `\@reinserts`.

```
6944 \newcommand*\X@doreinfeet}{%
6945   \unless\ifnocritical@%
6946   \def\do##1{%
6947     \ifvoid\csuse{##1footins}\else%
6948       \insert\csuse{##1footins}{\unvbox\csuse{##1footins}}%
6949     \fi}%
6950   \dolistloop{\@series}
6951   \fi%
6952 }
6953
6954 %
```

`\print@notesX` We have to add all the new kinds of familiar footnotes to the output routine. A macro  
`\append@notesX` which appends the familiar footnotes of one series onto the output routine, also adding  
`\do@feetX` vertical skip before notes.

```
\@doreinfeetX
6955 \newcommand{\append@notesX}[1]{%
6956   \ifvoid\csuse{footins#1}\else%
6957     \global\skip\csuse{footins#1}=\csuse{beforenotesX@#1}%
6958     \global\advance\skip\csuse{footins#1} by\csuse{afterruleX@#1}%
6959     \print@notesX{#1}%
6960   \fi%
6961 }%
6962 %
```

The normal way to print one series of notes. `\print@Xnotes` is replaced by `reledpar` when using `\Pages`.

```
6963 \newcommand\print@notesX[1]{%
6964   \xdef\@currentseries{#1}%
6965   \csuse{footstart#1}{#1}%
6966   \csuse{footgroup#1}{#1}%
6967 }%
6968 %
```

We print all the series of notes by looping over them. We check before printing them that they are not voided.

```

6969 \newcommand*{\do@feetX}{%
6970   \unless\ifnofamiliar@%
6971   \gdef\firstseriesX@{}%
6972   \setbox\@outputbox \vbox{%
6973     \unvbox\@outputbox%
6974     \def\do##1{%
6975       \append@notesX{##1}%
6976     }%
6977     \dolistloop{\@series}}%
6978   \fi%
6979 }%
6980
6981 \newcommand{\@doreinfeetX}{%
6982   \unless\ifnofamiliar@%
6983   \def\do##1{%
6984     \ifvoid\csuse{footins##1}\else
6985       \insert%
6986         \csuse{footins##1}
6987         {\unvbox\csuse{footins##1}}%
6988     \fi%
6989   }%
6990   \dolistloop{\@series}%
6991   \fi%
6992 }%
6993 %
6994 %

```

## 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.)

```

6995 \@ifclassloaded{memoir}{%
6996 %

```

memoir is loaded so we use memoir's built in hooks.

```

6997   \g@addto@macro{\m@mdoextrafeet}{\l@ddoextrafeet}%
6998   \g@addto@macro{\m@mdodoreinextrafeet}{\l@ddodoreinextrafeet}%
6999   }{%
7000 %

```

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`.

```

7001   \ifboolexpr{%

```

```

7002     test{\@ifpackageloaded{fancyhdr}}%
7003     and test {\ifdef{\latex@makecol}}%
7004   }{%
7005     \patchcmd%
7006       {\latex@makecol}%
7007       {\xdef\@freelist{\@freelist\@midlist}}%
7008       {\xdef\@freelist{\@freelist\@midlist}\l@ddoxtrafeet}%
7009       {}%
7010       {\led@error@fail@patch@makecol}%
7011   }{%
7012     \patchcmd%
7013       {\@makecol}%
7014       {\xdef\@freelist{\@freelist\@midlist}}%
7015       {\xdef\@freelist{\@freelist\@midlist}\l@ddoxtrafeet}%
7016       {}%
7017       {\led@error@fail@patch@makecol}%
7018   }%
7019
7020   \patchcmd%
7021     {\@reinserts}%
7022     {\ifvbox}%
7023     {\l@ddodoreinextrafeet\ifvbox}%
7024     {}%
7025     {\led@error@fail@patch@reinserts}%
7026 }
7027
7028 %

```

It turns out that `\@doclearpage` also needs modifying.

`\if@led@nofoot` We have to check if there are any leftover feet.

```

7029 \newif\if@led@nofoot
7030
7031 %

```

```

7032 \@ifclassloaded{memoir}{%
7033 %

```

If the memoir class is loaded, we hook into its modified `\@doclearpage`.

```

\@mem@extranofeet34 \g@addto@macro{\@mem@extranofeet}{%%
7035   \def\do#1{%
7036     \unless\ifnocritical@%
7037       \ifvoid\csuse{#1footins}\else\@mem@nofootfalse\fi%
7038     \fi%
7039     \unless\ifnofamiliar@%
7040       \ifvoid\csuse{footins#1}\else\@mem@nofootfalse\fi%
7041     \fi%
7042   }
7043   \dolistloop{\@series}%
7044 }%
7045 }{%
7046 %

```

As memoir is not loaded we have patch `\@docclearpage`.

```

\@led@testifnofoot 7047 \newcommand*{\@led@testifnofoot}{%
\@docclearpage 7048 \led@nofoottrue%
7049 \ifvoid\footins\else%
7050 \led@nofootfalse%
7051 \fi%
7052 \def\do##1{%
7053 \unless\ifnocritical@%
7054 \ifvoid\csuse{##1footins}\else%
7055 \led@nofootfalse%
7056 \fi%
7057 \fi%
7058 \unless\ifnofamiliar@%
7059 \ifvoid\csuse{footins##1}\else%
7060 \led@nofootfalse%
7061 \fi%
7062 \fi%
7063 }%
7064 \dolistloop{\@series}%
7065 }%
7066
7067 \pretocmd%
7068 {\@docclearpage}%
7069 {\@led@testifnofoot}%
7070 {}%
7071 {\led@error@fail@patch@@docclearpage}%
7072
7073 \patchcmd%
7074 {\@docclearpage}%
7075 {\ifvoid\footins}%
7076 {\if@led@nofoot}%
7077 {}%
7078 {\led@error@fail@patch@@docclearpage}%
7079
7080 }
7081
7082 %

```

## 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 `sameparallelpagnumber` option allows the same page number on both left and right side. The `prevpgnotnumbered` option allows an empty (not numbered) right-side page before `\Pages`.

We cannot implement these two options by changing the value of the page counter, since its value is used by many  $\text{\LaTeX}$  features to determine whether a page is left (even numbered) or right (odd numbered). Consequently, we have to do it by patching `\thepage`, in order to use the value of the `par@page` counter instead of the value of the page counter.



This counter will be increased in a patched version of L<sup>A</sup>T<sub>E</sub>X's `\@outputpage` macro, as is the page counter in this macro. However, this increase will take account of the options.

`\par@patch@thepage` and `\par@patch@pagenumbering` patches `\thepage` in order to use the value of `\par@page` counter and not the value of `\page`. It must be called after any redefinition of `\thepage`. That is why we insert it at the end of the L<sup>A</sup>T<sub>E</sub>X macro `\pagenumbering`, which is called by some `\xxxmatter` commands. In cases when we are using the memoir class, we insert it at the end of `\@mempnum`. When using `\pagenumbering`, we also need to reset the `\par@page` counter. Consequently, we put `\par@patch@thepage` and `\counter reset` in `\par@patch@pagenumbering`. We also call `\par@patch@thepage` at the beginning of the document.

```

7083
7084 \newcommand{\par@patch@thepage}{%
7085   \ifboolexpr{%
7086     bool{sameparallelpagenumber}%
7087     or bool{prevpnotnumbered}%
7088   }%
7089   {%
7090     \patchcmd{\thepage}%
7091       {page}{\par@page}%
7092       {}%
7093       {\led@error@fail@patch@thepage}%
7094   }{}%
7095 }%
7096
7097 \newcommand{\par@patch@pagenumbering}{%
7098   \setcounter{\par@page}{1}%
7099   \par@patch@thepage%
7100 }%
7101
7102 \ifboolexpr{%
7103   bool {l@dmemoir}%
7104   and test {ifdef{\@mempnum}}%Only with memoir < 3.8
7105 }{%
7106   \apptocmd{\@mempnum}%
7107     {\par@patch@pagenumbering}%
7108     {}%
7109     {\led@error@fail@patch@mempnum}%
7110   }%
7111   {%
7112     \pretocmd{\pagenumbering}%
7113       {\par@patch@pagenumbering}%
7114       {}%
7115       {\led@error@fail@patch@pagenumbering}%
7116     }%
7117
7118 \AtBeginDocument{\par@patch@thepage}%
7119 %

```

`\@outputpage` As its name says, `\@outputpage` is a L<sup>A</sup>T<sub>E</sub>X 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.).

```

7120 \AtBeginDocument{%
7121   \set@this@c@page%
7122   \apptocmd{\@outputpage}{%
7123     \ifsameparallelpagelength%
7124     \ifl@dprintingpages%
7125     \ifodd\c@page\else%
7126       \stepcounter{par@page}%
7127     \fi%
7128   \else%
7129     \stepcounter{par@page}%
7130   \fi%
7131 \else%
7132   \stepcounter{par@page}%
7133 \fi%
7134 }%
7135 {}%
7136 {\led@error@fail@patch@outputpage}%
7137 }%
7138 \catcode\#=#12%
7139 \AtBeginDocument{%
7140   \apptocmd{\setcounter}{%
7141     \IfStrEq{#1}{page}{\set@this@c@page}{}%
7142   }%
7143   {}%
7144   {\led@error@fail@patch@setcounter}%
7145   \apptocmd{\addtocounter}{%
7146     \IfStrEq{#1}{page}{\set@this@c@page}{}%
7147   }%
7148   {}%
7149   {\led@error@fail@patch@addtocounter}%
7150   \@ifpackageloaded{calc}{%
7151     \apptocmd{\stepcounter}{%Because calc.sty does not use addtocounter in
stepcounter
7152       \IfStrEq{#1}{page}{\set@this@c@page}{}%
7153     }%
7154     {}%
7155     {\led@error@fail@patch@steptocounter}%
7156   }{}%
7157 }%
7158 \catcode\#=#6%
7159 %

```

`\thepar@page` The `par@page` counter.

```

7160 \newcounter{par@page}%
7161 \setcounter{par@page}{1}%
7162 %

```

## 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.

```
7163 \list@create{\labelref@list}
7164 %
```

`\zz@@@` Two convenience macros to zero three / four labeling counters in one go.

```
\zz@@@
\zz@@@
7165 \newcommand*\zz@@@{000|000|000}% Set three counters to zero in one go
7166 \newcommand*\zz@@@{000|000|000|000}% Set four counters to zero in one go
7167
7168 %
```

`\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 L<sup>A</sup>T<sub>E</sub>X 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.

```
7169 \newcommand*\edlabel}[1]{%
7170   \leavevmode%
7171   \@bsphack%
7172   \ifboolexpr{bool{ledRcol} or bool{ledRcol@}}{%
7173     \ifXnote@%
7174     \protected@write\@auxout{%
7175       {\string\l@dmake@labelsR\space\thepage|\l@dparsedstartline|\
l@dparsedstartsub||\annot@start|\the\c@pstartR|{#1}}%
7176     \ifdef\hypertarget{%
7177       {\Hy@raisedlink{\hypertarget{#1}{}}}%
7178     }%
7179     \else%
7180     \write\linenum@outR{\string\@lab}%
7181     \ifx\labelref@listR\empty%
7182     \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.

```

7183 \else%
7184 \glp\labelref@listR\to\label@refs%
7185 \fi%
7186 \ifvmode%
7187 \advancelabel@refs%
7188 \fi%
7189 %

```

Use code from the kernel `\label` command to write the correct page number. Also define an `hypertarget` if `hyperref` package is loaded.

```

7190 \protected@write\@auxout{%
7191 {\string\l@dmake@labelsR\space\thepage|\label@refs|\the\c@pstartR
|{#1}}%
7192 \ifdef{\hypertarget}%
7193 {\Hy@raisedlink{\hypertarget{#1}{}}}%
7194 {}%
7195 \fi%
7196 }{%
7197 \ifXnote@%
7198 \ifl@dpairing%pstart or pstartL?
7199 \protected@write\@auxout{%
7200 {\string\l@dmake@labels\space\thepage|\l@dparsedstartline|\
l@dparsedstartsub||\annot@start|\the\c@pstartL|{#1}}%
7201 \ifdef{\hypertarget}%
7202 {\Hy@raisedlink{\hypertarget{#1}{}}}%
7203 {}%
7204 \else%
7205 \protected@write\@auxout{%
7206 {\string\l@dmake@labels\space\thepage|\l@dparsedstartline|\
l@dparsedstartsub||\annot@start|\the\c@pstart|{#1}}%
7207 \ifdef{\hypertarget}%
7208 {\Hy@raisedlink{\hypertarget{#1}{}}}%
7209 {}%
7210 \fi%
7211 \else%
7212 \write\linenum@out{\string\@lab}%
7213 \ifx\labelref@list\empty%
7214 \xdef\label@refs{\zz@@@}%
7215 \else%
7216 \glp\labelref@list\to\label@refs%
7217 \fi%
7218 \ifvmode%
7219 \advancelabel@refs%
7220 \fi%
7221 \ifl@dpairing%Pstart or PstartL?
7222 \protected@write\@auxout{%
7223 {\string\l@dmake@labels\space\thepage|\label@refs|\the\c@pstartL
|{#1}}%
7224 \ifdef{\hypertarget}%
7225 {\Hy@raisedlink{\hypertarget{#1}{}}}%
7226 {}%
7227 \else%
7228 \protected@write\@auxout{%
7229 {\string\l@dmake@labels\space\thepage|\label@refs|\the\c@pstart

```

```

|{#1}}%
7230     \ifdef{\hypertarget}%
7231         {\Hy@raisedlink{\hypertarget{#1}{}}}%
7232         {}%
7233     \fi%
7234 \fi%
7235 }%
7236 \@esphack}%
7237
7238 %

```

`\advance@label@refs` `\label@refs@parseline` `\label@refs@parsesubline` 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 `\advance@label@refs` command.

```

7239 \newcounter{line}%
7240 \newcounter{subline}%
7241 \newcounter{absline}%
7242 \newcommand{\advance@label@refs}{%
7243     \setcounter{line}{\expandafter\label@refs@parseline\label@refs}%
7244     \stepcounter{line}%
7245     \setcounter{absline}{\expandafter\label@refs@parseabsline\label@refs}%
7246     \stepcounter{absline}%
7247     \ifsublines%
7248         \setcounter{subline}{\expandafter\label@refs@parsesubline\label@refs}
7249         \stepcounter{subline}{1}%
7250         \def\label@refs{\theline|\thesubline|\theabsline}%
7251     \else%
7252         \def\label@refs{\theline|0|\theabsline}%
7253     \fi%
7254 }
7255 \def\label@refs@parseline#1|#2|#3{#1}%
7256 \def\label@refs@parsesubline#1|#2|#3{#2}%
7257 \def\label@refs@parseabsline#1|#2|#3{#3}%
7258 %

```

`\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.

```

7259 \newcommand*{\l@dmake@labels}{%
7260 \def\l@dmake@labels#1|#2|#3|#4|#5|#6|#7{%
7261     \expandafter\ifx\csname the@label\csuse{XR@prefix}#7\endcsname%
7262     \relax%
7263     \else%

```

```

7264 \led@warn@DuplicateLabel{\csuse{XR@prefix}#7}%
7265 \fi
7266 \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
7267 \csgdef{the@label\csuse{XR@prefix}#7}{#1|#2|#3|#4|#5|#6|\relax}%
7268 \ignorespaces}
7269
7270 %

```

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.

```

7271 \AtBeginDocument{%
7272 \def\l@dmake@labels#1|#2|#3|#4|#5|#6|#7{%
7273 }
7274
7275 %

```

**\@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.

```

7276
7277 \newcommand*{\@lab}{%
7278 \ifledRcol
7279 \xright@appenditem{\linenumr@p{\line@numR}}|%
7280 \ifsublines@ \sublinenumr@p{\subline@numR}\else 0\fi|%
7281 \the\absline@numR|%
7282 \current@annot%
7283 }%
7284 \to\labelref@listR
7285 \else
7286 \xright@appenditem{%
7287 \linenumr@p{\line@num}|%
7288 \ifsublines@ \sublinenumr@p{\subline@num}\else 0\fi|%
7289 \the\absline@num|%
7290 \current@annot%
7291 }%
7292 \to\labelref@list
7293 \fi}
7294 %

```

**\applabel** \applabel, if called in \edtext will insert automatically both a starting and an ending label for the current edtext lines.

```

7295 \newcommand*{\applabel}[1]{%
7296 \if@edtext@secondarg@%
7297 %

```

Label should not be already defined.

```

7298 \ifcsundef{the@label#1}{%
7299   \csdef{the@label#1}{applabel}%
7300 }%
7301 {%
7302   \led@warn@DuplicateLabel{#1 (applabel)}%
7303 }%
7304 %

```

Parse the `\edtext` line numbers and annotations.

```

7305 \expandafter\l@dp@rsefootspec\l@d@nums|%
7306 \expandafter\parse@annot\l@current@annot|%
7307 %

```

Use the  $\LaTeX$  standard hack for label.

```

7308 \@bsp@hack%
7309 %

```

And now, write the data in the auxiliary file.

```

7310 \ifledRcol%
7311   \protected@write\@auxout{%
7312     {\string\l@dmake@labelsR\space\l@d@p@r@s@d@s@t@r@t@p@a@g@e|\
l@d@p@r@s@d@s@t@l@i@n@e|\l@d@p@r@s@d@s@t@s@u@b||\a@n@n@o@t@s@t@r@t|\the\c@p@s@t@r@t@R|{#1:
start}}}%
7313   \ifdef{\hypertarget}%
7314     {\Hy@raisedlink{\hypertarget{#1:start}}}%
7315     {}%
7316   \protected@write\@auxout{%
7317     {\string\l@dmake@labelsR\space\l@d@p@r@s@d@s@e@n@d@p@a@g@e|\l@d@p@r@s@d@s@e@n@d@l@i@n@e
|\l@d@p@r@s@d@s@e@n@d@s@u@b||\a@n@n@o@t@e@n@d|\the\c@p@s@t@r@t@R|{#1:end}}}%
7318   \else%
7319     \ifl@dp@air@i@n@g%pstart or pstartL?
7320     \protected@write\@auxout{%
7321       {\string\l@dmake@labels\space\l@d@p@r@s@d@s@t@r@t@p@a@g@e|\
l@d@p@r@s@d@s@t@l@i@n@e|\l@d@p@r@s@d@s@t@s@u@b||\a@n@n@o@t@s@t@r@t|\the\c@p@s@t@r@t@L|{#1:
start}}}%
7322     \ifdef{\hypertarget}%
7323       {\Hy@raisedlink{\hypertarget{#1:start}}}%
7324       {}%
7325     \protected@write\@auxout{%
7326       {\string\l@dmake@labels\space\l@d@p@r@s@d@s@e@n@d@p@a@g@e|\
l@d@p@r@s@d@s@e@n@d@l@i@n@e|\l@d@p@r@s@d@s@e@n@d@s@u@b||\a@n@n@o@t@e@n@d|\the\c@p@s@t@r@t@L|{#1:end}}}%
7327     \else%
7328     \protected@write\@auxout{%
7329       {\string\l@dmake@labels\space\l@d@p@r@s@d@s@t@r@t@p@a@g@e|\
l@d@p@r@s@d@s@t@l@i@n@e|\l@d@p@r@s@d@s@t@s@u@b||\a@n@n@o@t@s@t@r@t|\the\c@p@s@t@r@t@L|{#1:
start}}}%
7330     \ifdef{\hypertarget}%
7331       {\Hy@raisedlink{\hypertarget{#1:start}}}%
7332       {}%
7333     \protected@write\@auxout{%
7334       {\string\l@dmake@labels\space\l@d@p@r@s@d@s@e@n@d@p@a@g@e|\
l@d@p@r@s@d@s@e@n@d@l@i@n@e|\l@d@p@r@s@d@s@e@n@d@s@u@b||\a@n@n@o@t@e@n@d|\the\c@p@s@t@r@t@L|{#1:end}}}%

```

```

7335     \fi%
7336     \fi%
7337 %

```

Use the L<sup>A</sup>T<sub>E</sub>X standard hack for label.

```

7338     \@esphack%
7339 %

```

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.

```

7340     \ifdef{\hypertarget}%
7341     {%
7342         \ifcsundef{applabel@thisedtext@the\@edtext@level}{\global\
cslet{applabel@thisedtext@the\@edtext@level}{\relax}}{}%
7343         \listcsgadd{applabel@thisedtext@the\@edtext@level}{#1}%
7344     }%
7345     {}%
7346 %

```

Warning if \applabel is called outside of \edtext.

```

7347     \else%
7348         \led@warn@AppLabelOutSecondArgEdtext{#1}%
7349     \fi%
7350 %

```

End of \applabel

```

7351 }%
7352 %

```

**\edlabelS** \edlabelS and \edlabelE are just used to mark the beginning and the end of a passage.

```

\edlabelE
\edlabelSE
7353 \newcommand{\edlabelS}[1]{%
7354     \edlabel{#1:start}%
7355 }
7356 \newcommand{\edlabelE}[1]{%
7357     \edlabel{#1:end}%
7358 }
7359 \newcommand{\edlabelSE}[1]{%
7360     \edlabelS{#1}%
7361     \edlabelE{#1}%
7362 }
7363 %

```

**\wrap@edcrossref** \wrap@edcrossref is called around all reledmac crossref commands, except those which start with x. It adds the hyperlink.

```

7364 \newrobustcmd{\wrap@edcrossref}[2]{%
7365     \ifdef{\hyperlink}%
7366         {\hyperlink{#1}{#2}}%
7367         {#2}%
7368 }
7369 %

```



`\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.

$\LaTeX$  already defines a `\pageref`, so changing the name to `\edpageref`.

```
7370 \newcommand*\edpageref}[1]{\l@dref@undefined{#1}\wrap@edcrossref{#1}{\l@dgetref@num{1}{#1}}
7371 \newcommand*\xpageref}[1]{\l@dgetref@num{1}{#1}}
7372
7373 %
```

`\edlineref` If the specified label exists, `\lineref` gives its line number.

`\xlineref`

```
7374 \newcommand*\edlineref}[1]{%
7375   \l@dref@undefined{#1}%
7376   \wrap@edcrossref{#1}{%
7377     \providetoggle{label@#1@ledRcol}%Required for the first run, when the
label has not yet been parsed on the .aux file
7378     \iftoggle{label@#1@ledRcol}%
7379       {\linenumrepR{\l@dgetref@num{2}{#1}}}%
7380       {\linenumrep{\l@dgetref@num{2}{#1}}}%
7381     \xflagref{#1}%
7382   }%
7383 }%
7384 \newcommand*\xlineref}[1]{\l@dgetref@num{2}{#1}}%
7385
7386 %
```

`\sublineref` If the specified label exists, `\sublineref` gives its sub-line number.

`\xsublineref`

```
7387 \newcommand*\sublineref}[1]{%
7388   \l@dref@undefined{#1}%
7389   \wrap@edcrossref{#1}{%
7390     \providetoggle{label@#1@ledRcol}%Required for the first run, when the
label has not yet been parsed on the .aux file
7391     \iftoggle{label@#1@ledRcol}%
7392       {\sublinenumrepR{\l@dgetref@num{3}{#1}}}%
7393       {\sublinenumrep{\l@dgetref@num{3}{#1}}}%
7394     }%
7395   }%
7396 \newcommand*\xsublineref}[1]{\l@dgetref@num{3}{#1}}
7397
7398 %
```

`\xabslineref` If the specified label exists, `\xabslineref` gives its absolute line number. This is generally used only by `reledmac` internal macros.

```
7399 \newcommand*\xabslineref}[1]{\l@dgetref@num{4}{#1}}%
7400 %
```

`\annotationref` If the specified label exists, `\edannotationref` gives the line number annotation with which it is associated.

`\xannotationref`

```

7401 \newcommand*\annotationref}[1]{%
7402   \l@dref@undefined{#1}%
7403   \wrap@edcrossref{#1}{%
7404     \l@wrap@ifnotemptybox{\Xwraplinenumannotation@ref}{%
7405       \l@dgetref@num{5}{#1}%
7406     }%
7407   }%
7408 }%
7409 \newcommand*\xannotationref}[1]{%
7410   \l@dgetref@num{5}{#1}%
7411 }%
7412 %

```

`\pstartref` If the specified label exists, `\pstartref` gives its pstart number.

`\xpstartref`

```

7413 \newcommand*\pstartref}[1]{\l@dref@undefined{#1}\wrap@edcrossref{#1}{\
7414   \l@dgetref@num{6}{#1}}}%
7415 \newcommand*\xpstartref}[1]{\l@dgetref@num{6}{#1}}}%
7416 %

```

`\xflagref` `\xflagref` finds the side flag of any ref defined with `\edlabel`.

```

7417 \newcommand*\xflagref}[1]{\l@dgetref@num{7}{#1}}%
7418 %

```

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.

```

7419 \newcommand*\l@dref@undefined}[1]{%
7420   \expandafter\ifx\csname the@label#1\endcsname\relax
7421     \led@warn@RefUndefined{#1}%
7422   \fi}
7423 %
7424 %

```

`\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.

```

7425 \newcommand*\l@dgetref@num}[2]{%
7426   \expandafter
7427   \ifx\csname the@label#2\endcsname \relax
7428     000%

```

```

7429 \else
7430 \expandafter\expandafter\expandafter
7431 \l@dlabel@parse\csname the@label#2\endcsname|#1%
7432 \fi}
7433
7434 %

```

`\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`.)

```

7435 \newcommand*\l@dlabel@parse{ }
7436 \def\l@dlabel@parse#1|#2|#3|#4|#5|#6|#7|#8{%
7437 \ifcase #8%
7438 \or #1%
7439 \or #2%
7440 \or #3%
7441 \or #4%
7442 \or #5%
7443 \or #6%
7444 \or #7%
7445 \fi}
7446 %

```

`\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.

```

7447 \newcommand*\xxref}[2]{%
7448 {%
7449 \expandafter\ifx\csname the@label#1\endcsname\relax%
7450 \let\@tempa\zz@@@%
7451 \def\@tempc{ }%
7452 \else%
7453 \def\@tempa{%
7454 \l@dgetref@num{1}{#1}|%
7455 \l@dgetref@num{2}{#1}|%
7456 \l@dgetref@num{3}{#1}%
7457 }%
7458 \edef\@tempc{%
7459 \l@dgetref@num{5}{#1}%
7460 }%
7461 \fi%
7462 \expandafter\ifx\csname the@label#2\endcsname\relax%

```

```

7463 \let\@tempb\zz@@@%
7464 \def\@tempd{}%
7465 \else%
7466 \def\@tempb{%
7467 \l@getref@num{1}{#2}|%
7468 \l@getref@num{2}{#2}|%
7469 \l@getref@num{3}{#2}%
7470 }%
7471 \edef\@tempd{%
7472 \l@getref@num{5}{#2}%
7473 }%
7474 \fi%
7475 \global\appto\@beforeinsertofthisedtext{%
7476 \def\@this@crossref@start{#1}%
7477 }%
7478 \global\appto\@beforeinsertofthisedtext{%
7479 \def\@this@crossref@end{#2}%
7480 }%
7481 \linenum{\@tempa|\@tempb}%
7482 \edef\@tempe{\@tempc|\@tempd}%
7483 \expandafter\lineannot\expandafter{\@tempe}%
7484 }%
7485 }%
7486
7487
7488 %

```

`\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`.

```

7489 \def\Xtwolines@appref{}%
7490 \def\Xtwolines@SEref{}%
7491
7492 \def\Xmorethantwolines@appref{}%
7493 \def\Xmorethantwolines@SEref{}%
7494
7495 \def\Xlinerangeseparator@appref{\endashchar}%
7496 \def\Xlinerangeseparator@SEref{\endashchar}%
7497
7498 \def\Xsublinesep@appref{\fullstop}%
7499 \def\Xsublinesep@SEref{\fullstop}%
7500
7501 \def\Xpagelinesep@appref{\fullstop}%
7502 \def\Xpagelinesep@SEref{\fullstop}%
7503
7504
7505 \newtoggle{Xtwolinesbutnotmore@appref}%
7506 \newtoggle{Xtwolinesbutnotmore@SEref}%
7507

```

```

7508 \newtoggle{Xtwolinesonlyinsamepage@appref}%
7509
7510 \newtoggle{Xtwolinesonlyinsamepage@SEref}%
7511
7512 \newtoggle{Xlineflag@appref}%
7513 \toggletrue{Xlineflag@appref}%Here exception
7514 \newtoggle{Xlineflag@SEref}%
7515 \toggletrue{Xlineflag@SEref}%%Here exception
7516
7517 \newtoggle{Xlinenumannotationonlyfirst@SEref}%Never changed, just for the
    test in \setprintlines
7518 \newtoggle{Xlinenumannotationonlyfirst@appref}%Idem
7519 \newtoggle{Xnolinenumber@SEref}%Idem
7520 \newtoggle{Xnolinenumber@appref}%Idem
7521 \newtoggle{Xnolinenumberifannotation@SEref}%Idem
7522 \newtoggle{Xnolinenumberifannotation@appref}%Idem
7523
7524
7525 \def\Xlinenumannotationposition@appref{after}%
7526 \def\Xlinenumannotationposition@SEref{after}%
7527
7528 \def\Xwraplinenumannotation@appref{\textsuperscript}%
7529 \def\Xwraplinenumannotation@SEref{\textsuperscript}%
7530 \newtoggle{Xnoidenticallinenumannotation@appref}%
7531 \newtoggle{Xnoidenticallinenumannotation@SEref}%
7532
7533 \def\Xendtwolines@apprefwithpage{}%
7534 \def\Xendtwolines@SErefwithpage{}%
7535
7536 \def\Xendmorethantwolines@apprefwithpage{}%
7537 \def\Xendmorethantwolines@SErefwithpage{}%
7538
7539 \def\Xendlinerangeseparator@apprefwithpage{\endashchar}
7540 \def\Xendlinerangeseparator@SErefwithpage{\endashchar}
7541 \def\Xendlinerangeseparator@SErefonlypage{\endashchar}
7542
7543 \def\Xendbeforepagenumber@apprefwithpage{p.}%
7544 \def\Xendbeforepagenumber@SErefwithpage{p.}%
7545 \def\Xendbeforepagenumber@SEonlypage{p.}%
7546
7547 \def\Xendafterpagenumber@apprefwithpage{} }%
7548 \def\Xendafterpagenumber@SErefwithpage{} }%
7549
7550
7551 \def\Xendlineprefixsingle@apprefwithpage{}%
7552 \def\Xendlineprefixsingle@SErefwithpage{}%
7553
7554 \def\Xendlineprefixmore@apprefwithpage{}%
7555 \def\Xendlineprefixmore@SErefwithpage{}%
7556
7557 \newtoggle{Xendtwolinesbutnotmore@apprefwithpage}%
7558 \newtoggle{Xendtwolinesbutnotmore@SErefwithpage}%
7559
7560 \def\Xendsublinesep@apprefwithpage{\fullstop}%

```

```

7561 \def\Xendsublinesep@SErefwithpage{\fullstop}%
7562
7563 \newtoggle{Xendtwolinesonlyinsamepage@apprefwithpage}%
7564 \newtoggle{Xendtwolinesonlyinsamepage@SErefwithpage}%
7565
7566 \newtoggle{Xendlineflag@apprefwithpage}
7567 \toggletrue{Xendlineflag@apprefwithpage}%Here, exception
7568 \newtoggle{Xendlineflag@SErefwithpage}
7569 \toggletrue{Xendlineflag@SErefwithpage}%Here, exception
7570
7571
7572 \def\Xendlinenumannotationposition@apprefwithpage{after}%
7573 \def\Xendlinenumannotationposition@SErefwithpage{after}%
7574
7575 \def\Xendlinenumannotationposition@apprefwithpage{after}%
7576 \def\Xendlinenumannotationposition@apprefwithpage@SErefwithpage{after}%
7577
7578 \def\Xendwraplinenumannotation@apprefwithpage{\textsuperscript}%
7579 \def\Xendwraplinenumannotation@SErefwithpage{\textsuperscript}%
7580
7581 \newtoggle{Xendnoidenticallinenumannotation@apprefwithpage}%
7582 \newtoggle{Xendnoidenticallinenumannotation@SErefwithpage}%
7583
7584
7585 \newtoggle{Xendnolinenum@SErefwithpage}%Never changed, only for test in
  \setprintendlines
7586 \newtoggle{Xendnolinenum@apprefwithpage}%Idem
7587 \newtoggle{Xendnolinenum@SErefwithpage}%Idem
7588 \newtoggle{Xendnolinenum@SErefwithpage}%Idem
7589
7590 %

```

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.

```

7591
7592 \newtoggle{Xendlinenumannotationonlyfirst@SEref}%Never changed, just for
  the test in \setprintendlines
7593 \newtoggle{Xendlinenumannotationonlyfirst@SErefwithpage}%Idem
7594 \newtoggle{Xendlinenumannotationonlyfirst@appref}%Idem
7595 \newtoggle{Xendlinenumannotationonlyfirst@apprefwithpage}%Idem
7596
7597 \gdef\Xboxstartlinenum@appref{0pt}
7598 \gdef\Xboxstartlinenum@SEref{0pt}
7599 \newtoggle{Xnotboxingsubline@appref}%
7600 \newtoggle{Xnotboxingsubline@SEref}%
7601
7602 \gdef\Xboxendlinenum@appref{0pt}
7603 \gdef\Xboxendlinenum@SEref{0pt}
7604
7605 \gdef\Xendboxstartlinenum@apprefwithpage{0pt}
7606 \gdef\Xendboxstartlinenum@SErefwithpage{0pt}
7607 \newtoggle{Xendnotboxingsubline@apprefwithpage}%
7608 \newtoggle{Xendnotboxingsubline@SErefwithpage}%

```

```

7609
7610 \gdef\Xendboxendlinenum@apprefwithpage{0pt}
7611 \gdef\Xendboxendlinenum@SErefwithpage{0pt}
7612
7613 \newtoggle{Xendpagenumberonlyfirst@apprefwithpage}
7614 \newtoggle{Xendpagenumberonlyfirst@SErefwithpage}
7615
7616 \newtoggle{Xendpagenumberonlyfirstifsingle@apprefwithpage}
7617 \newtoggle{Xendpagenumberonlyfirstifsingle@SErefwithpage}
7618
7619 \newtoggle{Xendpagenumberonlyfirstintwo@apprefwithpage}
7620 \newtoggle{Xendpagenumberonlyfirstintwo@SErefwithpage}
7621
7622 \gdef\Xendsympagenum@apprefwithpage{}
7623 \gdef\Xendsympagenum@SErefwithpage{}
7624
7625 \gdef\Xendinplaceofpagenumber@apprefwithpage{}
7626 \gdef\Xendinplaceofpagenumber@SErefwithpage{}
7627
7628 %

```

Now, declare the default values of \@apprefprefixsingle and \@apprefprefixmore, \@SErefprefix, \@SErefprefixmore and the commands which defines them.

```

7629 \newcommand\@apprefprefixsingle{}%
7630 \newcommand\@SErefprefixsingle{}%
7631
7632 \newcommand\@apprefprefixmore{}%
7633 \newcommand\@SErefprefixmore{}%
7634
7635 \newcommand{\setapprefprefixsingle}[1]{%
7636   \gdef\@apprefprefixsingle{#1}%
7637 }
7638 \newcommand{\setSErefprefixsingle}[1]{%
7639   \gdef\@SErefprefixsingle{#1}%
7640 }
7641
7642 \newcommand{\setapprefprefixmore}[1]{%
7643   \gdef\@apprefprefixmore{#1}%
7644 }
7645 \newcommand{\setSErefprefixmore}[1]{%
7646   \gdef\@SErefprefixmore{#1}%
7647 }
7648
7649 %

```

And not \setSErefonlypageprefixsingle and \setSErefonlypageprefixmore.

```

7650 \newcommand{\setSErefonlypageprefixsingle}[1]{%
7651   \gdef\SErefonlypage@prefixsingle{#1}%
7652 }%
7653 \newcommand{\setSErefonlypageprefixmore}[1]{%
7654   \gdef\SErefonlypage@prefixmore{#1}%
7655 }%
7656 %

```

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.

```

7657
7658 \newcommandx{\appref}[2][1,usedefault]{\reformatted@{#1}{#2}{appref}}
7659 \newcommandx{\SEref}[2][1,usedefault]{\reformatted@{#1}{#2}{SEref}}
7660
7661 \newcommandx{\apprefwithpage}[2][1,usedefault]{\reformattedwithpage@
7662 {#1}{#2}{appref}}
7663 \newcommandx{\SErefwithpage}[2][1,usedefault]{\reformattedwithpage@
7664 {#1}{#2}{SEref}}
7665
7666 \newcommand{\reformatted@}[3]{%
7667   \def\do##1{%
7668     \setkeys[mac]{truefootnoteoption}{##1}%
7669   }%
7670   \notblank{#1}{\docsvlist{#1}}{ }%
7671   \xdef\@currentseries{#3}%
7672   \iftoggle{noprefix@}{ }%
7673   {%
7674     \ifcempty{@#3prefixmore}%
7675     {\@apprefprefixsingle}%
7676     {%
7677       \IfEq{\xlineref{#2:start}}{\xlineref{#2:end}}%
7678       {\iftoggle{prefixmore@}%
7679         {\csuse{@#3prefixmore}}%
7680         {\csuse{@#3prefixsingle}}}%
7681       }%
7682       {\csuse{@#3prefixmore}}%
7683     }%
7684   }%
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   {%
7691     \def\@this@crossref@start{#2:start}%
7692     \def\@this@crossref@end{#2:end}%
7693     \xdef\annot@start{\xannotationref{#2:start}}%
7694     \xdef\annot@end{\xannotationref{#2:end}}%
7695     \printlines\xpageref{#2:start}|\xlineref{#2:start}|\xsublineref{#2:
7696 start}|\xpageref{#2:end}|\xlineref{#2:end}|\xsublineref{#2:end}|\relax|\
7697 xflagref{#2:start}|}%
7698   }%
7699   \def\do##1{%
7700     \setkeys[mac]{falsefootnoteoption}{##1}%
7701   }%
7702   \notblank{#1}{\docsvlist{#1}}{ }%

```



```

7701 }%
7702
7703 \newcommand{\reformattedwithpage@}[3]{%
7704   \def\do##1{%
7705     \setkeys[mac]{truefootnoteoption}{##1}%
7706   }%
7707   \notblank{#1}{\docsvlist{#1}}{%
7708     \xdef\@currentseries{#3withpage}%
7709     \ifboolexpr{%
7710       test{\ifcsundef{the@label#2:start}}%
7711       or test{\ifcsundef{the@label#2:end}}%
7712     }%
7713     {\led@warn@pairRefUndefined{#2}\nfss@text{\reset@font\bfseries ??}}%
7714     {%
7715       \def\@this@crossref@start{#2:start}%
7716       \def\@this@crossref@end{#2:end}%
7717       \xdef\annot@start{\xannotationref{#2:start}}%
7718       \xdef\annot@end{\xannotationref{#2:end}}%
7719       \printendlines\xpageref{#2:start}|\xlineref{#2:start}|\xsublineref{#2:
start}|\xpageref{#2:end}|\xlineref{#2:end}|\xsublineref{#2:end}|\relax|\
xflagref{#2:start}}%
7720     }%
7721     \def\do##1{%
7722       \setkeys[mac]{falsefootnoteoption}{##1}%
7723     }%
7724     \notblank{#1}{\docsvlist{#1}}{%
7725     }%
7726
7727 \newcommand{\reformattedonlypage@}[3]{%
7728   \def\do##1{%
7729     \setkeys[mac]{truefootnoteoption}{##1}%
7730   }%
7731   \notblank{#1}{\docsvlist{#1}}{%
7732     \xdef\@currentseries{#3onlypage}%
7733     \ifboolexpr{%
7734       test{\ifcsundef{the@label#2:start}}%
7735       or test{\ifcsundef{the@label#2:end}}%
7736     }%
7737     {\led@warn@pairRefUndefined{#2}\nfss@text{\reset@font\bfseries ??}}%
7738     {\ifnumequal{\xpageref{#2:end}}{\xpageref{#2:start}}%
7739       {%
7740         \ifcsvoid{#3onlypage@prefixsingle}%
7741         {}%
7742         {\csletcs{Xendbeforepagenumber@#3onlypage}{#3onlypage@prefixsingle
}}%
7743         \printnpnum{%
7744           \wrap@edcrossref{#2:start}{\xpageref{#2:start}}%
7745         }%
7746       }%
7747     }%
7748     \ifcsvoid{#3onlypage@prefixmore}%
7749     {}%
7750     {\csletcs{Xendbeforepagenumber@#3onlypage}{#3onlypage@prefixmore}}%
7751     \ifdefined\linerangesep%

```

```

7752     \printnpnum{%
7753         \wrap@edcrossref{#2:start}{\xpageref{#2:start}}}%
7754     \linangesep@%
7755     \wrap@edcrossref{#2:end}{\xpageref{#2:end}}}%
7756     }%
7757 \else%
7758     \printnpnum{%
7759         \wrap@edcrossref{#2:start}{\xpageref{#2:start}}}%
7760         \csuse{Xendlinangeseparator@{\currentseries}}%
7761         \wrap@edcrossref{#2:end}{\xpageref{#2:end}}}%
7762     }%
7763     \fi%
7764 }%
7765 }%
7766 \def\do##1{%
7767     \setkeys[mac]{falsefootnoteoption}{##1}%
7768 }%
7769 \notblank{#1}{\docsvlist{#1}}{-}%
7770 }%
7771 %

```

`\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`.

```

7772 \newcommand*{\edmakelabel}[2]{\expandafter\xdef\csname the@label#1\
7773     endcsname{#2}}
7774 %

```

(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`.

```

7775 \pretocmd{\XR@test}%
7776     {\XR@test@mac+++#1#2#3#4+++}%
7777     {}%
7778     {}%
7779 %

```

`\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`.

```
7780 \long\def\xR@test@mac+++#1+++{\XR@test@mac@test#1}
7781 %
```

`\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`.

```
7782 \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
7783 \ifx#1\l@dmake@labels%
7784 \l@dmake@labels#2%
7785 \else
7786 \ifx#1\l@dmake@labelsR%
7787 \l@dmake@labelsR #2%
7788 \fi%
7789 \fi%
7790 }%
7791 %
```

## 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.

```
7792 \pretocmd{\@xympar}%
7793 {\ifnumberedpar@
7794 \led@warn@NoMarginpars
7795 \esphack
7796 \else}%
7797 {}%
7798 {}%
7799
7800 \apptocmd{\@xympar}%
7801 {\fi}%
7802 {}
7803 {}
7804
7805 %
```

We provide sidenotes as replacement for `\marginpar` in numbered text.

`\sidenote@margin` These are the sidenote equivalents to `\line@margin` and `\linenummargin` for specifying which margin. The default is the right margin (opposite to the default for line numbers). `\l@dgetsidenote@margin` returns the number associated to sidenote margin:

**left:** 0

**right:** 1

**outer:** 2

**inner:** 3

```

7806 \newcount\sidenote@margin
7807 \newcommand*{\sidenotemargin}[1]{%
7808   \l@getsidenote@margin{#1}%
7809   \ifnum\@l@dttempcntb>\m@ne
7810     \ifledRcol
7811       \global\sidenote@marginR=\@l@dttempcntb
7812     \else
7813       \global\sidenote@margin=\@l@dttempcntb
7814     \fi
7815   \fi}}
7816 \newcommand*{\l@getsidenote@margin}[1]{%
7817   \def\@tempa{#1}\def\@tempb{left}%
7818   \ifx\@tempa\@tempb
7819     \@l@dttempcntb \z@
7820   \else
7821     \def\@tempb{right}%
7822     \ifx\@tempa\@tempb
7823       \@l@dttempcntb \@ne
7824     \else
7825       \def\@tempb{outer}%
7826       \ifx\@tempa\@tempb
7827         \@l@dttempcntb \tw@
7828       \else
7829         \def\@tempb{inner}%
7830         \ifx\@tempa\@tempb
7831           \@l@dttempcntb \thr@@
7832         \else
7833           \led@warn@BadSidenotemargin
7834           \@l@dttempcntb \m@ne
7835         \fi
7836       \fi
7837     \fi
7838   \fi}
7839 \sidenotemargin{right}
7840
7841 %

```

`\l@dlp@rbox` We need two boxes to store sidenote texts.

```

\l@drp@rbox
7842 \newbox\l@dlp@rbox
7843 \newbox\l@drp@rbox
7844
7845 %

```

`\ledlsnotewidth` These specify the width of the left/right boxes (initialised to `\marginparwidth`), their distance from the text (initialised to `\linenumsep`), and the fonts used.

```

\ledrsnotewidth
\ledlsnotesep
7846 \newdimen\ledlsnotewidth \ledlsnotewidth=\marginparwidth
\ledrsnotesep
\ledlsnotefontsetup
\ledrsnotefontsetup

```

```

7847 \newdimen\ledrsnotewidth \ledrsnotewidth=\marginparwidth
7848 \newdimen\ledlsnotesep \ledlsnotesep=\linenumsep
7849 \newdimen\ledrsnotesep \ledrsnotesep=\linenumsep
7850 \newcommand*\ledlsnotefontsetup{\raggedleft\footnotesize}
7851 \newcommand*\ledrsnotefontsetup{\raggedright\footnotesize}
7852
7853 %

```

`\ledleftnote` `\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@
7854 \newcommand*\ledleftnote[1]{%
7855   \ledsidenote@{\l@dlsnote{#1}}%
7856 }%
7857 \newcommand*\ledrightnote[1]{%
7858   \ledsidenote@{\l@drsnote{#1}}%
7859 }%
7860 \newcommand*\ledsidenote[1]{%
7861   \ledsidenote@{\l@dcsnote{#1}}%
7862 }%
7863 \newcommand*\ledinnernote[1]{%
7864   \ledsidenote@{\l@disnote{#1}}%
7865 }%
7866 \newcommand*\ledouternote[1]{%
7867   \ledsidenote@{\l@dosnote{#1}}%
7868 }%
7869 \newcommand*\ledsidenote@[1]{%
7870   \edtext{}{%
7871     \advance\edtext@notes@start by 1%
7872     #1%
7873   }%+
7874 }%
7875 %

```

`\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
7876 \newif\ifrightnoteup
7877 \rightnoteuptrue
7878
7879 \newcommand*\l@dlsnote[1]{%
7880   \begingroup%
7881   \newcommand{\content}{#1}%
7882   \ifnumberedpar@
7883     \ifledRcol%
7884       \xright@appenditem{\noexpand\l@dlsnote{\expandonce\content}}%
7885         \to\inserts@listR
7886       \global\advance\insert@countR \@ne%
7887     \else%
7888       \xright@appenditem{\noexpand\l@dlsnote{\expandonce\content}}%
7889         \to\inserts@list

```

```

7890 \global\advance\insert@count \@ne%
7891 \fi
7892 \fi%
7893 \ignorespaces%
7894 \endgroup%
7895 }%
7896
7897 \newcommand*{\l@drsnote}[1]{%
7898 \begingroup%
7899 \newcommand{\content}{#1}%
7900 \ifnumberedpar@
7901 \ifledRcol%
7902 \xright@appenditem{\noexpand\l@drsnote{\expandonce\content}}%
7903 \to\inserts@listR
7904 \global\advance\insert@countR \@ne%
7905 \else%
7906 \xright@appenditem{\noexpand\l@drsnote{\expandonce\content}}%
7907 \to\inserts@list
7908 \global\advance\insert@count \@ne%
7909 \fi
7910 \fi\ignorespaces%
7911 \endgroup%
7912 }%
7913
7914 \newcommand*{\l@dcsnote}[1]{%
7915 \begingroup%
7916 \newcommand{\content}{#1}%
7917 \ifnumberedpar@
7918 \ifledRcol%
7919 \xright@appenditem{\noexpand\l@dcsnote{\expandonce\content}}%
7920 \to\inserts@listR
7921 \global\advance\insert@countR \@ne%
7922 \else%
7923 \xright@appenditem{\noexpand\l@dcsnote{\expandonce\content}}%
7924 \to\inserts@list
7925 \global\advance\insert@count \@ne%
7926 \fi
7927 \fi\ignorespaces%
7928 \endgroup%
7929 }%
7930
7931 \newcommand*{\l@disnote}[1]{%
7932 \begingroup%
7933 \newcommand{\content}{#1}%
7934 \ifnumberedpar@%
7935 \ifledRcol%
7936 \xright@appenditem{\noexpand\l@disnote{\expandonce\content}}%
7937 \to\inserts@listR%
7938 \global\advance\insert@countR \@ne%
7939 \else%
7940 \xright@appenditem{\noexpand\l@disnote{\expandonce\content}}%
7941 \to\inserts@list%
7942 \global\advance\insert@count \@ne%
7943 \fi%

```

```

7944 \fi\ignorespaces%
7945 \endgroup%
7946 }%
7947
7948 \newcommand*{\l@dosnote}[1]{%
7949 \begingroup%
7950 \newcommand{\content}{#1}%
7951 \ifnumberedpar%
7952 \ifledRcol%
7953 \xright@appenditem{\noexpand\l@dosnote{\expandonce\content}}%
7954 \to\inserts@listR%
7955 \global\advance\insert@countR \@ne%
7956 \else%
7957 \xright@appenditem{\noexpand\l@dosnote{\expandonce\content}}%
7958 \to\inserts@list%
7959 \global\advance\insert@count \@ne%
7960 \fi%
7961 \fi\ignorespaces%
7962 \endgroup%
7963 }%
7964
7965 %

```

`\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.

```

7966 \newcommand*{\vl@dlsnote}[1]{%
7967 \ifledRcol%
7968 \@l@dttempcntb=\sidenote@marginR%
7969 \ifnum\@l@dttempcntb>\@ne%
7970 \advance\@l@dttempcntb by\page@numR%
7971 \fi%
7972 \else%
7973 \@l@dttempcntb=\sidenote@margin%
7974 \ifnum\@l@dttempcntb>\@ne%
7975 \advance\@l@dttempcntb by\page@num%

```

```

7976   \fi%
7977   \fi%
7978   \ifodd\@l@dttempcntb%
7979     \listgadd{\l@dcsnotetext@l}{#1}%
7980   \else%
7981     \listgadd{\l@dcsnotetext}{#1}%
7982   \fi
7983 }
7984 \newcommand*\vl@drsnote}[1]{%
7985   \ifledRcol@%
7986     \@l@dttempcntb=\sidenote@marginR%
7987     \ifnum\@l@dttempcntb>\@ne%
7988       \advance\@l@dttempcntb by\page@numR%
7989     \fi%
7990   \else%
7991     \@l@dttempcntb=\sidenote@margin%
7992     \ifnum\@l@dttempcntb>\@ne%
7993       \advance\@l@dttempcntb by\page@num%
7994     \fi%
7995   \fi%
7996   \ifodd\@l@dttempcntb%
7997     \listgadd{\l@dcsnotetext}{#1}%
7998   \else%
7999     \listgadd{\l@dcsnotetext@r}{#1}%
8000   \fi%
8001 }
8002 \newcommand*\vl@dcsnote}[1]{\listgadd{\l@dcsnotetext}{#1}}
8003
8004 \newcommand{\vl@disnote}[1]{%
8005   \ifledRcol@%
8006     \@tempcnta=\page@numR%
8007   \else%
8008     \@tempcnta=\page@num%
8009   \fi%
8010   \ifodd\@tempcnta% ODD => right page => inner side = left side
8011     \vl@dlsnote{#1}%
8012   \else%
8013     \vl@drsnote{#1}%
8014   \fi%
8015 }%
8016
8017 \newcommand{\vl@dosnote}[1]{%
8018   \ifledRcol@%
8019     \@tempcnta=\page@numR%
8020   \else%
8021     \@tempcnta=\page@num%
8022   \fi%
8023   \ifodd\@tempcnta% ODD => right page => outer side = right side
8024     \vl@drsnote{#1}%
8025   \else%
8026     \vl@dlsnote{#1}%
8027   \fi%
8028 }%
8029

```



8030 %

`\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.

```

8031 \newcommand*{\setl@dlp@rbox}[1]{%
8032   \begingroup%
8033   \parindent\z@\hspace=\ledlspotewidth%
8034   \ledlspotefontsetup%We kept it outside of the vbox, because can affect
the ragging
8035   \global\setbox\l@dlp@rbox%
8036   \ifleftnoteup%
8037     =\vbox to\z@{\ledlspotefontsetup\vss #1}}%We put \
ledlspotefontsetup inside footnote because required for color command. Note
the {} to keep setting local.
8038   \else%
8039     =\vbox to 0.70\baselineskip{\ledlspotefontsetup\strut#1\vss}}%
8040   \fi%
8041   \endgroup%
8042 }
8043
8044 \newcommand*{\setl@drp@rbox}[1]{%
8045   \begingroup%
8046   \parindent\z@\hspace=\ledrspotewidth%
8047   \ledrspotefontsetup%We kept it outside of the vbox, because can affect
the ragging
8048   \global\setbox\l@drp@rbox%
8049   \ifrighnoteup%
8050     =\vbox to\z@{\ledrspotefontsetup\vss#1}}%We put \ledrspotefontsetup
inside footnote because required for color command. Note the {} to keep
setting local.
8051   \else%
8052     =\vbox to0.7\baselineskip{\ledrspotefontsetup\strut#1\vss}}%
8053   \fi%
8054   \endgroup%
8055 }%
8056 \newif\ifleftnoteup
8057 \leftnoteuptrue
8058 %

```

`\@sidenotesep` This macro is used to separate sidenotes of the same line.

```

8059 \newcommand{\setsidenotesep}[1]{\gdef\@sidenotesep{#1}}
8060 \newcommand{\@sidenotesep}{, }
8061 %

```

`\affixside@note` This macro puts any moveable sidenote text into the left or right sidenote box, depending on which margin it is meant to go in. It's a very much stripped down version of `\affixline@num`.

Before do it, we concatenate all moveable sidenotes of the line, using `\@sidenotesep` as separator. It is the result that we put on the sidenote.

```

8062 \newcommand*{\affixside@note}{%
8063   \prepare@edindex@fornote{\the\page@num|\the\line@num|\the\subline@num|\
the\page@num|\the\line@num|\the\subline@num|}%
8064   \def\sidenotecontent@{}%
8065   \numgdef{\itemcount@}{0}%
8066   \def\do##1{%
8067     \ifnumequal{\itemcount@}{0}%
8068     {%
8069     \appto\sidenotecontent@{##1}}% Not print not separator before
the 1st note
8070     {\appto\sidenotecontent@{\@sidenotesep ##1}}%
8071     }%
8072     \numgdef{\itemcount@}{\itemcount@+\@ne}%
8073   }%
8074   \dolistloop{\l@dcsnotetext}%
8075   \ifnumgreater{\itemcount@}{1}{\led@err@ManySidenotes}{}%
8076   %

```

And we do the same for left and right notes (not movable).

```

8077 \gdef\@templ@d{}%
8078 \gdef\@templ@n{\l@dcsnotetext\l@dcsnotetext@l\l@dcsnotetext@r}%
8079 \ifx\@templ@d\@templ@n \else%
8080   \if@twocolumn%
8081     \if@firstcolumn%
8082       \setl@dlp@rbox{##1}{\sidenotecontent@}%
8083     \else%
8084       \setl@drp@rbox{\sidenotecontent@}%
8085     \fi%
8086   \else%
8087     \@l@dttempcntb=\sidenote@margin%
8088     \ifnum\@l@dttempcntb>\@ne%
8089       \advance\@l@dttempcntb by\page@num%
8090     \fi%
8091     \ifodd\@l@dttempcntb%
8092       \setl@drp@rbox{\sidenotecontent@}%
8093       \gdef\sidenotecontent@{}%
8094       \numgdef{\itemcount@}{0}%
8095       \dolistloop{\l@dcsnotetext@l}%
8096       \ifnumgreater{\itemcount@}{1}{\led@err@ManyLeftnotes}{}%
8097       \setl@dlp@rbox{\sidenotecontent@}%
8098     \else%
8099       \setl@dlp@rbox{\sidenotecontent@}%
8100       \gdef\sidenotecontent@{}%
8101       \numgdef{\itemcount@}{0}%
8102       \dolistloop{\l@dcsnotetext@r}%
8103       \ifnumgreater{\itemcount@}{1}{\led@err@ManyRightnotes}{}%
8104       \setl@drp@rbox{\sidenotecontent@}%
8105     \fi%
8106   \fi%
8107 \fi%
8108 \advance\@edindex@fornote@m@ne%
8109 }
8110 %

```

## XXVII Minipages and such

We can put footnotes into minipages. The preparatory code has been set up earlier, all that remains is to ensure that it is available inside a minipage box. This requires some alteration to the kernel code, specifically the `\@iiminipage` and `\endminipage` macros. We will arrange this so that additional series can be easily added.

`\l@dfetbeginmini` These will be the hooks in `\@iiminipage` and `\endminipage`.  
`\l@dfetendmini` They can be extended to handle other things if necessary.

```

8111 \ifnoledgroup@else%
8112 \newcommand*{\l@dfetbeginmini}{\@ledgrouptrue\l@dedbeginmini\
l@dfambeginmini}
8113 \newcommand*{\l@dfetendmini}{%
8114   \IfStrEq{critical-familiar}{\@mpfnpos}%
8115     {\l@dedendmini\l@dfamendmini}%
8116     {%
8117       \IfStrEq{familiar-critical}{\@mpfnpos}%
8118         {\l@dfamendmini\l@dedendmini}%
8119         {\do@feet@custom@order{mp@}{\@mpfnpos}}}%
8120   }%
8121 }%
8122 %

```

`\l@dedbeginmini` These handle the initiation and closure of critical footnotes in a minipage environment.

```

\l@dedendmini
\mp@append@Xnotes
8123 \newcommand*{\l@dedbeginmini}{%
8124   \unless\ifnocritical@%
8125     \def\do##1{%
8126       \csletcs{v##1footnote}{mpv##1footnote}%
8127     }%
8128     \dolistloop{\@series}%
8129     \fi%
8130   }
8131 \newcommand*{\l@dedendmini}{%
8132   \unless\ifnocritical@%
8133     \ifl@dpairing%
8134       \ifledRcol%
8135         \flush@notesR%
8136       \else%
8137         \flush@notes%
8138       \fi%
8139     \fi
8140     \def\do##1{%
8141       \mp@append@Xnotes{##1}%
8142     }%
8143     \dolistloop{\@series}%
8144     \fi%
8145   }%
8146 \newcommand{\mp@append@Xnotes}[1]{%
8147 \ifvoid\csuse{mp#1footins}\else%
8148   \ifl@dpairing%
8149     \ifparledgroup%
8150     \ifledRcol%

```

```

8151 \dingdef{\parledgroup@beforenotesR}{\parledgroup@beforenotesR+\skip
\@nameuse{mp#1footins}}%
8152 \else%
8153 \dingdef{\parledgroup@beforenotesL}{\parledgroup@beforenotesL+\
skip\@nameuse{mp#1footins}}%
8154 \fi%
8155 \fi%
8156 \fi%
8157 \ifcsstring{series@display#1}{paragraph}{-}{%
8158 \setbox\@nameuse{mp#1footins}=\vbox{%
8159 \csuse{Xnotefontsize@#1}%
8160 \ifcsdef{Xhsize\csuse{series@display#1}@#1}{%
8161 \hsize\csuse{Xhsize\csuse{series@display#1}@#1}%
8162 }-%
8163 \noindent\csuse{Xtxtbeforenotes@#1}%
8164 \unvbox\@nameuse{mp#1footins}%
8165 \@parboxrestore%
8166 }%
8167 }%
8168 \csuse{mp#1footgroup}{#1}%
8169 \fi%
8170 }%
8171 %

```

`\l@dfambeginmini` These handle the initiation and closure of familiar footnotes in a minipage environment.

```

\l@dfamendmini
\mp@append@notesX
8172 \newcommand*{\l@dfambeginmini}{%
8173 \unless\ifnofamiliar@%
8174 \def\do##1{\csletcs{vfootnote##1}{mpvfootnote##1}}%
8175 \dolistloop{\@series}%
8176 \fi%
8177 }%
8178
8179 \newcommand*{\l@dfamendmini}{%
8180 \unless\ifnofamiliar@%
8181 \def\do##1{%
8182 \mp@append@notesX{##1}%
8183 }%
8184 \dolistloop{\@series}%
8185 \fi%
8186 }%
8187 \newcommand{\mp@append@notesX}[1]{%
8188 \ifvoid\csuse{mpfootins#1}\else%
8189 \ifcsstring{series@displayX#1}{paragraph}{-}{%
8190 \setbox\@nameuse{mpfootins#1}=\vbox{%
8191 \csuse{notefontsizeX@#1}%
8192 \ifcsdef{hsize\csuse{series@display#1}X@#1}{%
8193 \hsize\csuse{hsize\csuse{series@display#1}X@#1}%
8194 }-%
8195 \noindent\csuse{txtbeforenotesX@#1}%
8196 \unvbox\@nameuse{mpfootins#1}%
8197 \@parboxrestore%
8198 }%
8199 }%

```

```

8200 \csuse{mpfootgroup#1}{#1}%
8201 \fi%
8202 }%
8203 %

```

`\@iiiminipage` This is our extended form of the kernel `\@iiiminipage` defined in `ltboxes.dtx`.

```

8204 \patchcmd%
8205   {\@iiiminipage}%
8206   {\let\@footnotetext\@mpfootnotetext}%
8207   {\let\@footnotetext\@mpfootnotetext\l@dfteetbeginmini}%
8208   {}%
8209   {\led@error@fail@patch@\@iiiminipage}%
8210 %

```

`\endminipage` This is our extended form of the kernel `\endminipage` defined in `ltboxes.dtx`.

```

8211 \patchcmd%
8212   {\endminipage}%
8213   {\footnoterule}%
8214   {\footnoterule\l@advance@parledgroup@beforenormalnotes}%
8215   {}%
8216   {\led@error@fail@patch@endminipage}
8217
8218 \patchcmd%
8219   {\endminipage}%
8220   {\@minipagefalse}%
8221   {\l@dfteetendmini\@minipagefalse}%
8222   {}%
8223   {\led@error@fail@patch@endminipage}
8224
8225 %

```

`\l@dunboxmpfootgroup@beforenormalnotes` `\ldunboxmpfoot` insert normal footnotes for ledgroup.

```

8226 \newcommand*{\l@dunboxmpfoot}{%
8227   \vskip\skip\@mpfootins
8228   \normalcolor
8229   \footnoterule
8230   \l@advance@parledgroup@beforenormalnotes
8231   \unvbox\@mpfootins%
8232 }
8233 %

```

When using parallel ledgroup, we need to store the vertical space added before footnote, in order to compensate them between left and right pages.

```

8234 \newcommand{\l@advance@parledgroup@beforenormalnotes}{%
8235   \ifparledgroup
8236     \ifl@pairing
8237       \ifledRcol
8238         \dingdef{\parledgroup@beforenotesR}{\parledgroup@beforenotesR+\
skip\@mpfootins}
8239       \else

```

```

8240 \dimdef{\parledgroup@beforenotesL}{\parledgroup@beforenotesL+
skip\@mpfootins}
8241 \fi
8242 \fi
8243 \fi
8244 }
8245 %

```

`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.

```

8246 \newenvironment{ledgroup}{%
8247 \resetprevpage@num%
8248 \def\@mpfn{mpfootnote}\def\thempfn{\thempfootnote}\c@mpfootnote\z@%
8249 \let\@footnotetext\@mpfootnotetext
8250 \l@dfeetbeginmini%
8251 }{%
8252 \par
8253 \unskip
8254 \ifvoid\@mpfootins\else
8255 \l@dunboxmpfoot
8256 \fi
8257 \l@dfeetendmini%
8258 \@ledgroupfalse%
8259 }
8260 }
8261 %
8262 %
8263 %

```

`ledgroupsize` `\begin{ledgroupsize} [pos] {width}`

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 *width* minipage. The optional *pos* controls the sideways position of numbered text.

```

8264 \newenvironment{ledgroupsize}[2][1]{%
8265 %

```

Set the various text measures.

```

8266 \hsize #2\relax
8267 %

```

Initialize fills for centering.

```

8268 \let\ledllfill\hfil
8269 \let\ledrlfill\hfil
8270 \def\@tempa{#1}\def\@tempb{1}%
8271 %

```

Left adjusted numbered lines

```

8272 \ifx\@tempa\@tempb
8273 \let\ledllfill\relax
8274 \else
8275 \def\@tempb{r}%

```

```

8276 \ifx\@tempa\@tempb
8277 %

Right adjusted numbered lines

8278 \let\ledrlfill\relax
8279 \fi
8280 \fi
8281 %

Set up the footnoting.

8282 \def\@mpfn{mpfootnote}\def\thempfn{\thempfootnote}\c@mpfootnote\z@
8283 \let\@footnotetext\@mpfootnotetext
8284 \l@dfetbeginmini%
8285 }-%
8286 \par
8287 \unskip
8288 \ifvoid\@mpfootins\else
8289 \l@dunboxmpfoot
8290 \fi
8291 \l@dfetendmini%
8292 }
8293 %
8294 %

Close the \ifnoledgroup@else.

8295 \fi%
8296 %

```

`\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`.

```

8297 \newif\ifledgroupnotesL@
8298 \newif\ifledgroupnotesR@
8299 %

```

## 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`.

```

8300 \AtBeginDocument{%
8301 \unless\ifl@imakeidx%
8302 \@ifpackageloaded{imakeidx}{\led@error@PackageAfterEledmac{imakeidx}}{}
8303 %
8304 \fi%
8305 \unless\ifl@indextools%
8306 \@ifpackageloaded{indextools}{\led@error@PackageAfterEledmac{indextools}}{}%
8307 \fi%

```

```

8307 \unless\ifl@footmisc%
8308 \ifpackageloaded{footmisc}{\led@error@PackageAfterEledmac{footmisc}}{}
%
8309 \fi%
8310 }
8311 %

```

## 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
8312 \newcommand{\pagelinesep}{-}
8313 \newcommand{\edindexlab}{$&}
8314 \newcounter{labidx}
8315 \setcounter{labidx}{0}
8316
8317 %

```

`\doedindexlabel` This macro sets an `\edlabel`.

```

8318 \newcommand{\doedindexlabel}{%
8319 \stepcounter{labidx}%
8320 \edlabel{\edindexlab\thelabidx}%
8321 }
8322
8323 %

```

`\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`.

```

8324 \newcounter{pageline}%
8325 \renewcommand{\thepageline}{%
8326 \thepage%
8327 \pagelinesep%
8328 \xlineref{\edindexlab\thelabidx}%
8329 }
8330 %

```

`\thestartpageline` These macros make up the page/line start/end number when the `\edindex` command  
`\theendpageline` is called in critical notes.

```

8331 \newcommand{\thestartpageline}{%
8332 \l@dparsedstartpage%
8333 \pagelinesep%
8334 \l@dparsedstartline%
8335 }
8336 \newcommand{\theendpageline}{%
8337 \l@dparsedendpage%
8338 \pagelinesep%
8339 \l@dparsedendline%
8340 }
8341 %

```



**XXVIII.3 Code specific to `\edindex` in critical footnotes**

`\@edindex@fornote@` This counter is incremented at the beginning of each note (either a footnote or a sidenote), and decremented at the end of each note. If its value is greater than 0, that means we are inside a note.

```
8342 \newcount\@edindex@fornote@
8343 %
```

`\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 |.

```
8344 \newcommand{\prepare@edindex@fornote}[1]{%
8345   \l@dp@rsefootspec#1|}%
8346   \advance\@edindex@fornote@\@ne%
8347 }
8348 %
```

`\index@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.

```
8349 \newcommand{\get@edindex@ledinnote@command}{%
8350   \ifxindy@%
8351     \gdef\@ledinnote@command{%
8352       ledinnote\thelabidx%
8353     }%
8354     \ifxindyhyperref@%
8355       \immediate\write\eledmac@xindy@out{%
8356         (define-attributes ("ledinnote\thelabidx"))^^J
8357         \space\space(markup-locref^^J
8358           \eledmacmarkuplocrefdepth^^J
8359           :open "\string\ledinnote[\edindexlab\thelabidx]{\@index@command
8360             }{\@currentseries}{\@currentnotetype}"^^J
8361           :close "}"^^J
8362           :attr "ledinnote\thelabidx"^^J
8363         )
8364       }%
8365     \else%
8366       \immediate\write\eledmac@xindy@out{%
8367         (define-attributes ("ledinnote\thelabidx"))^^J
8368         \space\space(markup-locref^^J
8369           \eledmacmarkuplocrefdepth^^J
8370           :open "\string\ledinnote{\@index@command}{\@currentseries}{\
8371             \@currentnotetype}"^^J
8372           :close "}"^^J
8373           :attr "ledinnote\thelabidx"^^J
8374         )
8375       }%
8376     \fi%
8377 %
```

If we do not use xindy option, `\@ledinnote@command` will produce something like `ledinnote{formattingcommand}`.

```

8376 \else%
8377   \gdef\@ledinnote@command{%
8378     ledinnote[\@indexlab\thelabidx]{\@index@command}{\@currentseries}{\@
@currentnotetype}%
8379   }%
8380 \fi%
8381 }
8382 %

```

#### 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 |.

```

8383 \def\get@index@command#1|#2+{%
8384   \gdef\@index@txt{#1}%
8385   \gdef\@index@command{#2}%
8386   \xdef\@index@parenthesis{}%
8387   \IfBeginWith{\@index@command}{(}{%
8388     \StrGobbleLeft{\@index@command}{1}[\@index@command@]%
8389     \global\let\@index@command\@index@command@%
8390     \xdef\@index@parenthesis{(%}%
8391   ){}%
8392   \IfBeginWith{\@index@command}{)}{%
8393     \StrGobbleLeft{\@index@command}{1}[\@index@command@]%
8394     \global\let\@index@command\@index@command@%
8395     \xdef\@index@parenthesis{)%}%
8396   ){}%
8397 }
8398 %

```

#### 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.

```

8399 \newcommandx{\ledinnote}[5][1,usedefault]{%
8400   \IfStrEq{#4}{critical}{%
8401     \def\tmp{Xinnotemark}%
8402   }{%
8403     \def\tmp{innotemarkX}%
8404   }%

```

```

8405 \ifboolexpr{%
8406   test{\ifdefequal{\iftrue}{\ifHy@hyperindex}}%
8407   or%
8408   bool {xindyhyperref@}%
8409   }%
8410   {%
8411     \csuse{#2}{\hyperlink{#1}{\csuse{\tmp @#3}{#5}}}%
8412   }%
8413   {%
8414     \csuse{#2}{\csuse{\tmp @#3}{#5}}%
8415   }%
8416 }%
8417 \newcommand{\ledinnotemark}[1]{#1\emph{n}}%
8418 %

```

## 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
8419 \newcommandx{\@wredindex}[2][1=\expandonce\jobname,usedefault]{%#1 = the
index name, #2 = the text
8420 \begingroup%
8421 \let\emph\@firstofone%
8422 \let\textbf\@firstofone%
8423 \let\textit\@firstofone%
8424 \let\textmd\@firstofone%
8425 \let\textnormal\@firstofone%
8426 \let\textrm\@firstofone%
8427 \let\textsc\@firstofone%
8428 \let\textsf\@firstofone%
8429 \let\textsl\@firstofone%
8430 \let\texttt\@firstofone%
8431 \let\textup\@firstofone%
8432 \xdef\@tmp{#2}%To be used in IfSubStr instead of #2 directly. Avoid
some expansion bugs (for example with \edindex{textsc{something}})
8433 \endgroup%
8434 \ifl@imakeidx%
8435 \ifnum\@edindex@fornote@>\z@%
8436 \IfSubStr[1]{\@tmp}{|}{\get@index@command#2+}{\get@index@command#2|+}
%
8437 \get@edindex@ledinnote@command%
8438 \expandafter\imki@wrindexentry{#1}{\@index@txt|(\@ledinnote@command
){\thestartpageline}%
8439 \expandafter\imki@wrindexentry{#1}{\@index@txt|)\@ledinnote@command
}{\theendpageline}%

```

```

8440 \else%
8441 \get@edindex@hyperref{#2}%
8442 \imki@wrindexentry{#1}{\@index@txt\@edindex@hyperref}{\thepageline}%
8443 \fi%
8444 \else%
8445 \ifnum\@edindex@fornote@>\z@%
8446 \IfSubStr[1]{\@tmp}{|}{\get@index@command#2+}{\get@index@command#2|+}
%
8447 \get@edindex@ledinnote@command%
8448 \expandafter\protected@write\@indexfile{}%
8449 {\string\indexentry{\@index@txt|(\@ledinnote@command){\thestartpageline}
8450 }%
8451 \expandafter\protected@write\@indexfile{}%
8452 {\string\indexentry{\@index@txt|)\@ledinnote@command}{\theendpageline}
8453 }%
8454 \else%
8455 \protected@write\@indexfile{}%
8456 {\string\indexentry{#2}{\thepageline}
8457 }%
8458 \fi%
8459 \fi%
8460 \endgroup
8461 \@esphack%
8462 }
8463 %

```

Need to add the definition of `\edindex` to `\makeindex`, and initialise `\edindex` to do nothing.

```

8464 \pretocmd{\makeindex}{%
8465 \def\edindex{%
8466 \ifboolexpr{bool{numbering} or bool{numberingR} or bool{
8467 l@dprintingpages} or bool{l@dprintingcolumns}}{%
8468 \@bsphack%
8469 \doedindexlabel%
8470 \begingroup%
8471 \@sanitize%
8472 \wredindex%
8473 }%
8474 {%
8475 \led@warn@edinde@outsidenumbering%
8476 \index%
8477 }%
8478 }%
8479 }%
8480 {\led@error@fail@patch@makeindex}%
8481 \newcommand{\edindex}[1]{\@bsphack\@esphack}
8482 \newcommand{\dummy@edindex}[2][1=\expandonce\jobname,usedefault]{}%
8483 \newcommand{\disable@edindex}{\let\old@edindex\edindex\let\edindex\
8484 dummy@edindex}%
8485 \newcommand{\restore@edindex}{\let\edindex\old@edindex}%
8486 %
8487 %

```

**XXVIII.7 Hyperlink**

`\hyperlinkformat` `\hyperlinkformat` command is to be used to have both a internal hyperlink and a format, when indexing.

```
8486 \newcommand{\hyperlinkformat}[3]{%
8487   \ifstrempy{#1}%
8488     {\hyperlink{#2}{#3}}%
8489     {\csuse{#1}{\hyperlink{#2}{#3}}}%
8490   }%
8491 }
```

`\hyperlinkR` `\hyperlinkR` command is to be used to create a internal hyperlink and `\ledRflag`, when indexing.

```
8492 \newcommand{\hyperlinkR}[2]{%
8493   \hyperlink{#1}{#2\@Rlineflag}%
8494 }%
8495 %
8496 %
```

`\hyperlinkformatR` `\hyperlinkformatR` command is to be used to create a internal hyperlink, a format and a `\@Rlineflag`, when indexing.

```
8497 \newcommand{\hyperlinkformatR}[3]{%
8498   \hyperlinkformat{#1}{#2}{#3\@Rlineflag}%
8499 }%
8500 %
8501 %
```

`\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`).

```
8502 \newcommand{\get@edindex@hyperref}[1]{%
8503 }
```

We have to disable temporary spaces to work through a `xstring` bug (or feature?)

```
8504 \edef\temp@{%
8505   \catcode`\ =9 %space need for catcode
8506   \detokenize{#1}%For active character in unicode
8507   \catcode`\ =10 % space need for catcode
8508 }%
8509 %
```

Now, we define `\@edindex@hyperref` if the hyperindex of `hyperref` is enabled.

```
8510 \ifdefequal{\iftrue}{\ifHy@hyperindex}{%
8511   \IfSubStr{\temp@}{|}%
8512     {\get@index@command#1+%
8513     \ifledRcol%
8514       \gdef\@edindex@hyperref{|\@index@parenthesis %space kept
8515       hyperlinkformatR{\@index@command}%
8516       {\edindexlab\thelabidx}}%
8517     \else%
8518       \gdef\@edindex@hyperref{|\@index@parenthesis %space kept
```

```

8519     hyperlinkformat{\@index@command}%
8520     {\edindexlab\thelabidx}}%
8521     \fi%
8522     }%
8523     {\get@index@command#1|+%
8524     \ifledRcol%
8525     \gdef\@edindex@hyperref{|hyperlinkR{\edindexlab\thelabidx}}%
8526     \else%
8527     \gdef\@edindex@hyperref{|hyperlink{\edindexlab\thelabidx}}%
8528     \fi%
8529     }%
8530     }%
8531 %

8532 % If we use both xindy and hyperref, first get the \protect\cs{
index@command} command.
8533 % Then define \protect\cs{@edindex@hyperref} in the form \verb+eledmacXXX+
8534 % \begin{macrocode}
8535 {\ifxindyhyperref%
8536 \IfSubStr{\temp@}{|}%
8537 {\get@index@command#1+%
8538 {\get@index@command#1|+%
8539 \gdef\@edindex@hyperref{|eledmac\thelabidx}%
8540 %

```

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`.

```

8541 \IfStrEq{\@index@parenthesis}{(}%
8542 {%
8543 \csdef{xindyparenthesis@\@index@txt}{\thelabidx}%
8544 \gdef\@edindex@hyperref{| (eledmac\thelabidx}%
8545 }%
8546 {)%
8547 %

```

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`.

```

8548 \IfStrEq{\@index@parenthesis}{)}%
8549 {%
8550 \xdef\@edindex@hyperref{|)eledmac\csuse{xindyparenthesis@\
@index@txt}}%
8551 \global\csundef{xindyparenthesis@\@index@txt}%
8552 }%
8553 %

```

Write in the `.xdy` file the attributes of the location.

```

8554 {%
8555 \immediate\write\eledmac@xindy@out{%
8556 (define-attributes ("eledmac\thelabidx"))^^J
8557 \space\space(markup-locref^^J
8558 \eledmacmarkuplocrefdepth^^J

```

```

8559         :open "\string\hyperlink%
8560             \ifledRcol R\fi%
8561             {\edindexlab\thelabidx}%
8562             {\ifdefempty{\@index@command}%
8563              {}}%
8564             {\@backslashchar\@index@command}%
8565             {"^^J
8566         :close "}}^^J
8567         :attr "eledmac\thelabidx"^^J
8568     )
8569 }%
8570 }%
8571 %

```

And now, in any other case.

```

8572 \else%
8573     \gdef\@index@txt{#1}%
8574     \gdef\@edindex@hyperref{}%
8575     \fi%
8576 }%
8577 }
8578 %

```

## 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 `indextols` package. Its only argument is the note series.

```

8579 \newcommand{\led@set@index@fornote}[1]{%
8580     \ifbool{indtl@innote}%
8581         {\let\index\nindex}%
8582         {}}%
8583     \ifbool{indtl@notenumber}%
8584         {%
8585         \renewcommand{\index}[2][\indtl@jobname]{%
8586             \orig@@index[#1]{%
8587                 ##2|innotenumber{\this@footnoteX@reading}%
8588             }%
8589         }%
8590     }%
8591     {}}%
8592 }%
8593 %

```

`\led@reinit@index@fornote` The `\led@reinit@index@fornote` just reset the default value of `\index`.

```

8594 \newcommand{\led@reinit@index@fornote}{%
8595     \ifbool{indtl@innote}%
8596         {\let\index\orig@@index}%
8597         {}}%
8598     \ifbool{indtl@notenumber}%

```

```

8599   {\let\index\orig@@index}%
8600   }%
8601 }%
8602 %

```

## XXIX Glossaries

Here, we define the `\gls`-like commands prefixed by `ed`, only if the package `glossaries` is loaded.

```

8603 \AtBeginDocument{%
8604   \ifpackageloaded{glossaries}{%
8605     \@ifpackageloaded{glossaries-extra}{\
8606       led@error@require@glossariesextra}%
8607     \newif\ifl@noexpands@edgl%A boolean switched to true in \no@expands
8608   }%

```

First those which arguments are [*options*]{*label*}[*insert*].

```

8608   \gdef\edglsomo{%
8609     \listgadd{\edglsomo}{gls}%
8610     \listgadd{\edglsomo}{Gls}%
8611     \listgadd{\edglsomo}{GLS}%
8612     \listgadd{\edglsomo}{glspl}%
8613     \listgadd{\edglsomo}{Glspl}%
8614     \listgadd{\edglsomo}{GLSpl}%
8615     \listgadd{\edglsomo}{glsstext}%
8616     \listgadd{\edglsomo}{Glstext}%
8617     \listgadd{\edglsomo}{GLStext}%
8618     \listgadd{\edglsomo}{Glsfirst}%
8619     \listgadd{\edglsomo}{GLSfirst}%
8620     \listgadd{\edglsomo}{glsplural}%
8621     \listgadd{\edglsomo}{Glsplural}%
8622     \listgadd{\edglsomo}{GLSplural}%
8623     \listgadd{\edglsomo}{glsfirstplural}%
8624     \listgadd{\edglsomo}{Glsfirstplural}%
8625     \listgadd{\edglsomo}{GLSfirstplural}%
8626     \listgadd{\edglsomo}{glsname}%
8627     \listgadd{\edglsomo}{Glsname}%
8628     \listgadd{\edglsomo}{GLSname}%
8629     \listgadd{\edglsomo}{glsymbol}%
8630     \listgadd{\edglsomo}{Glsymbol}%
8631     \listgadd{\edglsomo}{GLSsymbol}%
8632     \listgadd{\edglsomo}{glsdesc}%
8633     \listgadd{\edglsomo}{Glsdesc}%
8634     \listgadd{\edglsomo}{GLSdesc}%
8635     \listgadd{\edglsomo}{glsuseri}%
8636     \listgadd{\edglsomo}{Glsuseri}%
8637     \listgadd{\edglsomo}{GLSuseri}%
8638     \listgadd{\edglsomo}{glsuserii}%
8639     \listgadd{\edglsomo}{Glsuserii}%
8640     \listgadd{\edglsomo}{GLSuserii}%
8641     \listgadd{\edglsomo}{glsuseriii}%
8642     \listgadd{\edglsomo}{Glsuseriii}%

```



```

8643 \listgadd{\edglsomo}{GLSuseriii}%
8644 \listgadd{\edglsomo}{glsuseriv}%
8645 \listgadd{\edglsomo}{Glsuseriv}%
8646 \listgadd{\edglsomo}{GLSuseriv}%
8647 \listgadd{\edglsomo}{glsuserv}%
8648 \listgadd{\edglsomo}{Glsuserv}%
8649 \listgadd{\edglsomo}{GLSuserv}%
8650 \listgadd{\edglsomo}{glsuservi}%
8651 \listgadd{\edglsomo}{Glsuservi}%
8652 \listgadd{\edglsomo}{GLSuservi}%
8653 \renewcommand{\do}[1]{%
8654 \expandafter\DeclareRobustCommand\csname ed#1\endcsname [3] [1,3,
usedefault]{%
8655 \doedindexlabel%
8656 \ifl@noexpands@edgl%
8657 \csname#1\endcsname [counter=pageline,noindex,##1]{##2}{##3}%
8658 \else%
8659 \ifdef{\hypertarget}{%
8660 \Hy@raisedlink@left{\hypertarget{pageline.\thepageline}{}}%
8661 }{%
8662 \csname#1\endcsname [counter=pageline,##1]{##2}{##3}%
8663 \fi%
8664 }%
8665 \expandafter\WithSuffix\expandafter\DeclareRobustCommand\csname ed
#1\endcsname*[3] [1,3,usedefault]{%
8666 \doedindexlabel%
8667 \ifl@noexpands@edgl%
8668 \csname#1\endcsname*[counter=pageline,noindex,##1]{##2}{##3}%
8669 \else%
8670 \csname#1\endcsname*[counter=pageline,##1]{##2}{##3}%
8671 \ifdef{\hypertarget}{%
8672 \Hy@raisedlink@left{\hypertarget{pageline.\thepageline}{}}%
8673 }{%
8674 \fi%
8675 }%
8676 }%
8677 \dolistloop{\edglsomo}%
8678 %

```

Then those which arguments are [*options*]{*label*}{*link text*}.

```

8679 \gdef\edglsomm{%
8680 \listgadd\edglsomm{glsdisp}%
8681 \listgadd\edglsomm{glslink}%
8682 \renewcommand{\do}[1]{%
8683 \expandafter\DeclareRobustCommand\csname ed#1\endcsname [3] [1,
usedefault]{%
8684 \doedindexlabel%
8685 \ifl@noexpands@edgl%
8686 \csname#1\endcsname [counter=pageline,noindex,##1]{##2}{##3}%
8687 \else%
8688 \ifdef{\hypertarget}{%
8689 \Hy@raisedlink@left{\hypertarget{pageline.\thepageline}{}}%
8690 }{%
8691 \csname#1\endcsname [counter=pageline,##1]{##2}{##3}%

```

```

8692     \fi%
8693   }%
8694   \expandafter\WithSuffix\expandafter\DeclareRobustCommandx\csname ed
#1\endcsname*[3][1,usedefault]{%
8695     \doedindexlabel%
8696     \ifl@noexpands@edgl%
8697     \csname#1\endcsname*[counter=pageline,noindex,##1]{##2}{##3}%
8698   \else%
8699     \ifdef{\hypertarget}{%
8700       \Hy@raisedlink@left{\hypertarget{pageline.\thepageline}}{}%
8701     }{}%
8702     \csname#1\endcsname*[counter=pageline,##1]{##2}{##3}%
8703   \fi%
8704 }%
8705 }%
8706 \dolistloop{\edglsomm}%
8707 %

```

Then those which arguments are [*options*]{*label*}.

```

8708   \gdef\edglsom{%
8709     \listgadd\edglsom{glsadd}%
8710     \renewcommand{\do}[1]{%
8711       \expandafter\DeclareRobustCommandx\csname ed#1\endcsname[2][1,
usedefault]{%
8712         \doedindexlabel%
8713         \ifl@noexpands@edgl%
8714         \csname#1\endcsname[counter=pageline,noindex,##1]{##2}%
8715       \else%
8716         \ifdef{\hypertarget}{%
8717           \Hy@raisedlink@left{\hypertarget{pageline.\thepageline}}{}%
8718         }{}%
8719         \csname#1\endcsname[counter=pageline,##1]{##2}%
8720       \fi%
8721     }%
8722     \expandafter\WithSuffix\expandafter\DeclareRobustCommandx\csname ed
#1\endcsname*[2][1,usedefault]{%
8723       \doedindexlabel%
8724       \ifl@noexpands@edgl%
8725       \csname#1\endcsname*[counter=pageline,noindex,##1]{##2}%
8726     \else%
8727       \ifdef{\hypertarget}{%
8728         \Hy@raisedlink@left{\hypertarget{pageline.\thepageline}}{}%
8729       }{}%
8730       \csname#1\endcsname*[counter=pageline,##1]{##2}%
8731     \fi%
8732   }%
8733 }%
8734 \dolistloop{\edglsom}%
8735 %

```

\l@noexpands@edgl is a macro to switch \ifl@noexpands@edgl to true in \no@expands. Its definition is made dynamically.

```

\l@noexpands@edgl136 \def\l@noexpands@edgl{\l@noexpands@edgltrue}%

```

```

8737 }{\let\l@noexpands@edgl\relax}%
8738 }%
8739 %

```

## 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.

```

8740 \def\@hangingsymbol{}
8741 \newcommand*\sethangingsymbol}[1]{%
8742   \gdef\@hangingsymbol{#1}%
8743 }%
8744 \newif\ifinstanza
8745 %

```

`\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.

```

8746 \newif\ifinserthangingsymbol
8747 \newcommand{\inserthangingsymbol}{%
8748   \ifinserthangingsymbol%
8749     \ifinstanza%
8750       \@hangingsymbol%
8751     \fi%
8752   \fi%
8753 }
8754 %

```

### 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.

```

8755 \newcommand*\ampersand{\char`&}
8756
8757 %

```

### 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.

```
8758 \chardef\body=\catcode`\@
8759 \catcode`\@=11
8760 \chardef\next=\catcode`\&
8761 \catcode`\&=\active
8762
8763 %
```

### 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.

```
8764 \newcount\stanza@count
8765 \newlength{\stanzaindentbase}
8766 \setlength{\stanzaindentbase}{20pt}
8767
8768 %
```

`\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.

```
8769 \def\strip@szacnt#1,#2|{\def\@tempb{#1}\def\@tempa{#2|}}
8770 \newcommand*{\setstanzavalues}[2]{\def\@tempa{#2,|}%
8771 \stanza@count\z@
8772 \def\next{\expandafter\strip@szacnt\@tempa
8773 \ifx\@tempb\empty\let\next\relax\else
8774 \expandafter\mathchardef\csname #1@\number\stanza@count
8775 \@endcsname\@tempb\relax
8776 \advance\stanza@count\@ne\fi\next}%
8777 \next}
8778
8779 %
```

`\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).

```
8780 \newcommand*{\setstanzaindents}[1]{\setstanzavalues{sza}{#1}}
8781 \newcommand*{\setstanzapenalties}[1]{\setstanzavalues{szp}{#1}}
8782 %
8783 %
```

`\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.

```

8784 \newcounter{stanzaindentsrepetition}
8785 \newcounter{startstanzaindentsrepetition}%
8786 \setcounter{startstanzaindentsrepetition}{1}%
8787 \newcount\stanza@modulo
8788
8789 \newcommand*{\managestanza@modulo}[0]{%
8790   \advance\stanza@modulo\@ne%
8791   \ifnum\stanza@modulo>\value{stanzaindentsrepetition}%
8792     \stanza@modulo\@ne%
8793   \fi%
8794 }
8795 %

```

`\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.

```

8796 \newcommand{\stanzaindent}[1]{%
8797   \hspace{\dimexpr#1\stanzaindentbase-\parindent\relax}%
8798   \ignorespaces%
8799 }%
8800 \WithSuffix\newcommand\stanzaindent*[1]{%
8801   \stanzaindent{#1}%
8802   \global\advance\stanza@modulo-\@ne%
8803   \ifnum\stanza@modulo=0%
8804     \global\stanza@modulo=\value{stanzaindentsrepetition}%
8805   \fi%
8806   \ignorespaces%
8807 }%
8808 %

```

## XXX.5 Numbering stanza

Here, macro for numbering stanza. First, the stanza counter.

```

\thestanza 09 \newcounter{stanza}
8810 \renewcommand{\thestanza}{%
8811   \textbf{\arabic{stanza}}%
8812 }
8813 %

```

`\ifnumberstanza` Then, macro to activate automatically numbering of stanza.

```

8814 \newif\ifnumberstanza%
8815 %

```

`\@insertstanza` Now, macro called at the first line of of verse of a stanza.

```

8816 \newcommand{\@insertstanza}[0]{%
8817   \ifnumberstanza%
8818     \ifl@dpairing%
8819       \ifledRcol%
8820         \stanzanumwrapper{\thestanzaR}%
8821       \else%
8822         \stanzanumwrapper{\thestanzaL}%
8823       \fi%
8824     \else%
8825       \stanzanumwrapper{\thestanza}%
8826     \fi%
8827     \setline{1}%
8828   \fi%
8829 }%
8830 %

```

`\@advancestanza` Also a command to advance the counter of stanza.

```

8831 \newcommand{\@advancestanza}[0]{%
8832   \ifnumberstanza%
8833     \ifl@dpairing%
8834       \ifledRcol%
8835         \addtocounter{stanzaR}{1}%
8836       \else%
8837         \addtocounter{stanzaL}{1}%
8838       \fi%
8839     \else%
8840       \addtocounter{stanza}{1}%
8841     \fi%
8842   \fi%
8843 }%
8844 %

```

`\stanzanumwrapper` And finally, the wrapper for stanza number

```

8845 \newcommand{\stanzanumwrapper}[1]{%
8846   \flagstanza{#1}%
8847 }%
8848 %

```

## XXX.6 Stanza number in note

Here, the command called when printing stanza number in notes.

```

8849 \newcommand{\printstanza}[0]{%
8850   \ifboolexpr{bool{l@dpairing} or bool{l@dprintingpages} or bool{
8851     l@dprintingcolumns}}{%
8852     \ifledRcol@%
8853       \thestanzaR%
8854     \else%
8855       \thestanzaL%
8856     \fi%

```

```

8856 }{%
8857   \thestanza%
8858 }%
8859 }
8860 %

```

### XXX.7 Main work

`\stanza@line` Now we arrive at the main works. `\stanza@line` sets the indentation for the line and starts a numbered paragraph—each line is treated as a paragraph. `\stanza@hang` sets the hanging indentation to be used if the stanza line requires more than one print line.

`\stanza@indent`  
`\sza@penalty`

If it is known that each stanza line will fit on one print line, it is advisable to set the hanging indentation to zero. `\sza@penalty` places the specified penalty following each stanza line. By default, this facility is turned off so that no penalty is included. However, the user may initiate these penalties to indicate good and bad places in the stanza for page breaking.

```

8861 \newcommand{\stanza@line}[2][1,2,usedefault]{%
8862   \stanza@indent%
8863   \pstart[#1][#2]\stanza@hang\ignorespaces%
8864 }%
8865 \newcommand{\stanza@indent}{%
8866   \ifboolexpr{%
8867     test {\ifnumequal{\value{stanzaindentsrepetition}}{0}}%
8868     or%
8869     test {\ifnumless{\number\stanza@count}{\value{
startstanzaindentsrepetition}}}%
8870   }{%
8871     \ifcsdef{sza@\number\stanza@count @}{%
8872       {%
8873         \parindent=\csname sza@\number\stanza@count @\endcsname\
stanzaindentbase%
8874       }{%
8875         \led@err@StanzaIndentNotDefined%
8876       }%
8877     }{%
8878       \ifcsdef{sza@\number\stanza@modulo @}{%
8879         \parindent=\csname sza@%
8880         \the\numexpr%
8881         \stanza@modulo%
8882         - 1%
8883         + \value{startstanzaindentsrepetition}%
8884         \relax%
8885         @\endcsname\stanzaindentbase%
8886         \managestanza@modulo%
8887       }%
8888       {%
8889         \led@err@StanzaIndentNotDefined%
8890       }%
8891     }%
8892   }%
8893
8894 \def\stanza@hang{%

```

```

8895 \leavevmode\startlock%
8896 \hangindent\if@RTL-\fi\expandafter%
8897 \csname sza@0@\endcsname\stanzaindentbase%
8898 \hangafter\@ne%
8899 }%
8900
8901 \def\sza@penalty{\count@\csname szp@\number\stanza@count @\endcsname
8902 \ifnum\count@>\@M\advance\count@-\@M\penalty-\else
8903 \penalty\fi\count@}
8904 %

```

**\@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 `\&`.

**\stanza**  
**\@stopstanza**  
**\AtEveryStopStanza**  
**\AtEveryStanza**  
**\AtStartEveryStanza**  
**\BeforeEveryStopStanza**  
**\newverse**

```

8905 \xdef\@startstanza[#1][#2]{%
8906 \noexpand\instanzatruer\expandafter
8907 \begingroup%
8908 \catcode`\noexpand\&\active%
8909 \unexpanded{\newdimen\parindent@beforestanza%
8910 \parindent@beforestanza=\parindent}%Keep in memory the standard \
parindent
8911 \global\stanza@count\@ne\stanza@modulo\@ne
8912 \noexpand\ifnum\expandafter\noexpand\csname sza@0@\endcsname=\z@%
8913 \def\noexpand\stanza@hang{\unexpanded{\leavevmode\null}}%
8914 \let\noexpand\endlock\relax%
8915 \noexpand\else%
8916 \interlinepenalty\@M%
8917 \unexpanded{%
8918 \ifboolexpr{bool {@RTL} and bool{xetex}}{%
8919 \leftskip\z@ plus 1fil\relax%
8920 }{%
8921 \rightskip\z@ plus 1fil\relax%
8922 }%
8923 }%
8924 \noexpand\fi%
8925 \noexpand\ifnum\expandafter\noexpand\csname szp@0@\endcsname=\z@%
8926 \let\noexpand\sza@penalty%
8927 \relax\noexpand\fi%
8928 \def\noexpand&{%
8929 \noexpand\newverse[] []}%
8930 \def\noexpand\&{\noexpand\@stopstanza}%
8931 \noexpand\@advancestanza%
8932 \noexpand\stanza@line[#1][#2]%
8933 \noexpand\@insertstanza%
8934 \let\par\relax\ignorespaces%No paragraph in verses
8935 }
8936
8937 \newcommandx{\stanza}[2][1,2,usedefault]{%

```



```

8938 \ifboolexpr{%
8939   not test{\ifdefined\at@every@stanza}}%
8940   and test{\ifstrempy{#1}}%
8941   and test{\ifstrempy{#2}}}%
8942   {\@startstanza[] [\at@every@stanza]\at@start@every@stanza}%
8943   {\@startstanza[#1] [#2]\at@start@every@stanza}%
8944 }%
8945
8946 \newcommandx{\@stopstanza}[2][1,2,usedefault]{%
8947   \unskip%
8948   \endlock%
8949   \ifboolexpr{%
8950     not test{\ifdefined\at@every@stop@stanza}}%
8951     and test{\ifstrempy{#1}}%
8952     and test{\ifstrempy{#2}}}%
8953     {\before@every@stop@stanza\pend[] [\at@every@stop@stanza]}%
8954     {\before@every@stop@stanza\pend[#1] [#2]}%
8955   \endgroup%
8956   \instanzafalse%
8957   \global\undef\parindent@beforestanza%
8958 }
8959
8960 \newcommand{\AtEveryStopStanza}[1]{%
8961   \ifstrempy{#1}%
8962     {\gdef\at@every@stop@stanza{}}%
8963     {\gdef\at@every@stop@stanza{\noindent#1}}%
8964 }%
8965 \WithSuffix\newcommand\AtEveryStopStanza*[1]{%
8966   \ifstrempy{#1}%
8967     {\gdef\at@every@stop@stanza{}}%
8968     {\gdef\at@every@stop@stanza{#1}}%
8969 }%
8970 \def\at@every@stop@stanza{}%
8971
8972 \newcommand{\AtEveryStanza}[1]{%
8973   \ifstrempy{#1}%
8974     {\gdef\at@every@stanza{}}%
8975     {\gdef\at@every@stanza{\noindent#1}}%
8976 }%
8977 \WithSuffix\newcommand\AtEveryStanza*[1]{%
8978   \ifstrempy{#1}%
8979     {\gdef\at@every@stanza{}}%
8980     {\gdef\at@every@stanza{#1}}%
8981 }%
8982
8983
8984
8985 \newcommand{\AtStartEveryStanza}[1]{%
8986   \ifstrempy{#1}%
8987     {\gdef\at@start@every@stanza{}}%
8988     {\gdef\at@start@every@stanza{#1}}%
8989 }%
8990 \def\at@start@every@stanza{}%
8991

```

```

8992 \newcommand{\BeforeEveryStopStanza}[1]{%
8993   \ifstrempy{#1}%
8994     {\gdef\before@every@stop@stanza{}}%
8995     {\gdef\before@every@stop@stanza{#1}}%
8996 }%
8997 \def\before@every@stop@stanza{}%
8998
8999 \newcommandx*\newverse}[4][1,2,3,4,usedefault]{%
9000   \unskip%
9001   \endlock\pend[#1][#3]\sza@penalty\global%
9002   \advance\stanza@count\@ne\stanza@line[#2][#4]%
9003   }
9004
9005 %

```

`\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`.

```

9006 \newcommand*\flagstanza}[2][\stanzaindentbase]{%
9007   \hskip -#1\llap{#2}\hskip #1\ignorespaces}
9008
9009 %

```

### 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.

```

9010 \catcode`\&=\next
9011 \catcode`\@=\body
9012 \setstanzavalues{szp}{0}
9013
9014 %

```

### XXX.9 Dealing with antilabe

`\firsthalf@antilabe@wd` If the `antilabe` option is enabled, after having `vboxed` the content of `\pstart...pend` structure, we get the width of the last line of the `\pend`.

```

\get@firsthalf@antilabe@wd
\antilabe
9015 \ifantilabe@%
9016   \newdimen\firsthalf@antilabe@wd%
9017   \newcommand\get@firsthalf@antilabe@wd{%
9018     \ifboolexpr{bool {instanza} or bool {inastanzaL} or bool {inastanzaR}}{
%
9019       \vbadness=10000%
9020       \splittopskip=\z@%
9021       \ifl@pairing%
9022         \ifledRcol%
9023         \setbox0=\copy\csname l@dRcolrawbox\the\l@dnumstartsR\endcsname%

```

```

9024     \else%
9025         \setbox0=\copy\csname l@dLcolrawbox\the\l@dnumstartsL\endcsname%
9026     \fi%
9027     \else%
9028         \setbox0=\copy\raw@text%
9029     \fi%
9030     \dimen0=\dimexpr\ht0+\dp0-\baselineskip\relax%
9031     \ifdim\dimen0>0pt%
9032         \setbox1=\vsplit0 to \dimen0%
9033     \fi%
9034     \setbox0=\vbox{\unvbox0\global\setbox1=\lastbox}%
9035     \setbox1=\hbox{\noindent\unhbox1}%
9036     \global\firsthalf@antilabe@wd=\wd1%
9037     }{}%
9038 }%
9039 \def\beforeantilabe{}%
9040 \def\afterantilabe{}%
9041 \newcommand{\antilabe}[1]{%
9042     \beforeantilabe\hbox to\firsthalf@antilabe@wd{#1\hfill}\afterantilabe%
9043 }%
9044 \else%
9045     \let\get@firsthalf@antilabe@wd\relax%
9046 \fi%
9047 %

```

## 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@c`

`\msdata@cR`

```

9048 \def\msdata@c{}%
9049 \def\msdata@cR{}%
9050 \newcommand{\msdata}[1]{%
9051     \leavevmode%
9052     \unless\ifstopmsdata@inserted0%
9053         \stopmsdata%
9054         \led@warning@msdatawithoutstop%
9055     \fi%
9056     \global\stopmsdata@inserted@false%
9057     \unless\ifledRcol%
9058         \numgdef{\msdata@c}{\msdata@c+1}%
9059         \ifdef{\hypertarget}{%
9060             \edlabel{\msdata@c:start:msdata}%
9061         }{}%
9062         \protected@write\linenum@out{}{%
9063             \string\msd{#1}%
9064         }%
9065     \else%
9066         \numgdef{\msdata@cR}{\msdata@cR+1}%
9067         \ifdef{\hypertarget}{%
9068             \edlabel{\msdata@cR:start:msdata}%

```

```

9069 }{}%
9070 \protected@write\linenum@outR{}{}%
9071 \string\@msd{#1}%
9072 }%
9073 \fi%
9074 }%
9075 %

```

**\stopmsdata** The user level `\stopmsdata` command only writes information about the end of manuscripts data in numbered auxiliary file.

```

9076 \newcommand{\stopmsdata}[0]{}%
9077 \leavevmode%
9078 \unless\ifledRcol%
9079 \protected@write\linenum@out{}{}%
9080 \string\@stopmsd%
9081 }%
9082 \ifdef{\hypertarget}{}%
9083 \edlabel{\msdata@c:end:msdata}%
9084 }{}%
9085 \else%
9086 \protected@write\linenum@outR{}{}%
9087 \string\@stopmsd%
9088 }%
9089 \ifdef{\hypertarget}{}%
9090 \edlabel{\msdata@cR:end:msdata}%
9091 }{}%
9092 \fi%
9093 \global\stopmsdata@inserted@true%
9094 }%
9095 %

```

**\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`

```

9096 \newif\ifstopmsdata@inserted@%
9097 %

```

## 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?

```

9098 \newcommand{\setmsdataseries}[1]{}%
9099 \gdef\@msdata@series{#1}%
9100 }%
9101 \def\@msdata@series{A}%
9102 %

```

**\setmsdataposition** The label for the manuscripts data.

```

9103 \def\ms@data@position{msdata-regular}%
9104 \newcommand{\setmsdataposition}[1]{%
9105   \gdef\ms@data@position{#1}%
9106 }%
9107 %

```

`\setmsdatalabel` The label for the manuscripts data.

```

9108 \def\ms@data@label{Ms.}%
9109 \newcommand{\setmsdatalabel}[1]{%
9110   \gdef\ms@data@label{#1}%
9111 }%
9112 %

```

### XXXI.3 Counters and lists

`\@msd@c` `\@msd@c` is a counter incremented at each `\@msd` read in auxiliary file.

```

9113 \numdef{\@msd@c}{0}
9114 \numdef{\@msd@cR}{0}
9115 %

```

`\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.

```

9116 \numdef{\add@msd@c}{0}%
9117 \numdef{\add@msd@cR}{0}%
9118 %

```

`\@msdata@list` The `\@msdata@list` will contain, for each line, the lists of command to be executed to insert the manuscripts apparatus. It will be filled on `\add@msdata` and looped on `\insert@msdata`, then emptied.

```

9119 \def\@msdata@list{}%
9120 %

```

### XXXI.4 Auxiliary file macros

`\@msd` The `\@msd` macro is written in the auxiliary file. It just defines three macros by `\msdata` macro, which allow us to know the manuscripts data, the line number and the absolute line number where it was called

It also stores the action code 1010 in the list of actions by line.

```

9121 \newcommand{\@msd}[1]{%
9122   \unless\ifledRcol%
9123     \global\numdef{\@msd@c}{\@msd@c+\@ne}%
9124     \csgdef{\@msdata@\@msd@c @data}{#1}%
9125     \csxdef{\@msdata@\@msd@c @linenumber}{\the\line@num}%
9126     \csxdef{\@msdata@\@msd@c @sublinenumber}{\the\subline@num}%
9127     \csxdef{\@msdata@\@msd@c @abslinenumber}{\the\absline@num}%
9128     \xright@appenditem{\the\absline@num}\to\actionlines@list%
9129     \xright@appenditem{-1010}\to\actions@list%
9130   \else%

```

```

9131 \global\newcommand{\@msd@cR}{\@msd@cR+\@ne}%
9132 \csgdef{\@msd@data@\@msd@cR @dataR}{#1}%
9133 \csxdef{\@msd@data@\@msd@cR @linenumberR}{\the\line@numR}%
9134 \csxdef{\@msd@data@\@msd@cR @sublinenumberR}{\the\subline@numR}%
9135 \csxdef{\@msd@data@\@msd@cR @abslinenumberR}{\the\absline@numR}%
9136 \xright@appenditem{\the\absline@numR}\to\actionlines@listR%
9137 \xright@appenditem{-1010}\to\actions@listR%
9138 \fi%
9139 }%
9140 %

```

`\@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.

```

9141 \newcommand{\@stopmsd}[0]{%
9142   \unless\ifledRcol%
9143     \ifcsundef{\@msd@data@\@msd@c @stoplinenumberR}{%
9144       \csxdef{\@msd@data@\@msd@c @stopabslinenumberR}{\the\absline@numR}%
9145       \csxdef{\@msd@data@\@msd@c @stoplinenumberR}{\the\line@numR}%
9146       \csxdef{\@msd@data@\@msd@c @stopsublinenumberR}{\the\subline@numR}%
9147     }%
9148   \else%
9149     \ifcsundef{\@msd@data@\@msd@cR @stoplinenumberR}{%
9150       \csxdef{\@msd@data@\@msd@cR @stopabslinenumberR}{\the\absline@numR}%
9151       \csxdef{\@msd@data@\@msd@cR @stoplinenumberR}{\the\line@numR}%
9152       \csxdef{\@msd@data@\@msd@cR @stopsublinenumberR}{\the\subline@numR}%
9153     }%
9154   }%
9155 \fi%
9156 }%
9157 %

```

### 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`.

```

9158 \newcommand{\add@msdata}{%
9159   \bgroup%
9160   \normalfont%
9161   \unless\ifledRcol%
9162     \numgdef{\add@msd@c}{\add@msd@c+\@ne}%
9163     \ifcsdef{\@msd@data@\add@msd@c @data}{%
9164       \letcs{\@data}{\@msd@data@\add@msd@c @data}%
9165       \edef\l@d@nums{%
9166         000| % Start page = we don't print it
9167         \csuse{\@msd@data@\add@msd@c @linenumberR}| % Start line number
9168         \csuse{\@msd@data@\add@msd@c @sublinenumberR}| % Start subline number
9169         000| % End page number, we don't print it
9170         \ifnumless{\csuse{\@msd@data@\add@msd@c @stopabslinenumberR}}{\csuse{
@lastabsline@forpage@\the\page@num}}}%

```

```

9171         {\csuse{@msdata@add@msd@c @stoptineline}}%End line number if
           in the same page
9172         {\csuse{@lastline@forpage@the\page@num}}%Otherwise, last
           number of the page
9173         |%
9174         \ifnumless{\csuse{@msdata@add@msd@c @stopabsline}}{\csuse{
           @lastabsline@forpage@the\page@num}}%
9175         {\csuse{@msdata@add@msd@c @stopsubline}}%End subline
           number if in the same page
9176         {\csuse{@lastsubline@forpage@the\page@num}}%Otherwise, last
           subnumber of the page
9177         |%
9178         \edfont@info%Font
9179         }%
9180         \@msd@options@fullpagefalse%
9181         \if@firstlineofpage%Try if the data are for the full page. If yes
           , will add options to the list.
9182         \unless\if@msdata@insertedfrompreviouspage%
9183         \ifnumless{\csuse{@lastabsline@forpage@the\page@num}}{\csuse{
           @msdata@add@msd@c @stopabsline}+\@one}%
9184         {%
9185         \numdef{\@tmp}{\add@msd@c+\@one}%
9186         \ifcsdef{@msdata@\@tmp @absline}%
9187         {\ifnumequal{\csuse{@msdata@\@tmp @absline}}{\csuse{
           @lastabsline@forpage@the\page@num}}%
9188         {}%
9189         {\@msd@options@fullpagetrue}%
9190         }%
9191         {\@msd@options@fullpagetrue}%
9192         }%
9193         {}%
9194         \fi%
9195         \fi%
9196         \listxadd{\@msdata@list}{%
9197         \@msd@options@iffullpage%
9198         \ifluatex%
9199         \xdef\footnote@luatextextdir{\the\textdir}%
9200         \xdef\footnote@luatexpardir{\the\pardir}%
9201         \fi%
9202         \csdef{@this@crossref@start}{\add@msd@c:start:msdata}%
9203         \csdef{@this@crossref@end}{\add@msd@c:end:msdata}%
9204         \unexpanded{%
9205         \def\annot@start{}%
9206         \def\annot@end{}%
9207         }%
9208         \noexpand\csuse{v\@msdata@series footnote}{\@msdata@series}{\@
           expandonce\l@d@nums}{\@msdata@label}{\expandonce\data}}%
9209         \reset@msd@options@iffullpage%
9210         }%
9211         }%
9212         {}%
9213         \else%
9214         \numgdef{\add@msd@cR}{\add@msd@cR+\@one}%
9215         \ifcsdef{@msdata@add@msd@cR @dataR}{%

```

```

9216 \letcs{\@data}{@msdata@add@msd@cR @dataR}%
9217 \edef\l@d@nums{%
9218 000|% Start page = we don't print it
9219 \csuse{@msdata@add@msd@cR @linenumberR}|% Start line number
9220 \csuse{@msdata@add@msd@cR @sublinenumberR}|% Start subline
number, for now, not used
9221 000|% End page number, we don't print it
9222 \ifnumless{\csuse{@msdata@add@msd@cR @stopabslinenumberR}}{\
csuse{@lastline@forpageR@the\page@numR}}%
9223 {\csuse{@msdata@add@msd@cR @stoplinenumberR}}%End line number
if in the same page
9224 {\csuse{@lastline@forpageR@the\page@numR}}%Otherwise, last
number of the page
9225 |%
9226 \ifnumless{\csuse{@msdata@add@msd@cR @stopabslinenumberR}}{\
csuse{@lastline@forpageR@the\page@numR}}%
9227 {\csuse{@msdata@add@msd@cR @stopsublinenumberR}}%End subline
number if in the same page
9228 {\csuse{@lastsubline@forpageR@the\page@numR}}%Otherwise, last
number of the page
9229 000|% End sub line number, for now, not used
9230 \edfont@info%Font
9231 }%
9232 \@msd@options@fullpagefalse%
9233 \if@firstlineofpageR%
9234 \unless@if@msdata@insertedfrompreviouspage%
9235 \ifnumless{\csuse{@lastabsline@forpageR@the\page@numR}}{\
csuse{@msdata@add@msd@c @stopabslinenumberR}+\@ne}%
9236 {%
9237 \numdef{\@tmp}{\add@msd@cR+\@ne}%
9238 \ifcsdef{@msdata@\@tmp @abslinenumberR}%
9239 {\ifnumequal{\csuse{@msdata@\@tmp @abslinenumberR}}{\csuse{
@lastabsline@forpageR@the\page@numR}}%
9240 {}%
9241 {\@msd@options@fullpagetrue}%
9242 }%
9243 {\@msd@options@fullpagetrue}%
9244 }%
9245 {}%
9246 \fi%
9247 \fi%
9248 \listxadd{\@msdata@list}{%
9249 \@msd@options@iffullpage%
9250 \ifluatex%
9251 \csxdef{footnote@luatextextdir}{\the\textdir}%
9252 \csxdef{footnote@luatexpardir}{\the\pardir}%
9253 \fi%
9254 \csdef{@this@crossref@start}{\add@msd@cR:start:msdata}%
9255 \csdef{@this@crossref@end}{\add@msd@cR:end:msdata}%
9256 \unexpanded{%
9257 \def\annot@start{}%
9258 \def\annot@end{}%
9259 }%

```



```

9260 \noexpand\csuse{v\@msdata@series footnote}{\@msdata@series}{\{
expandonce\l@d@nums}{\ms@data@label}{\expandonce\data}}%
9261 \reset@msd@options@iffullpage%
9262 }%
9263 }%
9264 }%
9265 \fi%
9266 \egroup%
9267 }%
9268 %

```

`\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.

```

9269 \newif\if@msdata@insertedfrompreviouspage%
9270 %

```

`\@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.

```

9271 \newcommand{\add@msdata@firstlineofpage}{%
9272 \bgroup%
9273 \normalfont%
9274 \unless\ifledRcol{%
9275 \ifcsdef{@msdata@\add@msd@c @data}{%
9276 \ifnumless{\the\absline@num-\@ne}{\csuse{@msdata@\add@msd@c
@stopabslinenumber}}}%
9277 {%
9278 \global\@msdata@insertedfrompreviouspagetrue%
9279 \letcs{\@data}{@msdata@\add@msd@c @data}%
9280 \edef\l@d@nums{%
9281 000|% Start page = we don't print it
9282 \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
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@c @stopabslinenumber}}{\csuse{
@lastabsline@forpage@the\page@num}}}%
9286 {\csuse{@msdata@\add@msd@c @stoplinenumber}}%End line number if
in the same page
9287 {\csuse{@lastline@forpage@the\page@num}}%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@forpage@the\page@num}}{\csuse{
@msdata@\add@msd@c @stopabslinenumber}+\@ne}%We will test if the ms data is
for the full page
9294 {%
9295 \numdef{\@tmp}{\add@msd@c+\@ne}%
9296 \ifcsdef{@msdata@\@tmp @abslinenumber}%

```

```

9297         {\ifnumequal{\csuse{@msdata@\tmp @abslinenumber}}{\csuse{
@lastabsline@forpage@\the\page@num}}}%
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@c:start:msdata}%
9311 \csdef{@this@crossref@end}{\add@msd@c: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 \else%
9323 \ifcsdef{@msdata@\add@msd@cR @dataR}{%
9324 \ifnumless{\the\absline@numR-\@one}{\csuse{@msdata@\add@msd@cR
@stopabslinenumberR}}%
9325     {%
9326     \global\@msdata@insertedfrompreviouspagetrue%
9327     \letcs{\@data}{@msdata@\add@msd@cR @dataR}%
9328     \edef\l@d@nums{%
9329     000|% Start page = we don't print it
9330     \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
9331     000|% Start subline number, for now, not used
9332     000|% End page number, we don't print it
9333     \ifnumless{\csuse{@msdata@\add@msd@cR @stopabslinenumberR}}{\@
csuse{@lastline@forpageR@\the\page@numR}}%
9334     {\csuse{@msdata@\add@msd@cR @stoplinenumberR}}%End line number
if in the same page
9335     {\csuse{@lastline@forpageR@\the\page@numR}}%Otherwise, last
number of the page
9336     |%
9337     000|% End sub line number, for now, not used
9338     \edfont@info%Font
9339     }%
9340     \@msd@options@fullpagefalse%
9341     \ifnumless{\csuse{@lastabsline@forpageR@\the\page@numR}}{\csuse{
@msdata@\add@msd@cR @stopabslinenumberR}+\@one}%

```

```

9342     {%
9343     \numdef{\@tmp}{\add@msd@cR+\@ne}%
9344     \ifcsdef{@msdata@\@tmp @abslinenumberR}%
9345     {\ifnumequal{\csuse{@msdata@\@tmp @abslinenumberR}}{\csuse{
@lastabsline@forpageR@\the\page@numR}}}%
9346     {}%
9347     {\@msd@options@fullpagetrue}%
9348     }%
9349     {\@msd@options@fullpagetrue}%
9350     }%
9351     {}%
9352     \listxadd{\@msdata@list}{%
9353     \@msd@options@iffullpage%
9354     \ifluatex%
9355     \csxdef{footnote@luatextextdir}{\the\textdir}%
9356     \csxdef{footnote@luatexpardir}{\the\pardir}%
9357     \fi%
9358     \csdef{@this@crossref@start}{\add@msd@cR:start:msdata}%
9359     \csdef{@this@crossref@end}{\add@msd@cR:end:msdata}%
9360     \unexpanded{%
9361     \def\annot@start{}}%
9362     \def\annot@end{}}%
9363     }%
9364     \noexpand\csuse{v\@msdata@series footnote}{\@msdata@series}{{\
expandonce\l@d@nums}{\ms@data@label}{\expandonce\data}}%
9365     \reset@msd@options@iffullpage%
9366     }%
9367     }%
9368     {\global\@msdata@insertedfrompreviouspagefalse}%
9369     }{}%
9370     \fi%
9371     \egroup%
9372 }%
9373 %

```

## XXXI.6 Inserting footnote

Just before inserting standard insert (familiar and critical footnotes, sidenotes), we call `\insert@msdata` to insert manuscripts data's footnotes.

```

\insert@msdata 74 \newcommand{\insert@msdata}{%
9375     \def\do##1{##1}%
9376     \dolistloop{\@msdata@list}%
9377     \global\let\@msdata@list\relax%
9378 }%
9379 %

```

## 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.

```

9380 \newif\if@msd@options@fullpage%
9381 \newcommand{\@msd@options@iffullpage}[0]{%
9382   \if@msd@options@fullpage%
9383     \noexpand\toggletrue{nonum@}%
9384     \ifdefvoid{\ms@data@label}%
9385       {\noexpand\toggletrue{nosep@}}%
9386     {}%
9387   \fi%
9388 }%
9389 \newcommand{\reset@msd@options@iffullpage}[0]{%
9390   \noexpand\togglefalse{nonum@}%
9391   \noexpand\togglefalse{nosep@}%
9392 }%
9393 %

```

## XXXII Arrays and tables

### XXXII.1 Preamble: macro as environment

The following is borrowed, and renamed, from the `amsmath` package. See also the CTT thread “`eq` and `amstex`”, 1995/08/31, started by Keith Reckdahl and ended definitively by David M. Jones.

Several of the `[math]` macros scan their body twice. This means we must collect all text in the body of an environment form before calling the macro.

`\@emptytoks` This is actually defined in the `amsgen` package.

```

9394 \newtoks\@emptytoks
9395 %
9396 %

```

The rest is from `amsmath`.

`\l@denvbody` A token register to contain the body.

```

9397 \newtoks\l@denvbody
9398 %
9399 %

```

`\addtol@denvbody` `\addtol@denvdody{arg}` adds `arg` to the token register `\l@denvbody`.

```

9400 \newcommand{\addtol@denvbody}[1]{%
9401   \global\l@denvbody\expandafter{\the\l@denvbody#1}}
9402 %

```

`\if@edtabular` A boolean set to true inside a `reledmac`’ tabular environment.

```

9403 \newif\if@edtabular%
9404 %

```

`\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`.

```

9405 \newcommand{\l@dcollect@body}[1]{%
9406   \ifvmode\else%
9407   \\\%Ensure that we do not have spurious indentation at the very first
line, caused by some text before.
9408   \fi%
9409   \l@denvbody{\expandafter#1\expandafter{\the\l@denvbody}}%
9410   \edef\processl@denvbody{\the\l@denvbody\noexpand\end{\@currentenv}}%
9411   \l@denvbody\@emptytoks \def\l@dbegin@stack{b}%
9412   \begingroup
9413   \@edtabulartrue%
9414   \expandafter\let\csname\@currentenv\endcsname\l@dcollect@body
9415   \edef\processl@denvbody{\expandafter\noexpand\csname\@currentenv\
endcsname}%
9416   \processl@denvbody%
9417   }%
9418
9419 %

```

`\l@dpush@begins` When adding a piece of the current environment's contents to `\l@denvbody`, we scan it to check for additional `\begin` tokens, and add a "b" to the stack for any that we find.

```

9420 \def\l@dpush@begins#1\begin#2{%
9421   \ifx\end#2\else b\expandafter\l@dpush@begins\fi}
9422
9423 %

```

`\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.

```

9424 \def\l@dcollect@@body#1\end#2{%
9425   \edef\l@dbegin@stack{\l@dpush@begins#1\begin\end
9426     \expandafter\@gobble\l@dbegin@stack}%
9427   \ifx\@empty\l@dbegin@stack
9428     \endgroup
9429     \@checkend{#2}%
9430     \addtol@denvbody{#1}%
9431   \else
9432     \addtol@denvbody{#1\end{#2}}%
9433   \fi
9434   \processl@denvbody % A little tricky! Note the grouping
9435 }
9436
9437 %

```

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.

```

9438 \newcommand*{\l@dtabnoexpands}{%
9439   \let\rtab=0%
9440   \let\ctab=0%
9441   \let\ltab=0%
9442   \let\rtabtext=0%
9443   \let\ltabtext=0%
9444   \let\ctabtext=0%
9445   \let\edbeforetab=0%
9446   \let\edaftertab=0%
9447   \let\edatleft=0%
9448   \let\edatright=0%
9449   \let\edvertline=0%
9450   \let\edvertdots=0%
9451   \let\edrowfill=0%
9452 }
9453
9454 %

```

`\disable@familiarnotes`    Macros to disable and restore familiar notes, to prevent them from printing multiple times in edtabularx and edarrayx environments.

`\restore@familiarnotes`

```

9455 \newcommand{\disable@familiarnotes}{%
9456   \unless\ifnofamiliar@%
9457     \def\do##1{%
9458       \csletcs{footnote@##1}{footnote##1}%
9459       \expandafter\renewcommand \csname footnote##1\endcsname[1]{%
9460         \protected@csxdef{@thefnmark##1}{\csuse{thefootnote##1}}%
9461         \csuse{@footnotemark##1}%
9462       }%
9463     }%
9464     \dolistloop{\@series}%
9465   \fi%
9466 }%
9467 \newcommand{\restore@familiarnotes}{%
9468   \unless\ifnofamiliar@%
9469     \def\do##1{%
9470       \csletcs{footnote##1}{footnote@##1}%
9471     }%
9472     \dolistloop{\@series}%
9473   \fi%
9474 }%
9475
9476 %

```

`\disable@sidenotes`    The same for sidenotes.

`\restore@sidenotes`

```

9477 \newcommand{\disable@sidenotes}{%
9478   \let\@ledrightnote\ledrightnote%
9479   \let\@ledleftnote\ledleftnote%
9480   \let\@ledsidenote\ledsidenote%
9481   \let\ledrightnote@gobble%
9482   \let\ledleftnote@gobble%
9483   \let\ledsidenote@gobble%
9484 }%

```



```

9485 \newcommand{\restore@sidenotes}{%
9486   \let\ledrightnote\@ledrightnote%
9487   \let\ledleftnote\@ledleftnote%
9488   \let\ledsidenote\@ledsidenote%
9489 }%
9490 %

```

**\disable@endnotes** The same for endnotes.

**\restore@endnotes**

```

9491 \newcommand{\disable@endnotes}{%
9492   \unless\ifnoend0%
9493     \def\do##1{%
9494       \csletcs{##1@endnote}{##1endnote}%
9495       \expandafter\renewcommand \csname ##1endnote\endcsname [1]{}%
9496     }%
9497     \dolistloop{\@series}%
9498   \fi%
9499 }%
9500 \newcommand{\restore@endnotes}{%
9501   \unless\ifnofamiliar0%
9502     \def\do##1{%
9503       \csletcs{##1endnote}{##1@endnote}%
9504     }%
9505     \dolistloop{\@series}%
9506   \fi%
9507 }%
9508 %

```

**\disable@notes** Disable/restore side, familiar and end notes.

**\restore@notes**

```

9509 \newcommand{\disable@notes}{%
9510   \disable@sidenotes%
9511   \disable@familiarnotes%
9512   \disable@endnotes%
9513 }%
9514 \newcommand{\restore@notes}{%
9515   \restore@sidenotes%
9516   \restore@familiarnotes%
9517   \restore@endnotes%
9518 }%
9519 %

```

**\EDTEXT** We need to be able to modify the `\edtext` macros and also restore their original definitions.

**\xedtext**

```

9520 \let\EDTEXT=\edtext
9521 \newcommand{\xedtext}[2]{\EDTEXT{#1}{#2}}
9522 %

```

**\disable@edlabel** We need to be able to modify and restore the `\edlabel` macro.

**\restore@edlabel**

```

9523 \let\original@edlabel\edlabel%
9524 \newcommand{\disable@edlabel}{%
9525   \let\edlabel\@gobble%

```

```

9526 }%
9527 \newcommand{\restore@edlabel}{%
9528   \let\edlabel\original@edlabel%
9529 }%
9530 %

```

**\xedindex** Macros supporting modification and restoration of `\edindex`.

**\nulledindex**

```

9531 \AtBeginDocument{\let\xedindex\edindex}%
9532 \newcommand{\nulledindex}[2][\jobname]{\@bsphack\@esphack}
9533
9534 %

```

**\@line@@num** Macro supporting restoration of `\linenum`.

```

9535 \let\@line@@num=\linenum
9536 %

```

**\l@dgobbleoptarg** `\l@dgobbleoptarg[⟨arg⟩]{⟨arg⟩}` replaces these two arguments (first is optional) by `\relax`.

```

9537 \newcommand*{\l@dgobbleoptarg}[2][\relax]%
9538
9539 %

```

**\l@secondmandarg** `\l@secondoptarg[⟨arg⟩]{⟨arg⟩}` gobble the first (optional) argument, and expand to the second (mandatory) argument.

```

9540 \NewExpandableDocumentCommand{\l@secondmandarg}{om}{#2}%
9541 %

```

**\Relax**<sub>42</sub> `\let\Relax=\relax`

**\NEXT**<sub>43</sub> `\let\NEXT=\next`

```

9544
9545 %

```

**\l@dmodforedtext** Modify and restore various macros for when `\edtext` is used.

**\l@drestoreforedtext**

```

9546 \newcommand{\l@dmodforedtext}{%
9547   \let\edtext\relax
9548   \def\do##1{\global\csletcs{##1footnote}{\l@dgobbleoptarg}}%
9549   \dolistloop{\@series}%
9550   \let\edindex\nulledinde%
9551   \disable@edlabel%
9552   \let\linenum@gobble}
9553 \newcommand{\l@drestoreforedtext}{%
9554   \def\do##1{\global\csletcs{##1footnote}{##1@footnote}}
9555   \dolistloop{\@series}%
9556   \let\edindex\xedindex}
9557 %

```

**\l@dnullfills** Nullify and restore some column fillers, etc.

**\l@drestorefills**

```

9558 \newcommand{\l@dnnullfills}{%
9559   \disable@edlabel%
9560   \def\edrowfill##1##2##3{}%
9561 }
9562 \newcommand{\l@drestorefills}{%
9563   \def\edrowfill##1##2##3{\@EDROWFILL@{##1}{##2}{##3}}%
9564 }
9565 %
9566 %

```

**\letsforverteilen** Gathers some lets and other code that is common to the *\*verteilen\** macros.

```

9567 \newcommand{\letsforverteilen}{%
9568   \let\edtext\xedtext
9569   \let\edindex\xedindex
9570   \def\do##1{\global\csletcs{##1footnote}{##1@footnote}}
9571   \dolistloop{\@series}%
9572   \let\linenum\@line@num
9573   \hilfsskip=\l@dcolwidth%
9574   \advance\hilfsskip by -\wd\hilfsbox
9575   \restore@edlabel%
9576 }%
9577 %

```

**\disablel@dtabfeet** Declarations for using or using `\edtext` inside tabulars. The default at this point is for `\edtext`.

```

9578 \newcommand\disablel@dtabfeet{\l@dmodforedtext}%
9579 \newcommand\enablel@dtabfeet{\l@drestoreforedtext}%
9580 %

```

### 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.

```

9581 \newcount\l@dampcount
9582   \l@dampcount=1\relax
9583 \newcount\l@dcolcount
9584   \l@dcolcount=0\relax
9585 %
9586 %

```

**\hilfsbox** Some (temporary) helper items.

```

\hilfsskip
\Hilfsbox
\hilfscount
9587 \newbox\hilfsbox
9588 \newskip\hilfsskip
9589 \newbox\Hilfsbox
9590 \newcount\hilfscount
9591 %
9592 %

```

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).

```

9593 \newdimen\dcoli
9594 \newdimen\dcolii
9595 \newdimen\dcoliii
9596 \newdimen\dcoliv
9597 \newdimen\dcolv
9598 \newdimen\dcolvi
9599 \newdimen\dcolvii
9600 \newdimen\dcolviii
9601 \newdimen\dcolix
9602 \newdimen\dcolx
9603 \newdimen\dcolxi
9604 \newdimen\dcolxii
9605 \newdimen\dcolxiii
9606 \newdimen\dcolxiv
9607 \newdimen\dcolxv
9608 \newdimen\dcolxvi
9609 \newdimen\dcolxvii
9610 \newdimen\dcolxviii
9611 \newdimen\dcolxix
9612 \newdimen\dcolxx
9613 \newdimen\dcolxxi
9614 \newdimen\dcolxxii
9615 \newdimen\dcolxxiii
9616 \newdimen\dcolxxiv
9617 \newdimen\dcolxxv
9618 \newdimen\dcolxxvi
9619 \newdimen\dcolxxvii
9620 \newdimen\dcolxxviii
9621 \newdimen\dcolxxix
9622 \newdimen\dcolxxx
9623 \newdimen\dcolerr % added for error handling
9624
9625 %

```

`\l@dcolwidth` This is a cunning way of storing the columnwidths indexed by the column number `\l@dcolcount`, like an array. (was `\Dimenzuordnung`)

```

9626 \newcommand{\l@dcolwidth}{\ifcase \the\l@dcolcount \dcoli %???
9627 \or \dcoli \or \dcolii \or \dcoliii
9628 \or \dcoliv \or \dcolv \or \dcolvi
9629 \or \dcolvii \or \dcolviii \or \dcolix \or \dcolx
9630 \or \dcolxi \or \dcolxii \or \dcolxiii
9631 \or \dcolxiv \or \dcolxv \or \dcolxvi
9632 \or \dcolxvii \or \dcolxviii \or \dcolxix \or \dcolxx
9633 \or \dcolxxi \or \dcolxxii \or \dcolxxiii
9634 \or \dcolxxiv \or \dcolxxv \or \dcolxxvi
9635 \or \dcolxxvii \or \dcolxxviii \or \dcolxxix \or \dcolxxx
9636 \else \dcolerr \fi}
9637
9638 %

```

`\stepl@dcolcount` This increments the column counter, and issues an error message if it is too large.

```

9639 \newcommand*{\stepl@dcolcount}{\advance\l@dcolcount\@ne

```

```

9640 \ifnum\l@dc@colcount>30\relax
9641 \led@err@TooManyColumns
9642 \fi}
9643
9644 %

```

**\l@setmaxcolwidth** Sets the column width to the maximum value seen so far.

```

9645 \newcommand{\l@setmaxcolwidth}{%
9646 \ifdim\l@dc@colwidth < \wd\hilfsbox
9647 \l@dc@colwidth = \wd\hilfsbox
9648 \else \relax \fi}
9649
9650 %

```

**\measurecell** Measure (recursively) the width required for a math cell.

```

9651 \def\measurecell #1{%
9652 \ifx #1\ \ifnum\l@dc@colcount=0\let\NEXT\relax%
9653 \else\l@dcheckcols%
9654 \l@dc@colcount=0%
9655 \let\NEXT\measurecell%
9656 \fi%
9657 \else\setbox\hilfsbox=\hbox{$\displaystyle{#1}$}%
9658 \step1@dc@colcount%
9659 \l@setmaxcolwidth%
9660 \let\NEXT\measurecell%
9661 \fi\NEXT}
9662
9663 %

```

**\measuretcell** Measure (recursively) the width required for a text cell.

```

9664 \def\measuretcell #1{%
9665 \ifx #1\ \ifnum\l@dc@colcount=0\let\NEXT\relax%
9666 \else\l@dcheckcols%
9667 \l@dc@colcount=0%
9668 \let\NEXT\measuretcell%
9669 \fi%
9670 \else\setbox\hilfsbox=\hbox{#1}%
9671 \step1@dc@colcount%
9672 \l@setmaxcolwidth%
9673 \let\NEXT\measuretcell%
9674 \fi\NEXT}
9675
9676 %

```

**\measuremrow** Measure (recursively) the width required for a math row.

```

9677 \def\measuremrow #1\{%
9678 \ifx #1\ \let\NEXT\relax%
9679 \else\measurecell #1\&\&\&%
9680 \let\NEXT\measuremrow%
9681 \fi\NEXT}
9682 %

```

`\measurerow` Measure (recursively) the width required for a text row.

```

9683 \def\measurerow #1\{\%
9684   \ifx #1&\let\NEXT\relax%
9685   \else\measuretcell #1&\&\&\&%
9686     \let\NEXT\measurerow%
9687   \fi\NEXT}
9688
9689 %

```

`\edtabcolsep` The length `\edtabcolsep` controls the distance between columns.

```

9690 \newskip\edtabcolsep
9691 \global\edtabcolsep=10pt
9692
9693 %

```

```

\variab94 \newcommand{\variab}{\relax}
9695
9696 %

```

`\l@dcheckcols` Check that the number of columns is consistent.

```

9697 \newcommand*{\l@dcheckcols}{%
9698   \ifnum\l@dcolcount=1\relax
9699   \else
9700     \ifnum\l@dampcount=1\relax
9701     \else
9702       \ifnum\l@dcolcount=\l@dampcount\relax
9703       \else
9704         \l@d@err@UnequalColumns
9705       \fi
9706     \fi
9707     \l@dampcount=\l@dcolcount
9708   \fi}
9709
9710 %

```

`\edfilldimen` A length.

```

9711 \newdimen\edfilldimen
9712 \edfilldimen=0pt
9713
9714 %

```

`\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).

```

9715 \newcounter{addcolcount}
9716 \renewcommand{\theaddcolcount}{\romannumeral \c@addcolcount}
9717 %

```

## XXXII.2.3 Tabular typesetting

`\setmcellright` Typeset (recursively) cells of display math right justified.

```

9718 \def\setmcellright #1&{%
9719     \disable@edlabel%
9720     \let\edindex\nulledindex
9721     \ifx #1\ \ifnum\l@dcolcount=0%\removeastskip
9722         \let\Next\relax%
9723     \else\l@dcolcount=0%
9724         \let\Next=\setmcellright%
9725     \fi%
9726 \else%
9727     \disablel@dtabfeet%
9728     \stepl@dcolcount%
9729     \disable@notes%
9730     \setbox\hilfsbox=\hbox{$\displaystyle{#1}$}%
9731     \restore@notes%
9732     \letsforverteilen%
9733     \hskip\hilfsskip$\displaystyle{#1}$%
9734     \hskip\edtabcolsep%
9735     \let\Next=\setmcellright%
9736 \fi\Next}
9737
9738 %

```

`\settcellright` Typeset (recursively) cells of text right justified.

```

9739 \def\settcellright #1&{%
9740     \disable@edlabel%
9741     \let\edindex\nulledindex
9742     \ifx #1\ \ifnum\l@dcolcount=0%\removeastskip
9743         \let\Next\relax%
9744     \else\l@dcolcount=0%
9745         \let\Next=\settcellright%
9746     \fi%
9747 \else%
9748     \disablel@dtabfeet%
9749     \stepl@dcolcount%
9750     \disable@notes%
9751     \setbox\hilfsbox=\hbox{#1}%
9752     \restore@notes%
9753     \letsforverteilen%
9754     \hskip\hilfsskip#1%
9755     \hskip\edtabcolsep%
9756     \let\Next=\settcellright%
9757 \fi\Next}
9758 %

```

`\setmcelleft` Typeset (recursively) cells of display math left justified.

```

9759 \def\setmcelleft #1&{%
9760     \disable@edlabel%
9761     \let\edindex\nulledindex
9762     \ifx #1\ \ifnum\l@dcolcount=0 \let\Next\relax%

```

```

9763         \else\l@dcolcount=0%
9764             \let\Next=\setmcellleft%
9765         \fi%
9766     \else \disablel@dtabfeet%
9767         \step1@dcolcount%
9768         \disable@notes%
9769         \setbox\hilfsbox=\hbox{\displaystyle{#1}}%
9770         \restore@notes%
9771         \letsforverteilen%
9772         $\displaystyle{#1}$\hskip\hilfsskip\hskip\edtabcolsep%
9773         \let\Next=\setmcellleft%
9774     \fi\Next}
9775
9776 %

```

**\settcclleft** Typeset (recursively) cells of text left justified.

```

9777 \def\settcclleft #1&{%
9778     \disable@edlabel%
9779     \let\edindex\nulledindex
9780     \ifx #1\ \ifnum\l@dcolcount=0 \let\Next\relax%
9781         \else\l@dcolcount=0%
9782             \let\Next=\settcclleft%
9783         \fi%
9784     \else \disablel@dtabfeet%
9785         \step1@dcolcount%
9786         \disable@notes%
9787         \setbox\hilfsbox=\hbox{#1}%
9788         \restore@notes%
9789         \letsforverteilen%
9790         #1\hskip\hilfsskip\hskip\edtabcolsep%
9791         \let\Next=\settcclleft%
9792     \fi\Next}
9793 %

```

**\setmcellcenter** Typeset (recursively) cells of display math centered.

```

9794 \def\setmcellcenter #1&{%
9795     \disable@edlabel%
9796     \let\edindex\nulledindex
9797     \ifx #1\ \ifnum\l@dcolcount=0\let\Next\relax%
9798         \else\l@dcolcount=0%
9799             \let\Next=\setmcellcenter%
9800         \fi%
9801     \else \disablel@dtabfeet%
9802         \step1@dcolcount%
9803         \disable@notes%
9804         \setbox\hilfsbox=\hbox{\displaystyle{#1}}%
9805         \restore@notes%
9806         \letsforverteilen%
9807         \hskip 0.5\hilfsskip$\displaystyle{#1}$\hskip0.5\hilfsskip%
9808         \hskip\edtabcolsep%
9809         \let\Next=\setmcellcenter%
9810     \fi\Next}

```



```
9811
9812 %
```

**\settcclcenter** Typeset (recursively) cells of text centered.

```
9813 \def\settcclcenter #1\{%
9814     \disable@edlabel%
9815     \let\edindex\nulledindex
9816     \ifx #1\ \ifnum\l@dcolcount=0 \let\Next\relax%
9817         \else\l@dcolcount=0%
9818             \let\Next=\settcclcenter%
9819         \fi%
9820     \else \disablel@dtabfeet%
9821         \step1@dcolcount%
9822         \disable@notes%
9823         \setbox\hilfsbox=\hbox{#1}%
9824         \restore@notes%
9825         \letsforverteilen%
9826         \hskip 0.5\hilfsskip #1\hskip 0.5\hilfsskip%
9827         \hskip\edtabcolsep%
9828         \let\Next=\settcclcenter%
9829     \fi\Next}
9830
9831 %
```

**\NEXT** \let\NEXT=\relax

```
9833
9834 %
```

**\setmrowright** Typeset (recursively) rows of right justified math.

```
9835 \def\setmrowright #1\{%
9836     \ifx #1& \let\NEXT\relax
9837     \else \centerline{\setmcellright #1&\&\&}
9838         \let\NEXT=\setmrowright
9839     \fi\NEXT}
9840 %
```

**\settroright** Typeset (recursively) rows of right justified text.

```
9841 \def\settroright #1\{%
9842     \ifx #1& \let\NEXT\relax
9843     \else \centerline{\settcclright #1&\&\&}
9844         \let\NEXT=\settroright
9845     \fi\NEXT}
9846
9847 %
```

**\setmrowleft** Typeset (recursively) rows of left justified math.

```
9848 \def\setmrowleft #1\{%
9849     \ifx #1&\let\NEXT\relax
9850     \else \centerline{\setmcellleft #1&\&\&}
9851     \fi\NEXT}
```

```

9851 \let\NEXT=\setmrowleft
9852 \fi\NEXT}
9853 %

```

**\settrorleft** Typeset (recursively) rows of left justified text.

```

9854 \def\settrorleft #1\{ %
9855 \ifx #1& \let\NEXT\relax
9856 \else \centerline{\settrorleft #1&\&\&}
9857 \let\NEXT=\settrorleft
9858 \fi\NEXT}
9859
9860 %

```

**\setmrowcenter** Typeset (recursively) rows of centered math.

```

9861 \def\setmrowcenter #1\{ %
9862 \ifx #1& \let\NEXT\relax %
9863 \else \centerline{\setmrowcenter #1&\&\&}
9864 \let\NEXT=\setmrowcenter
9865 \fi\NEXT}
9866 %

```

**\settrorcenter** Typeset (recursively) rows of centered text.

```

9867 \def\settrorcenter #1\{ %
9868 \ifx #1& \let\NEXT\relax
9869 \else \centerline{\settrorcenter #1&\&\&}
9870 \let\NEXT=\settrorcenter
9871 \fi\NEXT}
9872
9873 %

```

**\nullsetzen** `\newcommand{\nullsetzen}{%`

```

9875 \step1@dcolcount %
9876 \l@dcolwidth=0pt %
9877 \ifnum\l@dcolcount=30\let\NEXT\relax %
9878 \l@dcolcount=0\relax
9879 \else\let\NEXT\nullsetzen %
9880 \fi\NEXT}
9881
9882 %

```

**\edatleft** `\edatleft[ $\langle math \rangle$ ]{ $\langle symbol \rangle$ }{ $\langle len \rangle$ }`. Left  $\langle symbol \rangle$ ,  $2\langle len \rangle$  high with prepended  $\langle math \rangle$  vertically centered.

```

9883 \newcommand{\edatleft}[3][\@empty]{ %
9884 \ifx#1\@empty
9885 \vbox to 10pt{\vss\hbox{\left#2\vrule width0pt height #3
9886 depth 0pt \right. $\hss}\vfil}
9887 \else
9888 \vbox to 4pt{\vss\hbox{\left#2\vrule width0pt height #3
9889 depth 0pt \right. $\}\vfil}
9890 \fi}
9891 %

```

`\edatright` `\edatright[ $\langle symbol \rangle$ ]{ $\langle len \rangle$ }`. Right  $\langle symbol \rangle$ ,  $2\langle len \rangle$  high with appended  $\langle math \rangle$  vertically centered.

```

9892 \newcommand{\edatright}[3][\@empty]{%
9893   \ifx#1\@empty
9894     \vbox to 10pt{\vss\hbox{\$ \left.\vrule width0pt height #3
9895       depth 0pt \right#2 $\hss}\vfil}
9896   \else
9897     \vbox to 4pt{\vss\hbox{\$ \left.\vrule width0pt height #3
9898       depth 0pt \right#2 #1 $\}\vfil}
9899   \fi}
9900
9901 %

```

`\edvertline` `\edvertline{ $\langle len \rangle$ }` vertical line  $\langle len \rangle$  high.

```

9902 \newcommand{\edvertline}[1]{\vbox to 8pt{\vss\hbox{\vrule height #1}\vfil}}
9903
9904 %

```

`\edvertdots` `\edvertdots{ $\langle len \rangle$ }` vertical dotted line  $\langle len \rangle$  high.

```

9905 \newcommand{\edvertdots}[1]{\vbox to 1pt{\vss\vbox to #1%
9906   {\cleaders\hbox{\$ \m@th\hbox{.}}\vbox to 0.5em{ }$\}\vfil}}
9907
9908 %

```

`\l@dtabaddcols` `\l@dtabaddcols{ $\langle startcol \rangle$ }{ $\langle endcol \rangle$ }` adds the widths of the columns  $\langle startcol \rangle$  through  $\langle endcol \rangle$  to `\edfilldimen`. It is a  $\text{\LaTeX}$  style reimplementation of the original `\@add@`.

```

9909 \newcommand{\l@dtabaddcols}[2]{%
9910   \l@dcheckstartend{#1}{#2}%
9911   \ifl@dstartendok
9912     \setcounter{addcolcount}{#1}%
9913     \@whilenum \value{addcolcount}<#2\relax \do
9914       {\advance\edfilldimen by \the \csname dcol\theaddcolcount\endcsname
9915         \advance\edfilldimen by \edtabcolsep
9916         \stepcounter{addcolcount}}%
9917     \advance\edfilldimen by \the \csname dcol\theaddcolcount\endcsname
9918   \fi
9919 }
9920
9921 %

```

`\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.

```

9922 \newif\ifl@dstartendok
9923 \newcommand{\l@dcheckstartend}[2]{%
9924   \l@dstartendoktrue
9925   \ifnum #1<\@ne
9926     \l@dstartendokfalse
9927     \led@err@LowStartColumn

```

```

9928 \fi
9929 \ifnum #2>30\relax
9930 \l@startendokfalse
9931 \led@err@HighEndColumn
9932 \fi
9933 \ifnum #1>#2\relax
9934 \l@startendokfalse
9935 \led@err@ReverseColumns
9936 \fi
9937 }
9938
9939 %

```

**\edrowfill** `\edrowfill{<startcol>}{<endcol>}` fill fills columns `<startcol>` to `<endcol>` inclusive with `<fill>` (e.g. `\hrulefill`, `\upbracefill`). This is a  $\TeX$  style reimplementation and generalization of the original `\waklam`, `\Waklam`, `\waklamec`, `\wastricht` and `\wapunktel` macros.

```

9940 \newcommand*{\edrowfill}[3]{%
9941 \l@dtabaddcols{#1}{#2}%
9942 \hb@xt@ \the\l@dcowidth{\hb@xt@ \the\edfilldimen{#3}\hss}}
9943 \let\@edrowfill=\edrowfill
9944 \def\@EDROWFILL@#1#2#3{\@edrowfill@{#1}{#2}{#3}}
9945
9946 %

```

**\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`.

```

9947 \newcommand{\leftltab}[1]{%
9948 \hb@xt@\z@\vbox{\edtabindent%
9949 \moveleft\Hilfsskip\hbox{\ #1}}\hss}}
9950
9951 %

```

**\leftrtab** `\leftrtab{<text>}{<math>}` for `\edbeforetab` in `\rtab`.

```

9952 \newcommand{\leftrtab}[2]{%
9953 #2\hb@xt@\z@\vbox{\edtabindent%
9954 \advance\Hilfsskip by\dcoli%
9955 \moveleft\Hilfsskip\hbox{\ #1}}\hss}}
9956
9957 %

```

**\leftctab** `\leftctab{<text>}{<math>}` for `\edbeforetab` in `\ctab`.

```

9958 \newcommand{\leftctab}[2]{%
9959 \hb@xt@\z@\vbox{\edtabindent\l@dcowidth=\l@dampcount%
9960 \advance\Hilfsskip by 0.5\dcoli%
9961 \setbox\hilfsbox=\hbox{%

```

```

9962     \disable@edlabel%
9963     \disablel@dtabfeet$\displaystyle{#2}$}%
9964     \advance\Hilfsskip by -0.5\wd\hilfsbox%
9965     \moveleft\Hilfsskip\hbox{\ #1}\hss}%
9966     #2}
9967
9968 %

```

`\rightctab` `\rightctab{<math><math>}{<text>}` for `\edaftertab` in `\ctab`.

```

9969 \newcommand{\rightctab}[2]{%
9970     \setbox\hilfsbox=\hbox{%
9971     \disable@edlabel%
9972     \disablel@dtabfeet#2}\l@dampcount=\l@dcolcount%
9973     #1\hb@xt@z@{\vbox{\edtabindent\l@dcolcount=\l@dampcount%
9974     \advance\Hilfsskip by 0.5\l@dcolwidth%
9975     \advance\Hilfsskip by -\wd\hilfsbox%
9976     \setbox\hilfsbox=\hbox{%
9977     \disable@edlabel%
9978     \disablel@dtabfeet$\displaystyle{#1}$}%
9979     \advance\Hilfsskip by -0.5\wd\hilfsbox%
9980     \advance\Hilfsskip by \edtabcolsep%
9981     \moveright\Hilfsskip\hbox{ #2}\hss}%
9982     }
9983
9984 %

```

`\rightltab` `\rightltab{<math><math>}{<text>}` for `\edaftertab` in `\ltab`.

```

9985 \newcommand{\rightltab}[2]{%
9986     \setbox\hilfsbox=\hbox{%
9987     \disable@edlabel%
9988     \disablel@dtabfeet#2}\l@dampcount=\l@dcolcount%
9989     #1\hb@xt@z@{\vbox{\edtabindent\l@dcolcount=\l@dampcount%
9990     \advance\Hilfsskip by\l@dcolwidth%
9991     \advance\Hilfsskip by-\wd\hilfsbox%
9992     \setbox\hilfsbox=\hbox{%
9993     \disable@edlabel%
9994     \disablel@dtabfeet$\displaystyle{#1}$}%
9995     \advance\Hilfsskip by-\wd\hilfsbox%
9996     \advance\Hilfsskip by\edtabcolsep%
9997     \moveright\Hilfsskip\hbox{ #2}\hss}%
9998     }
9999
10000 %

```

`\rightrtab` `\rightrtab{<math><math>}{<text>}` for `\edaftertab` in `\rtab`.

```

10001 \newcommand{\rightrtab}[2]{%
10002     \setbox\hilfsbox=\hbox{%
10003     \disable@edlabel%
10004     \disablel@dtabfeet#2}%
10005     #1\hb@xt@z@{\vbox{\edtabindent%
10006     \advance\Hilfsskip by-\wd\hilfsbox%

```

```

10007     \advance\Hilfsskip by\edtabcolsep%
10008     \moveright\Hilfsskip\hbox{ #2}\hss}%
10009     }
10010
10011 %

```

**\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 second pass to typeset the body.

**\edaftertab**

```

10012 \newcommand{\rtab}[1]{%
10013   \l@dnnullfills
10014   \def\edbeforetab##1##2{\lefttrtab{##1}{##2}}%
10015   \def\edaftertab##1##2{\righttrtab{##1}{##2}}%
10016   \measurebody{#1}%
10017   \l@drestorefills
10018   \variab
10019   \setmrowright #1\&\&%
10020   \enablel@dtabfeet}
10021
10022 %

```

**\measurebody** `\measurebody{<body>}` measures the array `<body>`.

```

10023 \newcommand{\measurebody}[1]{%
10024   \disablel@dtabfeet%
10025   \l@dcolcount=0%
10026   \nullsetzen%
10027   \l@dcolcount=0
10028   \measuremrow #1\&\&%
10029   \global\l@dampcount=1}
10030
10031 %

```

**\rtabtext** `\rtabtext{<body>}` typesets `<body>` as a tabular with the entries right justified.

```

10032 \newcommand{\rtabtext}[1]{%
10033   \l@dnnullfills
10034   \measuretbody{#1}%
10035   \l@drestorefills
10036   \variab
10037   \settroright #1\&\&%
10038   \enablel@dtabfeet}
10039
10040 %

```

**\measuretbody** `\measuretbody{<body>}` measures the tabular `<body>`.

```

10041 \newcommand{\measuretbody}[1]{%
10042   \disable@notes%
10043   \disablel@dtabfeet%
10044   \l@dcolcount=0%
10045   \nullsetzen%
10046   \l@dcolcount=0

```

```

10047 \measuretrrow #1\\&\\%
10048 \restore@notes%
10049 \global\l@dampcount=1}
10050
10051 %

```

**\ltab** Array with entries left justified.

```

\edbeforetab
\edaftertab
10052 \newcommand{\ltab}[1]{%
10053 \l@dnnullfills
10054 \def\edbeforetab##1##2{\leftltab{##1}{##2}}%
10055 \def\edaftertab##1##2{\rightltab{##1}{##2}}%
10056 \measuretbody{#1}%
10057 \l@drestorefills
10058 \variab
10059 \setmrowleft #1\\&\\%
10060 \enablel@dtabfeet}
10061
10062 %

```

**\ltabtext** Tabular with entries left justified.

```

10063 \newcommand{\ltabtext}[1]{%
10064 \l@dnnullfills
10065 \measuretbody{#1}%
10066 \l@drestorefills
10067 \variab
10068 \settrrowleft #1\\&\\%
10069 \enablel@dtabfeet}
10070
10071 %

```

**\ctab** Array with centered entries.

```

\edbeforetab
\edaftertab
10072 \newcommand{\ctab}[1]{%
10073 \l@dnnullfills
10074 \def\edbeforetab##1##2{\leftctab{##1}{##2}}%
10075 \def\edaftertab##1##2{\rightctab{##1}{##2}}%
10076 \measuretbody{#1}%
10077 \l@drestorefills
10078 \variab
10079 \setmrowcenter #1\\&\\%
10080 \enablel@dtabfeet}
10081
10082 %

```

**\ctabtext** Tabular with entries centered.

```

10083 \newcommand{\ctabtext}[1]{%
10084 \l@dnnullfills
10085 \measuretbody{#1}%
10086 \l@drestorefills
10087 \variab
10088 \settrrowcenter #1\\&\\%

```

```

10089 \enablel@dtabfeet}
10090
10091 %

```

```

\spreadtext92 \newcommand{\spreadtext}[1]{%\l@dcolcount=\l@dampcount%
10093 \hb@xt@ \the\l@dcolwidth{\hbox{#1}\hss}}
10094 %

```

```

\spreadmath95 \newcommand{\spreadmath}[1]{%
10096 \hb@xt@ \the\l@dcolwidth{\hbox{$\displaystyle{#1}$}\hss}}
10097
10098 %

```

**\HILFSskip** More helpers.

```

\Hilfsskip
10099 \newskip\HILFSskip
10100 \newskip\Hilfsskip
10101
10102 %

```

```

\EDTABINDENT03 \newcommand{\EDTABINDENT}{%
10104 \ifnum\l@dcolcount=30\let\NEXT\relax\l@dcolcount=0%
10105 \else\step\l@dcolcount%
10106 \advance\Hilfsskip by\l@dcolwidth%
10107 \ifdim\l@dcolwidth=0pt\advance\hilfscout\@ne
10108 \else\advance\Hilfsskip by \the\hilfscout\edtabcolsep%
10109 \hilfscout=1\fi%
10110 \let\NEXT=\EDTABINDENT%
10111 \fi\NEXT}%
10112 %

```

**\edtabindent** (was `\tabindent`)

```

10113 \newcommand{\edtabindent}{%
10114 \l@dcolcount=0\relax
10115 \Hilfsskip=0pt%
10116 \hilfscout=1\relax
10117 \EDTABINDENT%
10118 \hilfsskip=\hsize%
10119 \advance\hilfsskip -\Hilfsskip%
10120 \Hilfsskip=0.5\hilfsskip%
10121 }%
10122
10123 %

```

**\EDTAB** (was `\TAB`)

```

10124 \def\EDTAB #1|#2|{%
10125 \setbox\tabhilfbox=\hbox{$\displaystyle{#1}$}%
10126 \setbox\tabHilfbox=\hbox{$\displaystyle{#2}$}%
10127 \advance\tabelskip -\wd\tabhilfbox%

```



```

10128 \advance\tabelskip -\wd\tabHilfbox%
10129 \unhbox\tabhilfbox\hskip\tabelskip%
10130 \unhbox\tabHilfbox}%
10131
10132 %

```

`\EDTABtext` (was `\TABtext`)

```

10133 \def\EDTABtext #1|#2|{%
10134 \setbox\tabhilfbox=\hbox{#1}%
10135 \setbox\tabHilfbox=\hbox{#2}%
10136 \advance\tabelskip -\wd\tabhilfbox%
10137 \advance\tabelskip -\wd\tabHilfbox%
10138 \unhbox\tabhilfbox\hskip\tabelskip%
10139 \unhbox\tabHilfbox}%
10140 %

```

`\tabhilfbox` Further helpers.

```

10141 \newbox\tabhilfbox
10142 \newbox\tabHilfbox
10143
10144 %

```

### XXXII.2.4 Environments

`edarrayl edarrayc edarrayr` The environment forms for `\ltab`, `\ctab` and `\rtab`.

```

10145 \newenvironment{edarrayl}{\l@dcollect@body\ltab}{}
10146 \newenvironment{edarrayc}{\l@dcollect@body\ctab}{}
10147 \newenvironment{edarrayr}{\l@dcollect@body\rtab}{}
10148
10149 %

```

`edtabularl edtabularc edtabularr` The environment forms for `\ltabtext`, `\ctabtext` and `\rtabtext`.

```

10150 \newenvironment{edtabularl}{\l@dcollect@body\ltabtext}{}
10151 \newenvironment{edtabularc}{\l@dcollect@body\ctabtext}{}
10152 \newenvironment{edtabularr}{\l@dcollect@body\rtabtext}{}
10153
10154 %

```

## 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
\quote10155 \newcommand{\initnumbering@quote}{
\quote10156 \ifnoquotation@else
\endquote10157 \renewcommand{\quotation}{\par\leavevmode%

```

```

10158         \parindent=1.5em%
10159         \skipnumbering%
10160         \ifautopar%
10161             \vskip-\parskip%
10162         \else%
10163             \vskip\topsep%
10164         \fi%
10165         \global\leftskip=\leftmargin%
10166         \global\rightskip=\leftmargin%
10167     }
10168 \renewcommand{\endquotation}{\par%
10169     \global\leftskip=0pt%
10170     \global\rightskip=0pt%
10171     \leavevmode%
10172     \skipnumbering%
10173     \ifautopar%
10174         \vskip-\parskip%
10175     \else%
10176         \vskip\topsep%
10177     \fi%
10178 }
10179 \renewcommand{\quote}{\par\leavevmode%
10180     \parindent=0pt%
10181     \skipnumbering%
10182     \ifautopar%
10183         \vskip-\parskip%
10184     \else%
10185         \vskip\topsep%
10186     \fi%
10187     \global\leftskip=\leftmargin%
10188     \global\rightskip=\leftmargin%
10189 }
10190 \renewcommand{\endquote}{\par%
10191     \global\leftskip=0pt%
10192     \global\rightskip=0pt%
10193     \leavevmode%
10194     \skipnumbering%
10195     \ifautopar%
10196         \vskip-\parskip%
10197     \else%
10198         \vskip\topsep%
10199     \fi%
10200 }
10201 \fi
10202 }
10203 %

```

## XXXIV Section's title commands

### XXXIV.1 Commands to disable some feature

`\ledsectnotoc` The `\ledsectnotoc` only disables the `\addcontentsline` macro.

```
10204 \newcommand{\ledsectnotoc}{\let\addcontentsline@gobblethree}
10205 %
```

`\ledsectnomark` The `\ledsectnomark` only disables the `\chaptermark`, `\sectionmark` and `\subsectionmark` macros.

```
10206 \newcommand{\ledsectnomark}{%
10207   \let\chaptermark@gobble%
10208   \let\sectionmark@gobble%
10209   \let\subsectionmark@gobble%
10210 }
10211 %
```

## 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).

```
10212 \catcode`\#=12
10213 \notbool{@noeled@sec}{%
10214 %
```

`\beforeeledchapter` For technical reasons, not yet solved, page-breaking before chapters can't be made automatically by `eledmac`. Users have to use `\beforeeledchapter`.

```

10215 \ifl@dmemoir
10216 \newcommand\beforeeledchapter{%
10217 \clearforchapter%
10218 }
10219 \else
10220 \newcommand\beforeeledchapter{%
10221 \if@openright%
10222 \cleardoublepage%
10223 \else%
10224 \clearpage%
10225 \fi%
10226 }
10227 \fi
10228 %

```

#### 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.

```

10229 \def\print@rightmargin@eledsection{%
10230 \if@eled@sectioning%
10231 \begingroup%
10232 \if@RTL%
10233 \let\llap\rlap%
10234 \let\leftlinenum\rightlinenum%
10235 \let\leftlinenumR\rightlinenumR%
10236 \let\l@drd@ta\l@dld@ta%
10237 \let\l@drrsn@te\l@dlsn@te%
10238 \fi%
10239 \hfill\l@drd@ta \csuse{LR}{\l@drrsn@te}%
10240 \endgroup%
10241 \fi%
10242 }%
10243
10244 \def\print@leftmargin@eledsection{%
10245 \if@eled@sectioning%
10246 \leavevmode%
10247 \begingroup%
10248 \if@RTL%
10249 \let\rlap\llap%
10250 \let\rightlinenum\leftlinenum%
10251 \let\rightlinenumR\leftlinenumR%
10252 \let\l@dld@ta\l@drd@ta%
10253 \let\l@dlsn@te\l@drrsn@te%
10254 \fi%
10255 \l@dld@ta\csuse{LR}{\l@dlsn@te}%
10256 \endgroup%
10257 \fi%
10258 }%
10259
10260 %

```

## XXXIV.5 Patching standard commands

`\M@sect` 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.

```

10261 \AtBeginDocument{%
10262
10263
10264 \pretocmd{\M@sect}
10265   {\let\old@edtext=\edtext%
10266    \let\edtext=\dummy@edtext@showlemma%
10267   }
10268   {}
10269   {}
10270
10271 \apptocmd{\M@sect}
10272   {\let\edtext=\old@edtext}
10273   {}
10274   {}
10275
10276 \patchcmd{\M@sect}
10277   { #9}
10278   { #9%
10279    \print@rightmargin@eledsection%
10280   }
10281   {}
10282   {}
10283
10284 \patchcmd{\M@sect}
10285   {\hskip #3\relax}
10286   {\hskip #3\relax%
10287    \print@leftmargin@eledsection%
10288   }
10289   {}
10290   {}
10291
10292 \patchcmd{\@mem@old@ssect}
10293   {#5}
10294   {#5%
10295    \print@leftmargin@eledsection%
10296   }
10297   {}
10298   {}
10299
10300 \patchcmd{\@mem@old@ssect}
10301   {\hskip #1}
10302   {\hskip #1%
10303    \print@rightmargin@eledsection%
```

```

10304 }
10305 {}
10306 {}
10307
10308
10309
10310 \patchcmd{\scr@startchapter}{\if@openright\cleardoublepage\else\clearpage\
fi}{%
10311 \if@eled@sectioning\else%
10312 \ifl@dprintingpages\else%
10313 \if@openright\cleardoublepage\else\clearpage\fi}No clearpage inside a
\Pages: will keep critical notes from printing on the title page. Here for
scrbook.
10314 \fi%
10315 \fi%
10316 }
10317 {}
10318 {}
10319
10320 \patchcmd{\@makechapterhead}
10321 {#1}
10322 {\print@leftmargin@eledsection%
10323 #1%
10324 \print@rightmargin@eledsection%
10325 }
10326 {}
10327 {}
10328
10329 \patchcmd{\@makechapterhead}% For BIDI
10330 {\if@RTL\raggedleft\else\raggedright\fi}%
10331 {\if@eled@sectioning\else%
10332 \if@RTL\raggedleft\else\raggedright\fi%
10333 \fi%
10334 }%
10335 {}%
10336 {}%
10337
10338 \patchcmd{\@makeschapterhead}
10339 {#1}
10340 {\print@leftmargin@eledsection%
10341 #1%
10342 \print@rightmargin@eledsection%
10343 }
10344 {}
10345 {}
10346
10347 \pretocmd{\@sect}
10348 {\let\old@edtext=\edtext
10349 \let\edtext=\dummy@edtext@showlemma%
10350 }
10351 {}
10352 {}
10353
10354 \apptocmd{\@sect}

```

```

10355 {\let\edtext=\old@edtext}
10356 {}
10357 {}
10358
10359 \pretocmd{\@ssect}
10360 {\let\old@edtext=\edtext%
10361 \let\edtext=\dummy@edtext@showlemma%
10362 }
10363 {}
10364 {}
10365
10366 \apptocmd{\@ssect}
10367 {\let\edtext=\old@edtext}
10368 {}
10369 {}
10370
10371 %

```

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.

```

10372 \ifdef{\H@old@sect}{%
10373
10374 \patchcmd{\H@old@sect}%
10375 {#8}
10376 {#8%
10377 \print@rightmargin@eledsection%
10378 }
10379 {}
10380 {}
10381
10382 \patchcmd{\H@old@sect}%
10383 {\hskip #3\relax}
10384 {\hskip #3\relax%
10385 \print@leftmargin@eledsection%
10386 }
10387 {}
10388 {}
10389
10390 \patchcmd{\H@old@@sect}%
10391 {#5}
10392 {#5%
10393 \print@rightmargin@eledsection%
10394 }
10395 {}
10396 {}
10397
10398 \patchcmd{\H@old@@sect}%
10399 {\hskip #1}
10400 {\hskip #1%
10401 \print@leftmargin@eledsection%
10402 }
10403 {}
10404 {}

```

```

10405 }%
10406 {
10407 \patchcmd{\@sect}
10408   {#8}
10409   {#8%
10410   \print@rightmargin@eledsection%
10411   }
10412   {}
10413   {}
10414
10415 \patchcmd{\@sect}
10416   {\hskip #3\relax}
10417   {\hskip #3\relax%
10418   \print@leftmargin@eledsection%
10419   }
10420   {}
10421   {}
10422
10423 \patchcmd{\@ssect}
10424   {#5}
10425   {#5%
10426   \print@rightmargin@eledsection%
10427   }
10428   {}
10429   {}
10430
10431 \patchcmd{\@ssect}
10432   {\hskip #1}
10433   {\hskip #1%
10434   \print@leftmargin@eledsection%
10435   }
10436   {}
10437   {}
10438 }%
10439 }%
10440 %

```

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 `#`.

```

10441 {}}%
10442 \protect\catcode`\#=6 %Space NEEDS by \catcode
10443 %

```

**\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.

```

10444 \AtBeginDocument{%
10445 \patchcmd{\chapter}{\clearforchapter}{%
10446   \if@eled@sectioning\else%
10447   \ifl@dprintingpages\else%
10448   \clearforchapter%
10449   \fi%
10450 \fi%

```



```

10451 }%
10452 {}%
10453 {}%
10454
10455 \patchcmd{\chapter}{\if@openright\cleardoublepage\else\clearpage\fi}{%
10456 \if@eled@sectioning\else%
10457 \ifl@printingpages%
10458 \endgraf%
10459 \else%
10460 \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
10461 \fi%
10462 \fi%
10463 }%
10464 {}%
10465 {}%
10466 }%
10467 %

```

**\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.

```

10468 \newif\if@eled@sectioning%
10469 %

```

We reopen a new `\notbool{@noeled@sec}` statement, as we will define the `\elected` commands.

```

10470 \notbool{@noeled@sec}{%
10471 %

```

## 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.

```

10472 \newwrite\eled@sectioning@out
10473 %

```

**\eledchapter** **\eledsection** And now, the user sectioning commands, which write to the file, and also add content as a “normal” line.

```

\eledsubsection
10474 \newcommand{\eledchapter}[2] [] {%
\eledsubsubsection
10475 \disable@familiarnotes%
\eledchapter*
10476 #2%
\eledsection*
10477 \restore@familiarnotes%
\eledsubsection*
10478 \ifledRcol%
\eledsubsubsection*
10479 \immediate\write\eled@sectioningR@out{%
10480 \string\eledchapter{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR}{-}{R}
10481 }%
10482 \else%
10483 \immediate\write\eled@sectioning@out{%

```

```

10484     \string\eled@chapter{#1}{\unexpanded{#2}}{\the\pstarts@read@L}{-}{-}%
10485     }%
10486     \fi%
10487 }
10488
10489 \newcommand{\eledsection}[2] [] {%
10490     \disable@familiarnotes%
10491     #2%
10492     \restore@familiarnotes%
10493     \ifledRcol%
10494         \immediate\write\eled@sectioningR@out{%
10495             \string\eled@section{#1}{\unexpanded{#2}}{\the\l@dumpstartsR}{-}{R}
10496         }%
10497     \else%
10498         \immediate\write\eled@sectioning@out{%
10499             \string\eled@section{#1}{\unexpanded{#2}}{\the\pstarts@read@L}{-}{-}%
10500         }%
10501     \fi%
10502 }
10503
10504 \newcommand{\eledsubsection}[2] [] {%
10505     \disable@familiarnotes%
10506     #2%
10507     \restore@familiarnotes%
10508     \ifledRcol%
10509         \immediate\write\eled@sectioningR@out{%
10510             \string\eled@subsection{#1}{\unexpanded{#2}}{\the\l@dumpstartsR}{-}{R}
10511         }%
10512     \else%
10513         \immediate\write\eled@sectioning@out{%
10514             \string\eled@subsection{#1}{\unexpanded{#2}}{\the\pstarts@read@L}{-}{-}
10515         }%
10516     \fi%
10517 }
10518 \newcommand{\eledsubsubsection}[2] [] {%
10519     \disable@familiarnotes%
10520     #2%
10521     \restore@familiarnotes%
10522     \ifledRcol%
10523         \immediate\write\eled@sectioningR@out{%
10524             \string\eled@subsubsection{#1}{\unexpanded{#2}}{\the\l@dumpstartsR}
10525             {-}{R}
10526         }%
10527     \else%
10528         \immediate\write\eled@sectioning@out{%
10529             \string\eled@subsubsection{#1}{\unexpanded{#2}}{\the\pstarts@read@L}
10530             {-}{-}%
10531         }%
10532     \fi%
10533 }

```

```

10534 \WithSuffix\newcommand\eledchapter*[2] [] {%
10535   \disable@familiarnotes%
10536   #2%
10537   \restore@familiarnotes%
10538   \ifledRcol%
10539     \immediate\write\eled@sectioningR@out{%
10540       \string\eled@chapter{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR}{*}{R}
10541     }%
10542   \else%
10543     \immediate\write\eled@sectioning@out{%
10544       \string\eled@chapter{#1}{\unexpanded{#2}}{\the\pstarts@read@L}{*}{}%
10545     }%
10546   \fi%
10547 }
10548
10549 \WithSuffix\newcommand\eledsection*[2] [] {%
10550   \disable@familiarnotes%
10551   #2%
10552   \restore@familiarnotes%
10553   \ifledRcol%
10554     \immediate\write\eled@sectioningR@out{%
10555       \string\eled@section{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR}{*}{R}
10556     }%
10557   \else%
10558     \immediate\write\eled@sectioning@out{%
10559       \string\eled@section{#1}{\unexpanded{#2}}{\the\pstarts@read@L}{*}{}%
10560     }%
10561   \fi%
10562 }
10563
10564 \WithSuffix\newcommand\eledsubsection*[2] [] {%
10565   \disable@familiarnotes%
10566   #2%
10567   \restore@familiarnotes%
10568   \ifledRcol%
10569     \immediate\write\eled@sectioningR@out{%
10570       \string\eled@subsection{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR}{*}{R}
10571     }%
10572   \else%
10573     \immediate\write\eled@sectioning@out{%
10574       \string\eled@subsection{#1}{\unexpanded{#2}}{\the\pstarts@read@L}
10575     }{*}{}%
10576   \fi%
10577 }
10578
10579 \WithSuffix\newcommand\eledsubsubsection*[2] [] {%
10580   \disable@familiarnotes%
10581   #2%
10582   \restore@familiarnotes%
10583   \ifledRcol%
10584     \immediate\write\eled@sectioningR@out{%
10585       \string\eled@subsubsection{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR

```

```

}[*]{R}
10586   }%
10587   \else%
10588     \immediate\write\eled@sectioning@out{%
10589     \string\eled@subsubsection{#1}{\unexpanded{#2}}{\the\pstarts@read@L
}[*]{}%
10590   }%
10591   \fi%
10592 }
10593 %

```

### 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.

```

10594 \def\eled@chapter#1#2#3#4#5{%
10595   \ifstrempy{#4}%
10596   {%
10597     \ifstrempy{#1}%
10598     {%
10599       \csgdef{eled@sectioning@#3#5}{\let\edtext=\dummy@edtext@showlemma\
chapter{#2}}%
10600       \csgdef{eled@ssectmark@#3#5}{\let\edtext=\dummy@edtext{}\chaptermark
{#2}}%
10601       }%Need for \pairs, because of using parbox.
10602       {%
10603         \csgdef{eled@sectioning@#3#5}{\let\edtext=\dummy@edtext@showlemma\
chapter[#1]{#2}}%
10604         \csgdef{eled@ssectmark@#3#5}{\let\edtext=\dummy@edtext{}\chaptermark
{#2}}%Need for \pairs, because of using parbox.
10605         }%
10606         }%
10607         {%
10608         \ifstrempy{#1}%
10609         {\csgdef{eled@sectioning@#3#5}{\let\edtext=\dummy@edtext@showlemma\
chapter*{#2}}%
10610         {\csgdef{eled@sectioning@#3#5}{\let\edtext=\dummy@edtext@showlemma\
chapter*{#1}{#2}}}%Bug in LaTeX!
10611         }%
10612         \listcsgadd{eled@sections#5@}{#3}%
10613         }
10614 \def\eled@section#1#2#3#4#5{%
10615   \ifstrempy{#4}%
10616   {\ifstrempy{#1}%

```

```

10617     {%
10618     \csgdef{eled@sectioning@#3#5}{\section{#2}}%
10619     \csgdef{eled@sectmark@#3#5}{\let\edtext=\dummy@edtext}\sectionmark
10620     {#2}}%Need for \pairs, because of using parbox.
10621     }%
10622     {%
10623     \csgdef{eled@sectioning@#3#5}{\section[#1]{#2}}%
10624     \csgdef{eled@sectmark@#3#5}{\let\edtext=\dummy@edtext}\sectionmark
10625     {#1}}%Need for \pairs, because of using parbox.
10626     }%
10627     {\ifstrempy{#1}%
10628     {\csgdef{eled@sectioning@#3#5}{\section*{#2}}}%
10629     {\csgdef{eled@sectioning@#3#5}{\section*{#1}{#2}}}%Bug in LaTeX!
10630     }
10631     \listcsadd{eled@sections#5@@}{#3}%
10632     }
10633     \def\eled@subsection#1#2#3#4#5{%
10634     \ifstrempy{#4}%
10635     {\ifstrempy{#1}%
10636     {%
10637     \csgdef{eled@sectioning@#3#5}{\subsection{#2}}%
10638     \csgdef{eled@sectmark@#3#5}{\let\edtext=\dummy@edtext}\csuse{
10639     subsectionmark}{#2}}%Need for \pairs, because of using parbox. \csuse in
10640     case of \subsectionmark is not defined (book)
10641     }%
10642     {%
10643     \csgdef{eled@sectioning@#3#5}{\subsection[#1]{#2}}%
10644     \csgdef{eled@sectmark@#3#5}{\let\edtext=\dummy@edtext}\csuse{
10645     subsectionmark}{#1}}%Need for \pairs, because of using parbox. \csuse in
10646     case of \subsectionmark is not defined (book)
10647     }%
10648     {\ifstrempy{#1}%
10649     {\csgdef{eled@sectioning@#3#5}{\subsection*{#2}}}%
10650     {\csgdef{eled@sectioning@#3#5}{\subsection*{#1}{#2}}}%Bug in LaTeX!
10651     }
10652     \listcsadd{eled@sections#5@@}{#3}%
10653     }
10654     \def\eled@subsubsection#1#2#3#4#5{%
10655     \ifstrempy{#4}%
10656     {\ifstrempy{#1}%
10657     {\csgdef{eled@sectioning@#3#5}{\subsubsection{#2}}}%
10658     {\csgdef{eled@sectioning@#3#5}{\subsubsection[#1]{#2}}}%Bug in
10659     LaTeX!
10660     }
10661     \listcsadd{eled@sections#5@@}{#3}%
10662     }
10663     %

```

End of the conditional test about noeledsec option.

```
10664 }{}
10665 %
```

## 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.

```
10666 \def\normal@page@break{}
10667 %
```

`\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.

```
10668 \def\l@prev@pb{}
10669 \def\l@prev@nopb{}
10670 %
```

`\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.

```
10671 \newcommand{\ledpb}{\write\linenum@out{\string\led@pb}}
10672 \newcommand{\ledpbnum}[1]{\write\linenum@out{\string\led@pbnum{#1}}}
10673 \newcommand{\lednopb}{\write\linenum@out{\string\led@nopb}}
10674 \newcommand{\lednopbnum}[1]{\write\linenum@out{\string\led@nopbnum{#1}}}
10675 %
```

`\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.

```
\led@nopbnum
10676 \newcommand{\led@pb}{\listxadd{\l@prev@pb}{\the\absline@num}}
10677 \newcommand{\led@pbnum}[1]{\listxadd{\l@prev@pb}{#1}}
10678 \newcommand{\led@nopb}{\listxadd{\l@prev@nopb}{\the\absline@num}}
10679 \newcommand{\led@nopbnum}[1]{\listxadd{\l@prev@nopb}{#1}}
10680 %
```

`\ledpbsetting` The `\ledpbsetting` macro only changes the value of `\led@pb@macro`, for which the default value is before.

`\led@pb@setting`

```

10681 \def\led@pb@setting{before}
10682 \newcommand{\ledpbsetting}[1]{\gdef\led@pb@setting{#1}}
10683 %

```

`\led@check@pb` and `\led@check@nopb` The `\led@check@pb` and `\led@check@nopb` are called before or after each line. They check if a page break must occur, depending on the current line and on the content of `\l@pb`.

```

10684 \newcommand{\led@check@pb}{\xifinlist{the\absline@num}{\l@prev@pb}{\
pagebreak[4]}}
10685 \newcommand{\led@check@nopb}{%
10686   \IfStrEq{\led@pb@setting}{before}{%
10687     \xifinlist{the\absline@num}{\l@prev@nopb}{%
10688       {\numdef{\abs@prevline}{the\absline@num-1}%
10689       \xifinlist{\abs@prevline}{\normal@page@break}%
10690       {\nopagebreak[4]\enlargethispage{\baselineskip}}%
10691       }%
10692     }%
10693   }%
10694 }%
10695 \IfStrEq{\led@pb@setting}{after}{%
10696   \xifinlist{the\absline@num}{\l@prev@nopb}{%
10697   \xifinlist{the\absline@num}{\normal@page@break}%
10698     {\nopagebreak[4]\enlargethispage{\baselineskip}}%
10699   }%
10700 }%
10701 }%
10702 }%
10703 }%
10704 }
10705 %

```

## 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.

```

10706 \newcommand{\check@pb@in@verse}{%
10707   \ifinstanza\iflednopbinverse\ifinserthangingsymbol% Using stanzas and
enabling page breaks in verse control, while on a hanging verse.
10708   \ifnum\page@num=\last@page@num\else%If we have change page
10709   \IfStrEq{\led@pb@setting}{before}{%

```

```

10710         \numgdef{\abs@line@verse}{\the\absline@num-1}%
10711         \ledpbnum{\abs@line@verse}%
10712     }{}%
10713     \IfStrEq{\led@pb@setting}{after}{%
10714         \numgdef{\abs@line@verse}{\the\absline@num-1}%
10715         \lednopbnum{\abs@line@verse}%
10716     }{}%
10717     \fi%
10718     \fi\fi\fi%
10719 }
10720 %

```

## 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>

```

10721 \def\Hy@raisedlink@left#1{%
10722     \ifvmode
10723         #1%
10724     \else
10725         \Hy@SaveSpaceFactor
10726         \llap{\smash{%
10727             \begingroup
10728                 \let\HyperRaiseLinkLength\@tempdima
10729                 \setlength\HyperRaiseLinkLength\HyperRaiseLinkDefault
10730                 \HyperRaiseLinkHook
10731             \expandafter\endgroup
10732             \expandafter\raise\the\HyperRaiseLinkLength\hbox{%
10733                 \Hy@RestoreSpaceFactor
10734                 #1%
10735                 \Hy@SaveSpaceFactor
10736             }}%
10737         }}%
10738         \Hy@RestoreSpaceFactor
10739         \penalty\@M\hskip\z@\relax
10740     \fi
10741 }
10742 %

```

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>.



```

10743
10744 \apptocmd{\setcounter}{%
10745   \ifboolexpr{%
10746     test {\ifnumequal{#2}{0}}%
10747     and%
10748     test {\ifcsdef{#1@series}}%
10749   }%
10750   {%
10751     \global\advance\expandafter\csname #1@series\endcsname by 1%
10752   }%
10753   {}}%
10754 }%
10755   {}%
10756   {\led@error@fail@patch@setcounter}%
10757 \apptocmd{\@stpelt}{%
10758   \ifcsdef{#1@series}{%
10759     \global\advance\expandafter\csname #1@series\endcsname by 1%
10760   }%
10761   {}}%
10762   }%
10763   {}}%
10764   {\led@error@fail@patch@stpelt}%
10765 %

```

## XXXVIII Compatibility with eledmac

Here, we define some commands for the eledmac-compat option.

```

10766 \ifeledmaccompat@%
10767
10768   \newcommand{\footnormalX}[1]{\arrangementX[#1]{normal}}%
10769   \newcommand{\footparagraphX}[1]{\arrangementX[#1]{paragraph}}%
10770   \newcommand{\foottwocolX}[1]{\arrangementX[#1]{twocol}}%
10771   \newcommand{\footthreecolX}[1]{\XarrangementX[#1]{threecol}}%
10772
10773   \unless\ifnocritical@
10774     \newcommand{\footnormal}[1]{\Xarrangement[#1]{normal}}%
10775     \newcommand{\footparagraph}[1]{\Xarrangement[#1]{paragraph}}%
10776     \newcommand{\foottwocol}[1]{\Xarrangement[#1]{twocol}}%
10777     \newcommand{\footthreecol}[1]{\Xarrangement[#1]{threecol}}%
10778     \let\hsizetwocol\Xhsizetwocol
10779     \let\hsizethreecol\Xhsizethreecol
10780     \let\bhookXnote\Xbhooknote
10781     \let\boxsymlinenum\Xboxsymlinenum
10782     \let\symlinenum\Xsymlinenum
10783     \let\beforenumberinfootnote\Xbeforenumber
10784     \let\afternumberinfootnote\Xafternumber
10785     \let\beforeXsymlinenum\XbeforeXsymlinenum
10786     \let\afterXsymlinenum\XafterXsymlinenum
10787     \let\inplaceofnumber\Xinplaceofnumber
10788     \let\Xlemmaseparator\lemmaseparator
10789     \let\afterlemmaseparator\Xafterlemmaseparator
10790     \let\beforelemmaseparator\Xbeforelemmaseparator

```

```

10791 \let\inplaceoflemmaseparator\Xinplaceoflemmaseparator
10792 \let\txbeforeXnotes\Xtxbeforenotes
10793 \let\afterXrule\Xafterrule
10794 \let\numberonlyfirstinline\Xnumberonlyfirstinline
10795 \let\numberonlyfirstintwoline\Xnumberonlyfirstintwoline
10796 \let\nonumberinfootnote\Xnonumberinfootnote
10797 \let\pstartinfootnote\Xpstart
10798 \let\pstartinfootnoteeverytime\Xpstarteverytime
10799 \let\onlyXpstart\Xonlypstart
10800 \let\Xnonumberinfootnote\Xnonumber
10801 \let\Xnonbreakableafternumber\Xnonbreakableafternumber
10802 \let\XmaxhXnotes\Xmaxhnotes
10803 \let\XbeforeXnotes\Xbeforenotes
10804 \let\Xboxlinenum\Xboxlinenum
10805 \let\Xboxlinenumalign\Xboxlinenumalign
10806 \let\Xboxstartlinenum\Xboxstartlinenum
10807 \let\Xboxendlinenum\Xboxendlinenum
10808 \let\Xtwolines\Xtwolines
10809 \let\Xmorethantwolines\Xmorethantwolines
10810 \let\Xtwolinesbutnotmore\Xtwolinesbutnotmore
10811 \let\Xtwolinesonlyinsamepage\Xtwolinesonlyinsamepage
10812 \fi
10813
10814 \unless\ifnofamiliar@
10815 \let\XnotesXwidthliketwocolumns\XnoteswidthliketwocolumnsX
10816 \fi
10817 \newcommandx{\parafootsep}[2][1,usedefault]{%
10818 \Xparafootsep[#1]{#2}%
10819 \parafootsepX[#1]{#2}
10820 }%
10821
10822 \newcommandx{\afternote}[2][1,usedefault]{%
10823 \Xafternote[#1]{#2}%
10824 \afternoteX[#1]{#2}%
10825 }%
10826
10827 \unless\ifnoend@
10828 \let\XendXtwolines\Xendtwolines
10829 \let\XendXmorethantwolines\Xendmorethantwolines
10830 \let\XhookXendnote\Xendhooknote
10831 \let\XboxXendlinenum\Xendboxlinenum%
10832 \let\XboxXendlinenumalign\Xendboxlinenumalign%
10833 \let\XboxXendstartlinenum\Xendboxstartlinenum%
10834 \let\XboxXendendlinenum\Xendboxendlinenum%
10835 \let\XendXlemmaseparator\Xendlemmaseparator
10836 \let\XendXbeforelemmaseparator\Xendbeforelemmaseparator
10837 \let\XendXafterlemmaseparator\Xendafterlemmaseparator
10838 \let\XendXinplaceoflemmaseparator\Xendinplaceoflemmaseparator
10839 \fi
10840
10841 \AtBeginDocument{%
10842 \ifdef\lineref{\let\lineref\edlineref}%
10843 }%
10844

```

```
10845  
10846 \fi%  
10847 %
```

```
</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:

```
I saw my friend \critext{Smith}          1 I saw my friend
\Afootnote{Jones C, D.}/                 2 Smith on Tuesday.
on Tuesday.                              2 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 `\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:

```
\critext{I saw my friend                 1 I saw my friend
  \critext{Smith}{\Afootnote{Jones       2 Smith on Tuesday.
C, D.}/ on Tuesday.}                   2 Smith] Jones C, D.
\Bfootnote{The date was                 1-2 I saw my friend
July 16, 1954.}                          Smith on Tuesday.] The
/                                          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  $\text{\TeX}$  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 `\XXXXXfmt` 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. 29](#)). Otherwise please add a new ticket on Github to request a new function for doing this.<sup>41</sup>

If for some reason you do not want to make the modifications to use the new functions of *eledmac*, you can continue using your own `\XXXXXfmt` command, but you must replace:

```
\renewcommand*{XXXXfmt}[3]
```

with

```
\renewcommandx*{XXXXfmt}[4][4=Z]
```

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. 72.

### A.5 Migration to eledmac 17.1

This version changes the default setting of `\Xpstart`. Henceforth, `pstart` numbers will be printed in footnotes within the section of text where you have called `\numberpstarttrue`.

We do not see any reason to print them in the other sections. However, if you want to print the `\pstart` numbers in 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>\interparanoteglu</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>noend</code>
<code>\pageparbreak</code>	<code>\ledpb</code>

The `ledsecnolinenumber` option has been removed, because it was related to deprecated commands.

The `oldprintnpnumspace` option has been removed too, because it was related to a historical bug. The `\usingedtext` and `\usingcritext` commands are also deprecated.

### A.9.4 `\renewcommand` replaced by command

Many uses of `\renewcommand` have been replaced with uses of specific commands. Please read handbook about specific commands.

<i>Deprecated <code>\renewcommand</code></i>	<i>Replaced with</i>
<code>\@led@extranofeet</code>	<code>\newseries</code>
<code>\apprefprefixmore</code>	<code>\setapprefprefixmore</code>
<code>\apprefprefixsingle</code>	<code>\setapprefprefixsingle</code>
<code>\endstanzaextra</code>	Optional argument of <code>\&amp;</code>
<code>\hangingsymbol</code>	<code>\sethangingsymbol</code>
<code>\ledfootinsdim</code>	<code>\Xmaxhnotes</code> and <code>\maxhnotesX</code>
<code>\parafootftmsep</code>	<code>\Xparafootsep</code> and <code>\parafootsepX</code>
<code>\notenumfont</code>	<code>\Xnotenumfont</code> , <code>\Xendnotenumfont</code> and <code>\notenumfontX</code>



<i>Deprecated</i> <code>\renewcommand</code>	<i>Replaced with</i>	
<code>\notefontsetup</code>	<code>\Xnotefontsize,</code>	
	<code>\Xendnotefontsize</code>	and
	<code>\notefontsizeX</code>	
<code>\sidenoteseq</code>	<code>\setsidenotsep</code>	
<code>\startstanzahook</code>	Optional argument of <code>\stanza</code>	
<code>\symplinenum</code>	<code>\Xsymplinenum</code>	

### 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}{\langle...\rangle}` to the use of `\Xlinrangeseparator` or `/` and `\Xendlinrangeseparator` (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. 43.

`\xlineref` does not print anymore the side flag (R for right side), because it is incompatible with numerical test. Use `\xflagref` to obtain it.

The `\printlines` and `\printendlines` commands take now an eighth argument, which is the side flag. It is strongly recommended to NEVER redefine these two commands and to use the setting commands instead (or to ask for new setting commands if the actual does not answer to your needs). However, if you have done it, just change your redefinition to have a new argument.

It is strongly recommended to stop redefining `\fullstop` and to use `\Xsublinesep` instead.

### A.15 Migration to reledmac 2.7.0

`\Serefonlypage` (introduced in reledmac 2.5.0) added an parenthesis after the page number. This was just an error, linked to a bad imitation of `\Serefwithpage`. That has been deleted. And so, the `\XendafterpagenumberSerefonlypage` to set it was also deleted.

`\rigidbalance` is split to two new commands: `\Xrigidbalance` for critical footnotes and `\rigidbalanceX` for familiar footnotes. If you have redefined it — but why should you have? —, you should split your single redefinition in two redefinitions.

### A.16 Migration to reledmac 2.7.2

`\Xhsize` is already defined in the `floatrow` package. It becomes `\Xwidth`, and, consequently, `\hsizeX` becomes `\widthX`.

The ancient names are temporarily maintained as aliases.

### A.17 Migration to reledmac 2.8.0

Reledmac 2.8.0 fix spurious indents for paragraphed critical and familiar footnotes in `ledgroup` and `minipage`. You can re-establish the indent with `\Xparindent` and `\parindentX`.

### A.18 Migration to reledmac 2.13.1

Reledmac 2.5.0 added a bug, which makes the right flag to be printed on the right side of critical footnotes, even if not explicitly requested by using `\Xlineflag`.

Version 2.13.1 solves this issue. Please use `\Xlineflag` if you want to add the right flag.

### A.19 Migration to reledmac 2.18.0

After updating reledmac, and before any new compilation, you need to clean your `.aux` files, if you use `\edlabel` or related.

### A.20 Migration to reledmac 2.21.0

Previously, there was a bug, which meant that the description in the handbook was incorrect. If you wrote

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

“elephant” was indexed in the main text and in the critical footnotes. With the new version of reledmac, it is indexed only in main text. If you also want to index it in critical footnotes, do

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

### A.21 Migration to reledmac 2.24.0

When using `\labelpstarttrue`, a spurious space<sup>a</sup> was introduced after the `pstart` number (only for normal typesetting, not while typesetting in parallel). The new version of the package has deleted this spurious space. If you consider that it was NOT a spurious space, you should add it manually in your definition of `\thepstart`.

### A.22 Migration to reledmac 2.26.0

You must delete your `.aux` file after having upgraded to this new version of reledmac.

### A.23 Migration to reledmac 2.27.1

This release fixes spurious space in `\hidenumbering`. If you considered this spurious space as normal, you must insert it manually using.

```
xx\hidenumbering\ xx
```

To get the space after `xx`.

### A.24 Migration to reledmac 2.30.0

If you have multiple annotations for the same line, these are now separated with a comma in the margin. You can use `\setlinenumannotationsep` to change the separator.

### A.25 Migration to reledmac 2.31.1

If you use `\Xgroupbyline`, the indentation of the notes has been deleted, in order to have the same behavior as for normal critical notes.

Use `\Xparindent` to restore indentation.

### A.26 Migration to reledmac 2.32.7

For paragraphed familiar footnotes, the separator between notes was `\Xparafootsep` instead of `\parafootsepX`. This release fixes this bug. You may adapt your settings.

### A.27 Migration to reledmac 2.33.0

You should not redefine anymore `\ledinnotemark`, but use instead `\innotemarkX` and `\xinnotemark`.

If you have personal script which parses for `\ledinnote` or if you redefine `\ledinnote` (but normally, you should not), this macro have new arguments. Look at the code for more details.

## Appendix B Auxiliary softwares

This appendix lists some software and tools related to reledmac that may be useful.

### B.1 *samewords*

The *samewords* program (Michael Stenskjær Christensen) automatically adds `\sameword` commands (6.3 p. 32) 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 org-critical-edition for *Emacs* and *Org Mode*

This *Emacs* library lets you prepare a critical edition in *Org Mode* with  $\LaTeX$  and *reledmac* output (Juan Manuel Macías Chaín) <https://gitlab.com/maciaschain/org-critical-edition>

<https://gitlab.com/maciaschain/org-critical-edition>

## 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 customizing 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>\@hangingsymbol</code>	1
<code>\@iiiminipage</code>	1
<code>\@insertstanzanumber</code>	1
<code>\@insidethisline</code>	1
<code>\@k</code>	1
<code>\@l@dttempcnta</code>	1
<code>\@l@dttempcntb</code>	1
<code>\@lab</code>	1
<code>\@led@testifnofoot</code>	1
<code>\@lemma</code>	1
<code>\@line@num</code>	1
<code>\@linenumannotationsep</code>	1
<code>\@lock</code>	1
<code>\@lopL</code>	1
<code>\@lopR</code>	1
<code>\@makechapterhead</code>	1, 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
\@xympar	1
CLASSarticle	74
CLASSbook	74, 365
CLASSmemoir	214, 215, 278–281, 315, 365, 384, 448, 451
CLASSscrbook	452
COMMAND\*footnote	75
COMMAND\...@footnotemark...	217
COMMAND\...d@ta	158
COMMAND\ <hook< td=""> <td></td> </hook<>	
@<series	265
COMMAND\ <hookname< td=""> <td></td> </hookname<>	
<pseudoserious	267, 268
COMMAND\ <type< td=""> <td></td> </type<>	
footfmt	201
COMMAND\@@line	191
COMMAND\@MM	179, 449
COMMAND\@Rlineflag	317, 449
COMMAND\@SErefprefix	295
COMMAND\@SErefprefixmore	295
COMMAND\@add@	355
COMMAND\@adv	114
COMMAND\@annot	164
COMMAND\@annot@end@print	204
COMMAND\@annot@start@print	204, 209, 247
COMMAND\@apprefprefixmore	295
COMMAND\@apprefprefixsingle	295
COMMAND\@beforeinsertofthisedtext	137, 138
COMMAND\@bsphack	283
COMMAND\@docclearpage	279, 280, 442, 451
COMMAND\@doreinfeetX	451
COMMAND\@dprintingcolumns	448
COMMAND\@edindex@hyperref	317, 318
COMMAND\@edtext@level	134
COMMAND\@esphack	283
COMMAND\@firstofone	36
COMMAND\@fnpos	235, 275
COMMAND\@footnotemark	215, 442, 451
COMMAND\@footnotetext	215, 216, 442
COMMAND\@gobble	26, 35, 132, 164, 260
COMMAND\@gobblefive	450
COMMAND\@gobblefour	448
COMMAND\@gobbleseven	262
COMMAND\@gobblethree	441
COMMAND\@h	194
COMMAND\@hangingsymbol	323
COMMAND\@iiiminipage	307, 309, 441, 451
COMMAND\@iiiminpage	307
COMMAND\@l	446
COMMAND\@l@dtempcnta	161, 166, 173
COMMAND\@l@dtempcntb	166
COMMAND\@l@reg	446
COMMAND\@lab	110, 283, 286, 290, 441
COMMAND\@ldunboxmpfoot	309

COMMAND\@led@extranofeet	384
COMMAND\@ledinnote@command	313, 314
COMMAND\@lemma	137, 140
COMMAND\@lock	105, 323
COMMAND\@lopL	442
COMMAND\@lopR	442
COMMAND\@makecol	275, 276, 278, 451
COMMAND\@mempnum	281
COMMAND\@mpfnpos	235
COMMAND\@msd	333
COMMAND\@msd@c	333
COMMAND\@msd@options@iffullpage	339
COMMAND\@msdata@list	333, 334
COMMAND\@nl	110–112, 114, 116, 125, 165, 286, 441, 442
COMMAND\@nl@reg	111, 382, 442, 446
COMMAND\@opXfeet	442
COMMAND\@opfeetX	451
COMMAND\@opxtrafeeti	451
COMMAND\@outputpage	281
COMMAND\@page	111, 112, 286
COMMAND\@pend	442
COMMAND\@pendR	442
COMMAND\@ref	110, 120–122, 126, 132
COMMAND\@ref@later	122, 127
COMMAND\@ref@reg	121, 164, 442
COMMAND\@ref@reg@parsearg	122
COMMAND\@reinserts	275, 277, 278, 451
COMMAND\@resetannot	165
COMMAND\@secondoftwo	76
COMMAND\@sect	367
COMMAND\@series	264
COMMAND\@set	115
COMMAND\@sidenotese	305
COMMAND\@stopmsd	334
COMMAND\@stpelt	376
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	299, 451
COMMAND\Aendnote	16, 29
COMMAND\Afootfmt	178
COMMAND\Afootgroup	178
COMMAND\Afootnote	8, 16, 28, 29, 31, 135, 186, 214, 236, 254, 451
COMMAND\Afootstart	178
COMMAND\AtBeginDocument	278
COMMAND\AtEndEveryPend	19, 457
COMMAND\AtEveryPend	18, 19, 57, 152, 448, 449, 451, 457
COMMAND\AtEveryPend*	19
COMMAND\AtEveryPstart	18, 19, 57, 448, 449, 451, 454, 457

COMMAND\AtEveryPstart*	19
COMMAND\AtEveryStanza	57, 454, 457
COMMAND\AtEveryStopStanza	57, 454, 457
COMMAND\AtStartEveryPstart	19, 457
COMMAND\AtStartEveryStanza	57, 457
COMMAND\BeforeEveryStopStanza	57, 457
COMMAND\Bendnote	16, 28
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	367
COMMAND\Hy@raisedlink	376
COMMAND\Hy@raisedlink@left	376
COMMAND\LTR	50
COMMAND\Pages	90, 148, 276, 277, 280
COMMAND\ProcessOptionsX	80
COMMAND\RL	49
COMMAND\RaggedLeft	50
COMMAND\RaggedRight	50
COMMAND\RenewExpandableDocumentCommand	36, 144
COMMAND\SEonlypage	292, 453
COMMAND\SEref	62–64, 292, 296, 454, 456, 459
COMMAND\SErefonlypage	62–64, 387, 453
COMMAND\SErefwithpage	62–64, 292, 296, 387, 453, 455, 460
COMMAND\Stanza	447
COMMAND\Waklam	356
COMMAND\X@doreinfeet	277, 451
COMMAND\XR@prefix	299
COMMAND\XR@test	298, 299
COMMAND\XR@test@mac	299
COMMAND\XR@test@mac@test	299
COMMAND\XXXXXXfmt	381
COMMAND\XXXXXXfmt	381
COMMAND\Xafterenumber	45
COMMAND\Xafterlemmaseparator	47, 385
COMMAND\Xafternote	51, 384, 385
COMMAND\Xafternumber	45, 46, 385
COMMAND\Xafterrule	52, 237, 385, 446, 449
COMMAND\Xaftersymmlinum	46, 385
COMMAND\Xarrangement	41, 51, 75, 179, 180, 266, 385
COMMAND\Xarrangement@footparagraph	185
COMMAND\Xarrangement@normal	180
COMMAND\Xarrangement@paragraph	184
COMMAND\Xbeforeinserting	50
COMMAND\Xbeforelemmaseparator	47, 385
COMMAND\Xbeforenotes	52, 236, 385, 447, 449
COMMAND\Xbeforenumber	45, 46, 385
COMMAND\Xbeforepagenumber	42, 461
COMMAND\Xbeforesymmlinum	46, 385
COMMAND\Xbhookendnote	385

COMMAND\Xbhookgroup	52, 453, 454
COMMAND\Xbhooknote	50, 385–387, 452
COMMAND\Xboxendlinenum	46, 47, 385, 450
COMMAND\Xboxlinenum	46, 47, 385
COMMAND\Xboxlinenumalign	46, 47, 385, 450
COMMAND\Xboxstartlinenum	46, 47, 385, 450
COMMAND\Xboxsymlinenum	46, 385
COMMAND\Xcolalign	50, 449
COMMAND\Xdo@feet	451, 456
COMMAND\Xend	262
COMMAND\XendXafterlemmaseparator	386
COMMAND\XendXbeforelemmaseparator	386
COMMAND\XendXinplaceoflemmaseparator	386
COMMAND\XendXlemmaseparator	386
COMMAND\XendXmorethantwolines	386
COMMAND\XendXtwolines	386
COMMAND\Xendafterenumber	45, 452
COMMAND\Xendafterlemmaseparator	48, 386
COMMAND\Xendafternote	54, 454
COMMAND\Xendafternumber	49
COMMAND\Xendafterpagenumbe	455
COMMAND\Xendafterpagenumber	43, 63
COMMAND\XendafterpagenumberSerefonlypage	387
COMMAND\Xendaftersymlinenum	46, 49, 452
COMMAND\Xendahookinplaceofnumber	49, 452
COMMAND\Xendahooklinenum	49, 452
COMMAND\Xendbeforelemmaseparator	47, 386
COMMAND\Xendbeforelinenum	49
COMMAND\Xendbeforenumber	45, 452
COMMAND\Xendbeforepagenumber	43, 63
COMMAND\XendbeforepagenumberSerefonlypage	63
COMMAND\Xendbeforesymlinenum	46, 49, 452
COMMAND\Xendbhookinplaceofnumber	49, 452
COMMAND\Xendbhooklinenum	49, 452
COMMAND\Xendbhooknote	50
COMMAND\Xendboxendlinenum	47, 450
COMMAND\Xendboxlinenum	47, 385, 449
COMMAND\Xendboxlinenumalign	47, 385, 450
COMMAND\Xendboxstartlinenum	47, 450
COMMAND\Xendboxsymlinenum	46, 452
COMMAND\Xendfootnote	133
COMMAND\Xendhangindent	49, 452, 454
COMMAND\Xendinplaceoflemmaseparator	29, 48, 386
COMMAND\Xendinplaceofnumber	46, 451
COMMAND\Xendinplaceofpagenumber	42, 456
COMMAND\Xendinsertsep@	243
COMMAND\Xendlemmadisablefontselection	48
COMMAND\Xendlemmafont	48, 453
COMMAND\Xendlemmaseparator	30, 47, 386
COMMAND\Xendletboxendlinenum	385
COMMAND\Xendlineflag	63
COMMAND\Xendlinenumannotationonlyfirst	45, 247, 459
COMMAND\Xendlinenumannotationonlyfirstintwo	45, 247, 459
COMMAND\Xendlinenumannotationposition	25, 64

COMMAND\Xendlineprefixmore	43, 63
COMMAND\Xendlineprefixsingle	43, 63
COMMAND\Xendlinerangeseparator	43, 63, 199, 387, 453
COMMAND\Xendmorethantwolines	29, 43, 63, 386, 449, 451
COMMAND\Xendnoidenticallinenumannotation	26, 64, 247, 459
COMMAND\Xendnolinenum	461
COMMAND\Xendnolinenumifannotation	45, 461
COMMAND\Xendnonumber	44, 451
COMMAND\Xendnotboxingsubline	47, 461
COMMAND\Xendnote	239, 261, 262, 449
COMMAND\Xendnotefontsize	48, 385
COMMAND\Xendnotenumfont	48, 49, 384
COMMAND\Xendnotes	242
COMMAND\Xendnumberonlyfirstinline	42, 452
COMMAND\Xendnumberonlyfirstintwolines	42, 452
COMMAND\Xendpagenumberonlyfirst	42, 456
COMMAND\Xendpagenumberonlyfirstifsingle	42, 456
COMMAND\Xendpagenumberonlyfirstintwo	42, 456
COMMAND\Xendparagraph	54, 446
COMMAND\Xendsep	54, 459
COMMAND\Xendstorelineinfo	241
COMMAND\Xendsublinesep	45, 63, 199
COMMAND\Xendsymmlinum	42, 452, 459
COMMAND\Xendsymmlinumannotation	45, 459
COMMAND\Xendsympagenum	42, 456
COMMAND\Xendtwolines	29, 43, 63, 386, 449, 451
COMMAND\Xendtwolinesbutnotmore	43, 63, 449, 451
COMMAND\Xendtwolinesonlyinsamepage	43, 64, 449, 451
COMMAND\Xendtxtbeforenotes	51, 458
COMMAND\Xendwrapcontent	49, 454
COMMAND\Xendwraplemma	49, 454
COMMAND\Xendwraplinenumannotation	26, 64
COMMAND\Xfootfmt	386
COMMAND\Xfootgroup	184
COMMAND\Xfootins	183
COMMAND\Xfootnote	60, 66, 133, 381, 444, 447, 449, 453, 455
COMMAND\Xfootstarts	184
COMMAND\Xgroupbyline	51, 172, 213, 273, 389, 458–461
COMMAND\Xgroupbylines	456
COMMAND\Xgroupbylineseparetwolines	51
COMMAND\Xhangindent	49, 452, 459
COMMAND\Xhsize	388, 453, 454
COMMAND\Xsizethreecol	50, 53, 385
COMMAND\Xsizetwocol	50, 53, 267, 385
COMMAND\Xinnotemark	66, 389, 460
COMMAND\Xinplaceoflemmaseparator	29, 47, 385
COMMAND\Xinplaceofnumber	29, 46, 385, 449, 450
COMMAND\Xinsertparafootsep	189, 190
COMMAND\Xledsetnormalparstuff	383, 384, 450
COMMAND\Xlemmadisablefontselection	48
COMMAND\Xlemmafnt	48, 453
COMMAND\Xlemmaseparator	47, 199, 268, 271, 275, 385
COMMAND\Xlineflag	63, 388, 455
COMMAND\Xlinenumannotationonlyfirst	44, 45, 209, 459

COMMAND\Xlinenumannotationonlyfirstintwo	45, 209, 459
COMMAND\Xlinenumannotationposition	25, 63, 164
COMMAND\Xlinenumannotationposition@side	164
COMMAND\Xlinenumannotationpositionside	25
COMMAND\Xlinerangeseparator	43, 63, 199, 387, 453
COMMAND\Xmaxhnotes	53, 74, 75, 384, 385, 447, 449
COMMAND\Xmorethantwolines	29, 43, 63, 386, 449
COMMAND\Xnoidenticallinenumannotation	26, 63, 208, 459
COMMAND\Xnoindent	452
COMMAND\Xnolemmaseparator	47, 275, 386
COMMAND\Xnolinenum	204, 461
COMMAND\Xnolinenumifannotation	45, 204, 461
COMMAND\Xnonbreakableafternumber	45, 386, 445
COMMAND\Xnonote	41, 461
COMMAND\Xnonumber	43, 386
COMMAND\Xnonumberinfootnote	386
COMMAND\Xnopagenumberifcurrent	44, 211, 461
COMMAND\Xnotboxingsubline	47, 461
COMMAND\Xnotefontsize	48, 385
COMMAND\Xnotefontsize@{s}	189, 193, 194
COMMAND\Xnotenumfont	48, 384
COMMAND\Xnoteswidthliketwocolumns	53, 447
COMMAND\Xnumberonlyfirstinline	41, 42, 51, 107, 200, 241, 268, 271, 386, 444, 448, 457
COMMAND\Xnumberonlyfirstintwolines	42, 51, 386, 444, 457
COMMAND\Xonlyonerule	235, 462
COMMAND\Xonlyonerule@	235
COMMAND\Xonlypstart	44, 386, 444, 448
COMMAND\Xpagelinesep	45, 456
COMMAND\Xparafootsep	51, 108, 384, 386, 389, 455, 457, 458
COMMAND\Xparafootsep@series	189
COMMAND\Xparindent	49, 388, 389, 449, 452, 454, 459
COMMAND\Xprenotes	52, 237, 455
COMMAND\Xprenotes@	183, 237, 444
COMMAND\Xprintlines	107
COMMAND\Xpstart	44, 382, 386, 444, 448
COMMAND\Xpstarteverytime	44, 382, 386, 448
COMMAND\Xpstartonlyfirst	44, 460
COMMAND\Xpstartseparator	44, 460
COMMAND\Xragged	51
COMMAND\Xrigidbalance	191, 388, 454
COMMAND\Xsethangindent	198
COMMAND\Xstanza	44, 58
COMMAND\Xstanzaonlyfirst	44, 460, 461
COMMAND\Xstanzaseparator	44
COMMAND\Xstorelineinfo	200
COMMAND\Xsublinesep	22, 45, 63, 199, 387
COMMAND\Xsublinesepside	22, 45
COMMAND\Xsymlinenum	42, 51, 385, 386, 450, 457
COMMAND\Xsymlinenumannotation	45, 459
COMMAND\Xtextbeforenotes	174
COMMAND\Xtoendnotes	30, 262
COMMAND\Xtwolines	29, 43, 63, 208, 209, 267, 386, 449
COMMAND\Xtwolinesappref	267
COMMAND\Xtwolinesbutnotmore	43, 63, 386, 449

COMMAND\Xtwolinesbutnotmoreappref	268
COMMAND\Xtwolinesonlyinsamepage	43, 63, 386, 449
COMMAND\Xtxtbeforenotes	51, 52, 386, 455–457
COMMAND\Xtxtbeforenotesonlyonce	52, 457
COMMAND\Xtxtbeforenumber	42
COMMAND\Xunvxh	187, 386
COMMAND\Xwidth	53, 388, 454
COMMAND\Xwrapcontent	49, 454
COMMAND\Xwraplemma	49, 50, 454
COMMAND\Xwraplinenumannotation	26, 63, 164
COMMAND\Xwraplinenumannotation@ref	164
COMMAND\Xwraplinenumannotation@side	164
COMMAND\Xwraplinenumannotationref	26
COMMAND\Xwraplinenumannotationside	26
COMMAND\&	384
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	273
COMMAND\add@inserts	105, 171, 172
COMMAND\add@inserts@next	171, 172
COMMAND\add@msd@	333
COMMAND\add@msdata	333, 334
COMMAND\add@msdata@firstlineofpage	337
COMMAND\add@msddata	333
COMMAND\add@penalties	160, 173, 447
COMMAND\addcontentsline	362
COMMAND\addfootins	381, 384
COMMAND\addfootinsX	381, 384
COMMAND\addtoendnotes	262
COMMAND\advancelabel@refs	285
COMMAND\advanceline	23, 106, 114, 128, 452
COMMAND\affixline@num	165, 168, 170, 305, 442
COMMAND\affixpstart@num	170
COMMAND\afterXrule	385
COMMAND\afterXsymlinenum	385
COMMAND\afterantilabe	58
COMMAND\aftergroup	131, 136
COMMAND\afterlemmaseparator	385
COMMAND\afternote	385
COMMAND\afternoteX	51, 384, 385
COMMAND\afternumberinfootnote	385
COMMAND\afterruleX	52, 446, 449
COMMAND\annot@end@print	209, 247
COMMAND\annot@list	107, 165
COMMAND\annotationref	26, 60
COMMAND\antilabe	15, 58, 460
COMMAND\applabel	62, 137, 286, 288, 449, 455, 461
COMMAND\appref	60, 62–64, 292, 296, 453, 454, 459
COMMAND\apprefprefixmore	62, 384
COMMAND\apprefprefixsingle	62, 384
COMMAND\apprefwithpage	62–64, 292, 296, 451, 453, 460

COMMAND\arrangementX	41, 75, 218, 266, 385
COMMAND\arrangementX@normal	223
COMMAND\article	15
COMMAND\at@every@pend	152
COMMAND\autopar	18, 148, 153, 154, 233, 443, 445, 446, 449, 458
COMMAND\ballast	75
COMMAND\ballast@count	160, 173
COMMAND\baselineskip	41, 185, 189
COMMAND\beforeXnotes	385
COMMAND\beforeXsymlinenum	385
COMMAND\beforeantilabe	58
COMMAND\beforeelectedchapter	10, 72, 363
COMMAND\beforeinsertingX	50
COMMAND\beforelemmaseparator	385
COMMAND\beforenotesX	52, 387, 445, 446, 449
COMMAND\beforenumberinfootnote	385
COMMAND\begin	341
COMMAND\beginnumbering	16–20, 90–92, 95, 96, 104, 108, 123, 124, 153, 239, 332, 443, 446, 450, 451, 456
COMMAND\beginnumberingR	90
COMMAND\bf	443
COMMAND\bfseries	48, 443
COMMAND\bhookXnote	385
COMMAND\bhookgroupX	52, 453
COMMAND\bhooknoteX	50, 386, 387, 452
COMMAND\body	324
COMMAND\bodyfootmarkA	38
COMMAND\book	15
COMMAND\boxXendlinenum	385
COMMAND\boxXendlinenumalign	385
COMMAND\boxXendstartlinenum	385
COMMAND\boxendlinenum	385
COMMAND\boxlinefootnote	202
COMMAND\boxlinenum	385
COMMAND\boxlinenumalign	385
COMMAND\boxstartlinenum	385
COMMAND\boxsymlinenum	385
COMMAND\break	41, 187
COMMAND\brokenpenalty	173
COMMAND\c@page	125
COMMAND\c@page@page	125
COMMAND\centering	50
COMMAND\ch@ck@l@ck	442
COMMAND\ch@cksub@l@ck	168, 442
COMMAND\chapter	71, 365, 368, 447, 449, 452, 455
COMMAND\chaptermark	363
COMMAND\check@pb@in@verse	375
COMMAND\colalignX	50, 449
COMMAND\collect@body	342
COMMAND\color	454
COMMAND\colorbox	75
COMMAND\columns	53
COMMAND\columnwidth	185, 447
COMMAND\command names	267, 268



COMMAND\continuousnumberingwithcolumns@sync@linenumber@singletext	162
COMMAND\copyright	132
COMMAND\correct@Xfootins@box	449
COMMAND\correct@footinsX@box	449
COMMAND\count	192, 193
COMMAND\critex	443
COMMAND\critext	140, 380, 381, 384
COMMAND\csname	78
COMMAND\csquotes	258
COMMAND\ctab	356, 357, 361
COMMAND\ctabtext	361
COMMAND\current@annot	165
COMMAND\dcoll	350
COMMAND\def	77, 93
COMMAND\dimen	192
COMMAND\dimexpr	53
COMMAND\discretionary	186
COMMAND\displaywidowpenalty	173
COMMAND\do@Xfeet	276, 442, 451, 456
COMMAND\do@actions	159, 160, 162, 442
COMMAND\do@actions@fixedcode	442
COMMAND\do@actions@next	160, 161
COMMAND\do@ballast	159, 160, 173
COMMAND\do@feet@custom@order	275
COMMAND\do@insidelinehook	445
COMMAND\do@line	105, 131, 150, 154, 157, 171, 173, 323, 442, 443, 445, 446
COMMAND\do@linehook	442
COMMAND\do@lockoff	106
COMMAND\do@lockon	106
COMMAND\dodoreintrafeet	441
COMMAND\doendnotes	29, 243, 386, 450, 457
COMMAND\doendnotesbysection	29, 243, 262, 450, 457
COMMAND\doennotes	456
COMMAND\doinsidelinehook	27, 447
COMMAND\doinsidethislinehook	27, 158, 459
COMMAND\dolinehook	27, 447
COMMAND\doreintrafeeti	451
COMMAND\doreintrafeetii	451
COMMAND\doxtrafeet	441
COMMAND\doxtrafeeti	451
COMMAND\doxtrafeetii	451
COMMAND\dummy@ref	132
COMMAND\edaftertab	71, 356, 357
COMMAND\edannotationref	290
COMMAND\edatleft	70, 354
COMMAND\edatright	70, 71, 355
COMMAND\edbeforetab	70, 356
COMMAND\edfilldimen	355
COMMAND\edfont@info	139
COMMAND\edgls	68, 312, 461
COMMAND\edgls...	454
COMMAND\edglsadd	456
COMMAND\edindex	36, 66–68, 312, 313, 316, 318, 346, 445, 447, 448, 451, 452, 455–457, 460
COMMAND\edindexlab	68

- COMMAND\edlabel . . . . . 60–62, 64, 132,  
     283, 285, 286, 289, 290, 298, 312, 345, 376, 388, 441, 443, 444, 446, 448, 453, 458, 461, 462  
 COMMAND\edlabelE . . . . . 62, 288  
 COMMAND\edlabelS . . . . . 62, 288  
 COMMAND\edlabelSE . . . . . 62  
 COMMAND\edlineref . . . . . 60, 283, 386, 448, 450, 453, 456  
 COMMAND\edmakelabel . . . . . 61, 298  
 COMMAND\edpageref . . . . . 60, 283, 289, 298  
 COMMAND\edrowfill . . . . . 356  
 COMMAND\edsublineref . . . . . 60  
 COMMAND\edtabcolsep . . . . . 350  
 COMMAND\edtext . . . . . 7, 24, 25, 28, 31, 33–35, 54,  
     60–62, 66, 69, 75, 94, 105, 107, 120, 122, 126, 127, 130–140, 142, 143, 145, 146, 163–165,  
     286–288, 291, 345, 347, 365, 376, 380, 381, 384, 441, 443, 444, 447–450, 455–458, 460, 461  
 COMMAND\edtext@level . . . . . 450  
 COMMAND\edtext@notes . . . . . 133, 134  
 COMMAND\edtext@notes@start . . . . . 133, 134  
 COMMAND\edtextlater . . . . . 122  
 COMMAND\edvertdots . . . . . 71, 355  
 COMMAND\edvertline . . . . . 71, 355  
 COMMAND\elechapter . . . . . 72  
 COMMAND\eled@sectioning@out . . . . . 369  
 COMMAND\eledchapter . . . . . 72, 384, 447, 452  
 COMMAND\eledchapter\* . . . . . 72  
 COMMAND\eledmac@error . . . . . 441  
 COMMAND\eledsection . . . . . 7, 16, 72, 132, 156, 363, 384, 449, 457, 458, 460, 461  
 COMMAND\eledsection\* . . . . . 72  
 COMMAND\eledsubsection . . . . . 72, 384  
 COMMAND\eledsubsection\* . . . . . 72  
 COMMAND\eledsubsubsection . . . . . 72, 384  
 COMMAND\eledsubsubsection\* . . . . . 72  
 COMMAND\eledxxx . . . . . 10, 72, 369, 446, 447  
 COMMAND\eledxxxx . . . . . 363  
 COMMAND\elesection . . . . . 369  
 COMMAND\else . . . . . 311, 363  
 COMMAND\emph . . . . . 36  
 COMMAND\empty . . . . . 89, 166, 167, 283  
 COMMAND\end . . . . . 341  
 COMMAND\end@lemmas . . . . . 131  
 COMMAND\endashchar . . . . . 199  
 COMMAND\endgraf . . . . . 150, 189, 233  
 COMMAND\endlock . . . . . 22, 106, 129, 328  
 COMMAND\endminipage . . . . . 307, 309, 441, 451  
 COMMAND\endmsdata . . . . . 37  
 COMMAND\endnotes . . . . . 450, 453, 454  
 COMMAND\endnumbering . . . . . 16–18, 20, 91, 93, 94, 124, 442, 450, 456  
 COMMAND\endnumberingR . . . . . 90  
 COMMAND\endprint . . . . . 239, 242, 262, 383  
 COMMAND\endstanzaextra . . . . . 384  
 COMMAND\endsub . . . . . 22, 106, 128  
 COMMAND\endverse . . . . . 387  
 COMMAND\everypar . . . . . 153  
 COMMAND\extensionchars . . . . . 73, 91  
 COMMAND\externaldocument . . . . . 64, 298

COMMAND\fax@l@cks	442
COMMAND>falseverse	384, 445, 447
COMMAND\fi	363
COMMAND\firstlinenum	21, 166, 443
COMMAND\firstsublinenum	21, 443
COMMAND\flag@end	126, 127, 138, 447
COMMAND\flag@end@RTL	127
COMMAND\flag@end@later	127
COMMAND\flag@start	126, 127, 138, 447
COMMAND\flag@start@RTL	127
COMMAND\flag@start@later	127
COMMAND\flagstanza	59
COMMAND\floatingpenalty	179, 449
COMMAND\flush@notes	174
COMMAND\fnpos	40, 235, 446, 456
COMMAND\footfmt	178, 181
COMMAND\footfmt...	218
COMMAND\footfootmarkA	38
COMMAND\footfudgefactor	187
COMMAND\footfudgefiddle	75, 185, 441
COMMAND\footgroup	178
COMMAND\footins	183
COMMAND\footnormal	266, 385, 442
COMMAND\footnormalX	385
COMMAND\footnote	38, 74, 214–216, 381, 442
COMMAND\footnote@lang	199
COMMAND\footnoteA	16, 38
COMMAND\footnoteB	16
COMMAND\footnoteC	28
COMMAND\footnoteE	38
COMMAND\footnoteX	8, 38, 258, 260, 455
COMMAND\footnoteX@reading	260
COMMAND\footnoteX@series	260
COMMAND\footnoteXmark	260, 457, 460, 461
COMMAND\footnoteXmk	275
COMMAND\footnoteXtext	260, 457, 460, 461
COMMAND\footnoteXtxt	462
COMMAND\footnote⟨X⟩	132
COMMAND\footnote⟨X⟩mark	39
COMMAND\footnote⟨X⟩mk	39
COMMAND\footnote⟨X⟩nomk	39
COMMAND\footnote⟨X⟩text	39
COMMAND\footnotelang@lua	177
COMMAND\footnotelang@poly	177
COMMAND\footnotemark	39, 260
COMMAND\footnoteoption@	176, 453
COMMAND\footnoterule	192
COMMAND\footnotesize	48
COMMAND\footnotetext	39, 260
COMMAND\footparagraph	185, 266, 385, 447
COMMAND\footparagraphX	228, 385, 446
COMMAND\footplitskips	442, 449
COMMAND\footstart	178, 183, 192
COMMAND\footstrut	189

COMMAND\footthreecol	385
COMMAND\footthreecolX	385, 449
COMMAND\foottwocol	385
COMMAND\foottwocolX	385, 449
COMMAND\foreignlanguage	49
COMMAND\fullstop	387
COMMAND\get@edindex@hyperref	317
COMMAND\get@edindex@ledinnote@command	313
COMMAND\get@fnmark	216
COMMAND\get@index@command	445
COMMAND\get@linelistfile	442
COMMAND\get@thisfootnote	222
COMMAND\getline@num	159, 161
COMMAND\gl@p	97
COMMAND\global	110
COMMAND\globaldefs	110
COMMAND\gls	68, 320
COMMAND\hangindentX	49, 449, 452, 459
COMMAND\hangingsymbol	384, 443
COMMAND\hbox	187
COMMAND\hfill	445
COMMAND\hidenumbering	23, 119, 389, 449, 459, 460
COMMAND\hidenumberingonleftpage	23, 119, 455
COMMAND\hidenumberingonrightpage	23, 119, 455
COMMAND\hline	69
COMMAND\hooktoggle@specific	273
COMMAND\hrulefill	356
COMMAND\hspace	41, 183, 185–187, 193, 196, 234, 442, 447
COMMAND\hspaceX	388, 453, 454
COMMAND\hsizethreecol	385
COMMAND\hsizethreecolX	51, 53
COMMAND\hsizetwocol	385
COMMAND\hsizetwocolX	51, 53
COMMAND\hyperlinkR	317
COMMAND\hyperlinkformat	317
COMMAND\hyperlinkformatR	317
COMMAND@if@RTL	81, 136
COMMAND@if@edtext@	448, 450
COMMAND@if@eled@sectioning	369
COMMAND@if@firstlineofpage	81
COMMAND@if@firstlineofpageR	81
COMMAND@if@msd@options@fullpage	339
COMMAND@if@msdata@insertedfrompreviouspage	337
COMMAND@if@nobreak	151
COMMAND@if@noneed@notes	134
COMMAND\ifXnote@	90
COMMAND\ifafterrendnumberingR	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	323
COMMAND\ifinstanza	323
COMMAND\ifistwofollowinglines	209
COMMAND\ifl@d@Xlinenumber	204
COMMAND\ifl@d@Xmorethantwolines	204, 449
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	261
COMMAND\ifl@dmemoir	441
COMMAND\ifl@dpaging	447
COMMAND\ifl@dpairing	90, 443
COMMAND\ifl@dprintingpages	448
COMMAND\ifl@dskipnumber	166
COMMAND\ifl@dstartendok	355
COMMAND\ifl@imakeidx	81
COMMAND\ifl@noexpands@edgl	322
COMMAND\ifledRcol	90, 443
COMMAND\ifledRcol@	90, 446
COMMAND\iflemmacommand@	448
COMMAND\iflinenumannotationotherside	164
COMMAND\ifnoend@	244
COMMAND\ifnoledgroup@	311
COMMAND\ifnoteschanged@	107
COMMAND\ifnumberedpar@	148
COMMAND\ifnumbering	91, 94
COMMAND\ifnumberingR	90, 443
COMMAND\ifnumberline	138, 166
COMMAND\ifpst@rted	443
COMMAND\ifpst@rtedL	92
COMMAND\ifresumenummering@start	95
COMMAND\ifseriesbefore	265
COMMAND\ifstopmsdata@inserted@	332
COMMAND\ifsublines@	104, 117
COMMAND\ifsublines@R	104
COMMAND\iftrue	450
COMMAND\ifvmode	285
COMMAND\ifxxx	363
COMMAND\ignorespaces	135
COMMAND\imki@wrindexentry	81
COMMAND\immediate	123, 124, 238
COMMAND\indent	18, 153, 387
COMMAND\index	319, 456
COMMAND\indtl@wrindexentry	81
COMMAND\initnumbering@quote	361, 451
COMMAND\initnumbering@reg	442
COMMAND\initnumbering@sectcmd	451
COMMAND\innotemarkX	66, 389, 460
COMMAND\inplaceoflemmaseparator	385

COMMAND\inplaceofnumber	385
COMMAND\insert	138, 171, 178, 181, 194, 213, 218
COMMAND\insert@Xtxtbeforenotes	175, 194
COMMAND\insert@count	120, 126, 135
COMMAND\insert@countR	135
COMMAND\insert@msdata	333, 339
COMMAND\insert@txtbeforenotesX	175
COMMAND\inserthangingsymbol	445
COMMAND\insertlines@list	105, 120, 121
COMMAND\insertparafootsepX	231
COMMAND\inserts@list	131, 148, 171, 186
COMMAND\interAfootnotelinepenalty	442
COMMAND\interfootnotelinepenalty	442
COMMAND\interlinepenalty	178
COMMAND\interparanoteglua	384
COMMAND\justifying	50
COMMAND\keep@this@crossref@forinserting	214
COMMAND\l@advance@parledgroup@beforenormalnotes	451
COMMAND\l@d@@wrindexhyp	448
COMMAND\l@d@add	141
COMMAND\l@d@end	239, 261
COMMAND\l@d@nums	135, 138, 140, 141, 204, 205
COMMAND\l@d@section	239
COMMAND\l@d@set	116, 129
COMMAND\l@dampcount	347
COMMAND\l@dbfnote	216, 442
COMMAND\l@dcheckstartend	355
COMMAND\l@dchset@num	116
COMMAND\l@dcolcount	347, 348
COMMAND\l@dcollect@@body	341
COMMAND\l@dcollect@body	341
COMMAND\l@dcsnote	446
COMMAND\l@dcsnotetext	158, 303
COMMAND\l@dcsnotetext@l	158, 303
COMMAND\l@dcsnotetext@r	158, 303
COMMAND\l@ddodoreintrafeet	277, 441
COMMAND\l@ddoxtrafeet	275, 277, 441
COMMAND\l@demptyd@ta	442
COMMAND\l@dend@close	238
COMMAND\l@dend@open	238
COMMAND\l@dend@stuff	239
COMMAND\l@denvbody	341
COMMAND\l@dfeetbeginmini	441
COMMAND\l@dfeetendmini	441
COMMAND\l@dgetline@margin	443
COMMAND\l@dgetlock@disp	443
COMMAND\l@dgetref@num	290, 291
COMMAND\l@dgetsidenote@margin	299, 443
COMMAND\l@dgobbeloptarg	447
COMMAND\l@dgobblearg	447
COMMAND\l@dgobbleoptarg	346
COMMAND\l@dlabel@parse	290, 291
COMMAND\l@dld@ta	165, 167
COMMAND\l@dlp@rbox	305

COMMAND\@dlsn@te	443
COMMAND\@dlsnote	446
COMMAND\@dmake@labels	285, 299
COMMAND\@dmake@labelsR	299
COMMAND\@dnumpstartsL	92, 443
COMMAND\@dp@rsefootspec	205
COMMAND\@dparsefootspec	205
COMMAND\@dpush@begins	341
COMMAND\@drd@ta	165, 167
COMMAND\@dref@undefined	290
COMMAND\@drsn@te	443
COMMAND\@drsnote	446
COMMAND\@dtabaddcols	355
COMMAND\@dtabnoexpands	441
COMMAND\@dunboxmpfoot	451
COMMAND\@dunboxmpfoot	442
COMMAND\@dunhbox@line	154
COMMAND\@dzeropenalties	443, 447
COMMAND\@noexpands@edgl	322
COMMAND\@pb	375
COMMAND\@prev@nopb	374
COMMAND\@prev@pb	374
COMMAND\@reg	382
COMMAND\@secondoptarg	346
COMMAND\@wrap@ifnotemptybox	89
COMMAND\label	19, 61, 64, 68, 283, 284, 291
COMMAND\label@refs	283
COMMAND\labelpstarttrue	19, 388, 444, 458
COMMAND\labelref@list	283, 286
COMMAND\language	187
COMMAND\lastbox	153
COMMAND\lastskip	128
COMMAND\latex@makecol	278
COMMAND\leavevmode	18, 153
COMMAND\led@check@nopb	375
COMMAND\led@check@pb	375
COMMAND\led@nopb	374, 375
COMMAND\led@nopbnum	374
COMMAND\led@pb	374, 375
COMMAND\led@pb@macro	374
COMMAND\led@pbnum	374
COMMAND\led@reinit@index@fornote	319
COMMAND\led@set@index@fornote	319
COMMAND\ledRflag	317
COMMAND\ledchapter	384, 444
COMMAND\ledfootinsdim	384
COMMAND\ledinnernote	64, 301, 446, 447, 455
COMMAND\ledinnote	314, 383, 389, 450
COMMAND\ledinnotemark	383, 389, 450, 460
COMMAND\ledleftnote	65, 301
COMMAND\ledlinenum	102, 443
COMMAND\ledllfill	159
COMMAND\ledlsnotefontsetup	454
COMMAND\ledlsnotesep	65

COMMAND\ledlsnotewidth	65
COMMAND\lednopb	73, 374
COMMAND\lednopbinverse	375
COMMAND\lednopbinversetrue	57, 73
COMMAND\lednopbnum	374
COMMAND\ledouternote	64, 301, 446, 447, 455
COMMAND\ledpb	73, 374, 384
COMMAND\ledpbnum	374
COMMAND\ledpbsetting	73, 374, 452
COMMAND\ledrightnote	65, 301
COMMAND\ledrsnotefontsetup	454
COMMAND\ledrsnotesep	65
COMMAND\ledrsnotewidth	65
COMMAND\ledsection	384
COMMAND\ledsectnomark	363
COMMAND\ledsectnotoc	362
COMMAND\ledsetnormalparstuff	383, 384, 450
COMMAND\ledsetnormalparstuff@common	233
COMMAND\ledsetnormalparstuffX	383, 384, 450
COMMAND\ledsidenote	64, 65, 301, 303
COMMAND\ledsidenote@	301
COMMAND\ledsubsection	384
COMMAND\ledsubsubsection	384
COMMAND\ledxxx	446
COMMAND\left	70
COMMAND\leftctab	356
COMMAND\leftheadline	102
COMMAND\leftlinenum	22, 102, 441, 443
COMMAND\leftlinenumannotation	103
COMMAND\leftltab	356
COMMAND\leftnoteupfalse	65
COMMAND\leftpstartnum	170
COMMAND\leftrtab	356
COMMAND\leftsidenote	303
COMMAND\leftskip	183, 186, 187
COMMAND\lemma	3, 28, 31, 33–35, 131, 134–136, 139, 140, 142, 380, 443, 444, 451, 453, 461
COMMAND\lemmaseparator	385
COMMAND\let	31, 54, 132, 138, 260, 278, 328, 441
COMMAND\letboxXendendlinenum	385
COMMAND\line	191, 194
COMMAND\line@list	105, 107, 121, 138
COMMAND\line@list@stuff	92, 108, 123, 441, 443
COMMAND\line@list@version	110
COMMAND\line@margin	99, 167, 299
COMMAND\line@num	104–106, 166, 441
COMMAND\line@set	140, 141
COMMAND\lineannot	25, 61, 141
COMMAND\lineannot@set	141
COMMAND\lineation	21, 98, 458, 460
COMMAND\linebreak	41
COMMAND\lineation	454
COMMAND\linenum	25, 28, 31, 60–62, 131, 140, 141, 289, 291, 298, 380
COMMAND\linenum@out	123, 283, 286
COMMAND\linenumannotation	24, 163, 458, 459



COMMAND\linenumannotationothersidefalse	26
COMMAND\linenumannotationothersidetrue	26, 461
COMMAND\linenumberlist	21, 89, 166, 167, 441
COMMAND\linenumberstyle	23, 102, 441, 456
COMMAND\linenumincrement	21, 443
COMMAND\linenummargin	21, 22, 99, 299
COMMAND\linenumr@p	102, 441, 443
COMMAND\linenumrep	102, 443
COMMAND\linenumsep	22, 65, 102, 300
COMMAND\linrangesep@	274
COMMAND\lineref	283, 289, 298, 386, 448
COMMAND\list@clear	96
COMMAND\list@clearing@reg	443
COMMAND\list@create	96
COMMAND\lock@disp	101
COMMAND\lock@off	118
COMMAND\lock@on	117
COMMAND\lockdisp	22, 101
COMMAND\loop	174, 324
COMMAND\ltab	356, 357, 361
COMMAND\ltabtext	361
COMMAND\m@mmf@prepare	215
COMMAND\mainmatter	458, 459
COMMAND\makeatletter	157
COMMAND\makehboxofhboxes	188, 189
COMMAND\makeindex	66, 316
COMMAND\makelabel	298
COMMAND\managestanza@modulo	325
COMMAND\marginpar	64, 74, 299, 442
COMMAND\marginparwidth	65, 300
COMMAND\markboth	27, 157
COMMAND\mathchardef	324
COMMAND\maxhXnotes	385
COMMAND\maxhnotesX	53, 75, 384, 445, 446, 449
COMMAND\maxlinesinpar@list	108
COMMAND\measurebody	358
COMMAND\measuretbody	358
COMMAND\memorybreak	20
COMMAND\metaXXXvfootnote	213
COMMAND\microtype	154
COMMAND\morenoexpands	75, 76, 131, 132
COMMAND\morethantwolines	386
COMMAND\mpfnpos	40, 235, 446, 456
COMMAND\mpnormalfootgroup	441
COMMAND\mpnormalvfootnote	441
COMMAND\msdata	37, 331–333, 456, 457, 459
COMMAND\msdataposition	38
COMMAND\multfootsep	38, 214
COMMAND\multiplefootnotemarker	214
COMMAND\musixtex	446
COMMAND\n@num	443, 450
COMMAND\n@num@ref	450
COMMAND\new@line	125, 442
COMMAND\new@series	132

COMMAND\newcommand	31, 77, 94, 214, 285
COMMAND\newcommandx	31
COMMAND\newhookarg@specific	273
COMMAND\newhookcommand@series	267, 268, 450, 460
COMMAND\newhookcommand@series@reload	268
COMMAND\newhookcommand@toggle@reload	268, 447
COMMAND\newhooktoggle@series	268, 450
COMMAND\newhooktoggle@specific	273
COMMAND\newif	449
COMMAND\newline	41
COMMAND\newlinechar	261
COMMAND\newseries	39, 381, 384
COMMAND\newseries@	252, 265
COMMAND\newverse	57, 59, 384, 447, 457
COMMAND\next	324
COMMAND\next@action	109
COMMAND\next@actionline	109
COMMAND\next@insert	171
COMMAND\next@line@list@stuff	124, 458
COMMAND\nl@regR	111
COMMAND\no@expands	75, 132, 139, 260, 322, 441
COMMAND\noXlemmaseparator	386
COMMAND\nobreak	204
COMMAND\nocritical	252
COMMAND\noeledsec	384
COMMAND\noendnotes	384
COMMAND\noexpand	381
COMMAND\nofamiliar	270
COMMAND\noindent	18, 19, 57, 153, 452
COMMAND\noindentX	452
COMMAND\nomk@	275
COMMAND\nonbreakableafternumber	386
COMMAND\nonoteX	41, 461
COMMAND\nonumberinfootnote	386
COMMAND\norelax	54
COMMAND\normal@footnotemarkX	218
COMMAND\normal@page@break	374
COMMAND\normal@pars	233
COMMAND\normalbfnoteX	443
COMMAND\normalbodyfootmarkX	218
COMMAND\normalfont	456
COMMAND\normalfootfmt	54, 182, 189, 199, 239, 455
COMMAND\normalfootfmtX	219
COMMAND\normalfootfootmarkX	219
COMMAND\normalfootgroup	184
COMMAND\normalfootgroupX	220
COMMAND\normalfootnoterule	179
COMMAND\normalfootstart	183, 186
COMMAND\normalfootstartX	220
COMMAND\normalvfootnote	181, 182
COMMAND\normalvfootnote@inserted	181, 182
COMMAND\normalvfootnoteX	218
COMMAND\notbool	363
COMMAND\notefontsetup	385

COMMAND\notefontsizeX	48, 385
COMMAND\notenumfont	384
COMMAND\notenumfontX	48, 384
COMMAND\notesXwidthliketwocolumns	386
COMMAND\noteswidthliketwocolumnsX	53, 386, 447, 449
COMMAND\num@lines	148, 173
COMMAND\numberlinefalse	20, 459
COMMAND\numberlinetrue	20, 459
COMMAND\numberonlyfirstinline	265, 386
COMMAND\numberonlyfirstintwoline	386
COMMAND\numberpstartfalse	19
COMMAND\numberpstarttrue	19, 44, 382, 443, 451, 456, 458
COMMAND\numberstanza	44
COMMAND\numberstanzafalse	58
COMMAND\numberstanzatrue	58
COMMAND\numlabfont	22, 54, 102
COMMAND\none@line	148
COMMAND\nonehalfspacing	452
COMMAND\nonlyXpstart	386
COMMAND\nonlyoneruleX	235, 462
COMMAND\nonlyoneruleX@	235
COMMAND\nonlysideX	259
COMMAND\npage@action	106, 116
COMMAND\npage@start	106, 443
COMMAND\npagecontents	106
COMMAND\npagenumbering	281, 462
COMMAND\npageparbreak	384
COMMAND\npageref	61, 289
COMMAND\npar	30, 41, 153, 233
COMMAND\npar@line	148, 173
COMMAND\npar@patch@pagenumbering	281
COMMAND\npar@patch@thepage	125, 281
COMMAND\npara@footgroup	186
COMMAND\npara@footgroupX	230
COMMAND\npara@footsetup	185, 441
COMMAND\npara@footsetupX	228, 441, 447
COMMAND\npara@vfootnoteX	229
COMMAND\nparafootfmt	188, 189, 455
COMMAND\nparafootfmtX	230
COMMAND\nparafootfm	190
COMMAND\nparafootfmX	231
COMMAND\nparafootfmsep	384
COMMAND\nparafootsep	386, 445, 450
COMMAND\nparafootsepX	51, 108, 384, 386, 389, 455, 460, 461
COMMAND\nparafootstart	186
COMMAND\nparafootstartX	229
COMMAND\nparavfootnote	186, 189
COMMAND\nparfillskip	179, 188, 462
COMMAND\nparindent	452
COMMAND\nparindentX	49, 388, 452, 454, 459
COMMAND\nparse@annot	165
COMMAND\nparshape	75
COMMAND\nparskip	18, 19, 57, 153
COMMAND\npause@numbering	94

COMMAND\pausenumbering	20, 94, 95, 109, 110, 154, 446, 447, 454
COMMAND\pausenumbering@page@num	94
COMMAND\penalty	188
COMMAND\pend	2, 7, 17–20, 22, 23, 72, 73, 129, 131, 134, 141, 148, 150–154, 170, 171, 330, 382, 445–447, 455–458
COMMAND\preXnotes	449, 455
COMMAND\prenotesX	52, 238, 449
COMMAND\prepare@Xgroupbyline	213, 214
COMMAND\prepare@Xprenotes	236
COMMAND\prev@nopb	374
COMMAND\prev@pb	374
COMMAND\prevlineX	107
COMMAND\prevpageX@num	108
COMMAND\print@Xfootnoterule	450
COMMAND\print@Xnotes	276, 277
COMMAND\print@Xnotes@forpages	449
COMMAND\print@eledsection	156
COMMAND\print@footnoteXrule	450
COMMAND\print@leftmargin@eledsection	364
COMMAND\print@lemma	200
COMMAND\print@line	155
COMMAND\print@notesX@forpages	449
COMMAND\print@rightmargin@eledsection	364
COMMAND\printendlines	244, 251, 296, 387, 441, 442
COMMAND\printlinefootnote	201, 202, 448
COMMAND\printlinefootnotearea	202, 204, 448
COMMAND\printlinefootnotenumbers	201
COMMAND\printlines	182, 199, 204, 205, 244, 296, 387, 441, 443, 449, 453
COMMAND\printnpnum	383, 387
COMMAND\printpstart	199
COMMAND\protect	132, 381
COMMAND\providecommand	214, 441
COMMAND\pstarL	148
COMMAND\pstart	2, 7, 17–20, 22, 23, 71–73, 116, 129, 134, 138, 141, 148, 151–153, 156, 171, 330, 368, 382, 443–447, 449–452, 454–458, 460
COMMAND\pstartinfootnote	386
COMMAND\pstartinfootnoteeverytime	386
COMMAND\pstartnum	170
COMMAND\pstartref	60, 283, 290, 446, 457
COMMAND\pstarts	443
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, 68
COMMAND\refformated@	296
COMMAND\refformatedwithpage	296
COMMAND\relax	19, 116, 138, 160, 171, 328, 346, 382
COMMAND\renewcommand	75, 384, 385, 387
COMMAND\reset@current@annot	165

COMMAND\reset@msd@options@iffullpage	339
COMMAND\resetlinenumannotation	25, 165
COMMAND\resetprevline@	107
COMMAND\resetprevpage@	108
COMMAND\resumenumbering	20, 90, 91, 95, 96, 109, 110, 154, 443, 446, 447, 454
COMMAND\right	70
COMMAND\rightctab	357
COMMAND\rightlinenum	22, 102, 441, 443
COMMAND\rightlinenumannotation	103
COMMAND\rightltab	357
COMMAND\rightnoteupfalse	65
COMMAND\rightrtab	357
COMMAND\rightsidenote	303
COMMAND\rightskip	183, 186–188
COMMAND\rightstartnum	170
COMMAND\rigidbalance	191, 192, 194, 388, 454
COMMAND\rigidbalanceX	191, 388, 454
COMMAND\robustify	42, 45
COMMAND\roman	350, 453
COMMAND\rtab	356–358, 361
COMMAND\rtabtext	358, 361
COMMAND\sameparallelpagenumber	125
COMMAND\sameword	16, 32–35, 122, 141–144, 146, 389, 448, 450–452, 455, 457, 458, 460
COMMAND\sameword@inedtext	143
COMMAND\saweword	142
COMMAND\scriptsize	102
COMMAND\section	71, 443, 456
COMMAND\section@num	91
COMMAND\sectionmark	363
COMMAND\select@lemmafnt	54, 176
COMMAND\series	252
COMMAND\series@	252
COMMAND\seriesatbegin	39, 264, 449
COMMAND\seriesatend	39, 264, 450
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, 63, 295, 453, 457
COMMAND\setSErefonlypageprefixsingle	62, 63, 295, 453, 457
COMMAND\setSErefprefixmore	62
COMMAND\setSErefprefixsingle	62
COMMAND\setapprefprefixmore	62, 384
COMMAND\setapprefprefixsingle	62, 64, 384, 453
COMMAND\setcommand@series	266
COMMAND\setcounter	376
COMMAND\sethangindentX	198
COMMAND\sethangingsymbol	56, 323, 384, 452
COMMAND\sethanginsymbol	55
COMMAND\setistwofollowinglines	209
COMMAND\setl@dlprbox	305
COMMAND\setline	23, 106, 111, 115, 129, 132, 452
COMMAND\setlinenum	23, 111, 116, 129, 151, 441, 458

COMMAND\setlinenumannotationsep	27, 389, 459
COMMAND\setmsdatalabel	38
COMMAND\setmsdataposition	457
COMMAND\setmsdataseries	37
COMMAND\setprintendlines	244, 248, 442
COMMAND\setprintlines	205, 209, 244, 443
COMMAND\setsidenotesep	65
COMMAND\setsidenotsep	385
COMMAND\setstanzaindent	325
COMMAND\setstanzaindents	56, 324, 382, 460
COMMAND\setstanzapenalties	324
COMMAND\setstanzavalues	324
COMMAND\settoggle@series	266, 444, 447
COMMAND\showlemma	132, 441, 442
COMMAND\showwordrank	35, 143
COMMAND\sidenote@margin	442
COMMAND\sidenotemargin	65, 442, 446
COMMAND\sidenotesep	385
COMMAND\sidepstartnumtrue	19
COMMAND\skip	183
COMMAND\skipnumbering	23, 118, 119, 130, 443, 450
COMMAND\skipnumbering@reg	450
COMMAND\small	48
COMMAND\somemacro	36
COMMAND\special	13
COMMAND\splitmaxdepth	178, 193
COMMAND\splitoff	191
COMMAND\splittopskip	178, 193, 194
COMMAND\stanza	22, 23, 57–59, 91, 328, 385, 387, 452, 457
COMMAND\stanza@hang	327
COMMAND\stanza@line	327
COMMAND\stanzaindent	56, 325, 448
COMMAND\stanzaindent*	56
COMMAND\stanzaindentbase	324
COMMAND\stanzanumwrapper	58
COMMAND\startlock	21, 22, 106, 129, 328
COMMAND\startstanzahook	385
COMMAND\startsub	22, 106, 128
COMMAND\stopmsd	334
COMMAND\stopmsdata	37, 332
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, 286
COMMAND\sub@on	114, 286
COMMAND\subline@num	104–106
COMMAND\sublinenum@rep	441
COMMAND\sublinenumberstyle	23, 102, 441
COMMAND\sublinenumincrement	21
COMMAND\sublinenumr@p	102, 441, 443
COMMAND\sublinenumrep	102, 443
COMMAND\sublineref	60, 283, 289

COMMAND\subsectionmark	363
COMMAND\sw@inthisdtext	134
COMMAND\sw@list@inedtext	139, 146
COMMAND\sw@txt	143
COMMAND\swnoexpands	35, 458
COMMAND\symlinenum	386
COMMAND\symplinum	385
COMMAND\sza@penalty	327
COMMAND>tag	448
COMMAND\text	380
COMMAND\text( <i>language</i> )	49
COMMAND\textcolor	76
COMMAND\textheight	75
COMMAND\textsuperscript	26, 89, 164
COMMAND\the	125, 441
COMMAND\thefootnoteA	38
COMMAND\thefootnoteX	445
COMMAND\thelabidx	318
COMMAND\thepage	111, 125, 280, 281
COMMAND\thepstart	19, 388
COMMAND\thepstartL	443
COMMAND\thepstartR	443
COMMAND\thestanza	58
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	194, 455
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	30, 242, 455
COMMAND\twocolfootfmt	455
COMMAND\twocolfootfmtX	225
COMMAND\twocolfootgroupX	225
COMMAND\twocolfootsetupX	224
COMMAND\twocolvfootnoteX	224
COMMAND\twolines	265, 386
COMMAND\twolines@A	265
COMMAND\twolines@B	265
COMMAND\twolines@C	265
COMMAND\twolinesbutnotmore	386
COMMAND\twolinesonlyinsamepage	386
COMMAND\txbeforeXnotes	386
COMMAND\txbeforenotesX	51, 175, 457
COMMAND\txbeforenotesonlyonceX	52, 457
COMMAND\uline	38, 39

COMMAND\unhbox	154, 187
COMMAND\unpenalty	188, 189
COMMAND\unskip	188
COMMAND\unvxh	188, 386
COMMAND\unvxhX	386
COMMAND\upbracefill	356
COMMAND\usingcritext	381, 384
COMMAND\usingedtext	381, 384
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, 442
COMMAND\vl@disnote	303
COMMAND\vl@dlsnote	303
COMMAND\vl@dosnote	303
COMMAND\vl@drsnote	303
COMMAND\vnumfootnoteX	443
COMMAND\vsizer	53, 75
COMMAND\vsplit	173
COMMAND\waklam	356
COMMAND\waklamec	356
COMMAND\wapunktel	356
COMMAND\wastricht	356
COMMAND\widthX	53, 388, 454
COMMAND\wrap@edcrossref	288, 447
COMMAND\wrapcontentX	49, 454
COMMAND\wrapped@bodyfootmarkX	232
COMMAND\wrapped@footfootmarkX	232
COMMAND\x...	60
COMMAND\xabslineref	289
COMMAND\xannotationref	60
COMMAND\xdef	96, 328
COMMAND\xflagref	61, 290, 387, 453
COMMAND\xleft@appenditem	97, 131
COMMAND\xlineref	60, 61, 387, 453
COMMAND\xpageref	60
COMMAND\xparse	36
COMMAND\xpstartref	60, 446, 458
COMMAND\xr	64
COMMAND\xright@appenditem	96, 97
COMMAND\xsublineref	60
COMMAND\xxref	61, 291, 298, 446, 449, 455
COMMAND\xxxmatter	281
ENVIRONMENTastanza	91, 453
ENVIRONMENTedarrayc	361
ENVIRONMENTedarrayl	361
ENVIRONMENTedarrayr	361
ENVIRONMENTedtabular	460, 462
ENVIRONMENTedtabularc	361
ENVIRONMENTedtabularl	361
ENVIRONMENTedtabularr	361
ENVIRONMENTledgroup	81, 310, 388, 453



ENVIRONMENTledgroupsized	310
PACKAGE(r)(e)ledmac	39
PACKAGEEDMAC	1, 6, 10, 12–14, 77, 204, 214, 283, 324, 380, 391, 441
PACKAGEEledmac	106, 315, 383, 384, 448, 450
PACKAGEEledpar	450
PACKAGEEtoolbox	78
PACKAGEParallel	391
PACKAGEReledmac	58, 386–388
PACKAGEamsgen	340
PACKAGEamsmath	340
PACKAGEbabel	49, 76, 350, 453
PACKAGEbiblatex	74
PACKAGEbidi	49, 50, 81, 452, 460
PACKAGEcalc	461
PACKAGEccaption	90
PACKAGEcolor	75
PACKAGECSquotes	457
PACKAGEedstanza	1, 13, 323
PACKAGEeledmac	1, 10, 14, 15, 67, 214, 311, 315, 343, 365, 377, 381, 383–385, 444, 446, 448
PACKAGEeledpar	179, 391, 443, 446–449
PACKAGEetex	452
PACKAGEetoolbox	96, 141, 252, 265, 275, 303, 363, 374
PACKAGEfancyhdr	278, 455
PACKAGEfloatrow	74, 388
PACKAGEfootmisc	38, 76, 81, 214, 391, 456
PACKAGEgeometry	15
PACKAGEglossaries	68, 320, 453, 461
PACKAGEglossaries-extra	68, 461
PACKAGEhandout	448
PACKAGEhyperref	61, 134, 137, 232, 284, 288, 317, 367, 376, 445–448, 454, 457, 459, 461
PACKAGEifluatex	78
PACKAGEifxetex	78
PACKAGEimakeidx	66, 74, 81, 311, 315, 384, 445, 446, 448
PACKAGEindextols	319
PACKAGEindextool	384
PACKAGEindextools	66, 74, 81, 91, 311, 315, 319, 384, 448, 453, 456
PACKAGEledarab	76
PACKAGEledmac	1, 10, 13, 14, 76, 96, 315, 380, 381, 384, 387
PACKAGEledpar	76
PACKAGEmemoir	81, 315, 384, 391, 447, 451
PACKAGEmorewrites	74
PACKAGEmusixtex	446
PACKAGEperpage	453
PACKAGEpolyglossia	47, 76, 177, 199, 453, 460, 461
PACKAGEragged2e	50, 78
PACKAGEReledmac	1, 6, 10–12, 14–17, 20, 21, 24, 27, 29, 31–33, 36, 38–40, 42–44, 46, 49, 50, 52–54, 56, 59, 60, 62, 64, 66–68, 72–78, 97, 99, 105, 106, 110, 114, 123–125, 132, 136, 138, 141, 171, 180, 183, 187, 199, 214, 239, 252, 257, 265, 275, 280, 288, 289, 292, 298, 315, 337, 340, 343, 363, 364, 375, 376, 383–390, 451, 454, 456
PACKAGEReledpar	1, 4, 6, 8, 16, 19, 20, 39, 53, 61, 63, 73, 74, 76, 78, 90, 97, 108, 114, 124, 125, 135, 138, 148, 180, 183, 233, 234, 252, 258, 275–277, 280, 311, 323, 363, 452, 453, 455–458, 461
PACKAGESuffix	78
PACKAGEtabmac	1, 13, 391
PACKAGEthalie	58

PACKAGEulem	39
PACKAGEuninormalize	32
PACKAGExargs	31, 78
PACKAGExkeyval	78, 274
PACKAGExparse	78, 458
PACKAGEXR	5, 64, 298, 299, 454
PACKAGExref	298
PACKAGExstring	78, 317

## A

\absline@num	<u>1</u>
Abu Kamil Shuja' b. Aslam	13
\actionlines@list	<u>1</u>
\actions@list	<u>1</u>
\add@hookarg@specific@to@cs	<u>1</u>
\add@hooktoggle@specific@to@cs	<u>1</u>
\add@inserts	<u>1</u>
\add@inserts@next	<u>1</u>
\add@msd@	<u>1</u>
\add@msdata	<u>1</u>
\add@msdata@firstlineofpage	<u>1</u>
\add@penalties	<u>1</u>
\add@Xgroupbyline	<u>1</u>
\addtol@denvbody	<u>1</u>
Adelard II	13
\advancelabel@refs	<u>1</u>
\advanceline	<u>1</u> , 23
\Aendnote	29
\affixline@num	<u>1</u>
\affixpstart@num	<u>1</u>
\affixside@note	<u>1</u>
\Afootnote	28
\afterantilabe	58
\afternoteX	51
\afterruleX	52
\ampersand	<u>1</u> , 59
\annot@list	<u>1</u>
\annotationref	<u>1</u> , 60
\antilabe	<u>1</u> , 58
\append@notesX	<u>1</u>
\append@Xnotes	<u>1</u>
\applabel	<u>1</u> , 62
\appref	<u>1</u> , 62
\apprefwithpage	<u>1</u> , 62
\arrangementX	<u>1</u> , 41
\arrangementX@normal	<u>1</u>
\arrangementX@paragraph	<u>1</u>
\arrangementX@threecol	<u>1</u>
\arrangementX@twocol	<u>1</u>
\at@every@pend	<u>1</u>
\AtEndEveryPend	<u>1</u> , 19
\AtEveryPend	<u>1</u> , 19
\AtEveryPstart	<u>1</u> , 19
\AtEveryStanza	<u>1</u> , 57

<code>\AtEveryStopStanza</code> .....	<a href="#">1</a> , <a href="#">57</a>
<code>\AtStartEveryPstart</code> .....	<a href="#">1</a> , <a href="#">19</a>
<code>\AtStartEveryStanza</code> .....	<a href="#">1</a> , <a href="#">57</a>
<code>\autopar</code> .....	<a href="#">1</a> , <a href="#">18</a>

## B

<code>\ballast</code> .....	<a href="#">75</a>
<code>\ballast@count</code> .....	<a href="#">1</a>
Beeton, Barbara Ann Neuhaus Friend .....	18
<code>\beforeantilabe</code> .....	<a href="#">58</a>
<code>\beforeeledchapter</code> .....	<a href="#">1</a>
<code>\BeforeEveryStopStanza</code> .....	<a href="#">1</a> , <a href="#">57</a>
<code>\beforeinsertingX</code> .....	<a href="#">50</a>
<code>\beforeinsertion@X</code> .....	<a href="#">1</a>
<code>\beforenotesX</code> .....	<a href="#">52</a>
<code>\beginnumbering</code> .....	<a href="#">1</a> , <a href="#">16</a>
<code>\Bendnote</code> .....	<a href="#">29</a>
<code>\Bfootnote</code> .....	<a href="#">28</a>
<code>\bhookgroupX</code> .....	<a href="#">52</a>
<code>\bhooknoteX</code> .....	<a href="#">50</a>
<code>\bodyfootmarkA</code> .....	<a href="#">38</a>
<code>\boxfootnotenumbers</code> .....	<a href="#">1</a>
Bredon, Simon .....	13
Breger, Herbert .....	13, <a href="#">343</a>
Brey, Gerhard .....	13
Busard, Hubert L. L. ....	13
<code>\bypage@false</code> .....	<a href="#">1</a>
<code>\bypage@true</code> .....	<a href="#">1</a>
<code>\bypstart@false</code> .....	<a href="#">1</a>
<code>\bypstart@true</code> .....	<a href="#">1</a>

## C

<code>\c@addcolcount</code> .....	<a href="#">1</a>
<code>\c@ballast</code> .....	<a href="#">1</a>
<code>\c@firstlinenum</code> .....	<a href="#">1</a>
<code>\c@firstsublinenum</code> .....	<a href="#">1</a>
<code>\c@labidx</code> .....	<a href="#">1</a>
<code>\c@linenumincrement</code> .....	<a href="#">1</a>
<code>\c@sublinenumincrement</code> .....	<a href="#">1</a>
<code>\Cendnote</code> .....	<a href="#">29</a>
<code>\Cfootnote</code> .....	<a href="#">28</a>
<code>\ch@ck@l@ck</code> .....	<a href="#">1</a>
<code>\ch@cksub@l@ck</code> .....	<a href="#">1</a>
<code>\chapter</code> .....	<a href="#">1</a>
<code>\check@pb@in@verse</code> .....	<a href="#">1</a>
Chester, Robert of .....	13
Claassens, Geert H. M. ....	13
<code>\colalignX</code> .....	<a href="#">50</a>
<code>\continuousnumberingwithcolumns@sync@linenumber@singletext</code> .....	<a href="#">1</a>
Copernicus, Nicolaus .....	13
<code>\critext</code> .....	<a href="#">380</a>
<code>\ctab</code> .....	<a href="#">1</a>
<code>\ctabtext</code> .....	<a href="#">1</a>
<code>\current@annot</code> .....	<a href="#">1</a>

## D

Dekker, Dirk-Jan	76
\Dendnote	29
\Dfootnote	28
\disable@edlabel	1
\disable@endnotes	1
\disable@familiarnotes	1
\disable@notes	1
\disable@sidenotes	1
\disablel@dtabfeet	1
\do@actions	1
\do@actions@fixedcode	1
\do@actions@next	1
\do@ballast	1
\do@feet@custom@order	1
\do@feetX	1
\do@insidelinehook	1
\do@line	1
\do@linehook	1
\do@lockoff	1
\do@lockoffL	1
\do@lockon	1
\do@lockonL	1
\do@Xfeet	1
\doedindexlabel	1
\doendnotes	1, 29
\doendnotesbysection	1, 29
\doinsidelinehook	1, 27
\doinsidethislinehook	1, 27
\dolinehook	1, 27
\dosplits	1
Downes, Michael	75, 186, 188
\dummy@edindex	1
\dummy@edtext	1
\dummy@edtext@showlemma	1
\dummy@ref	1

## E

\edaftertab	1, 1, 1, 70, 356
edarrayc (env.)	68
edarrayl (env.)	68
edarrayr (env.)	68
\edatleft	1, 70
\edatright	1, 70
\edbeforetab	1, 1, 1, 70, 356
\edfilldimen	1
\edfont@info	1
\edindex	1, 66
\edindexlab	1, 68
\edlabel	1, 60
\edlabelE	1, 62
\edlabelS	1, 62
\edlabelSE	1, 62
\edlineref	1, 60

<code>\edmakelabel</code>	1, 61
<code>\edpageref</code>	1, 60
<code>\edrowfill</code>	1, 69
<code>\EDTAB</code>	1
<code>\edtabcolsep</code>	1, 69
<code>\EDTABINDENT</code>	1
<code>\edtabindent</code>	1
<code>\EDTABtext</code>	1
<code>edtabularc (env.)</code>	68
<code>edtabularl (env.)</code>	68
<code>edtabularr (env.)</code>	68
<code>\EDTEXT</code>	1
<code>\edtext</code>	1, 28
<code>\edtext@notes</code>	1
<code>\edtext@notes@start</code>	1
<code>\edvertdots</code>	1, 71
<code>\edvertline</code>	1, 71
<code>\Eendnote</code>	29
<code>\Efootnote</code>	28
<code>\eled@chapter</code>	1
<code>\eled@section</code>	1
<code>\eled@sectioning@out</code>	1
<code>\eled@subsection</code>	1
<code>\eled@subsubsection</code>	1
<code>\eledchapter</code>	1
<code>\eledchapter*</code>	1
<code>\eledsection</code>	1
<code>\eledsection*</code>	1
<code>\eledsubsection</code>	1
<code>\eledsubsection*</code>	1
<code>\eledsubsubsection</code>	1
<code>\eledsubsubsection*</code>	1
<code>\enablel@dtabfeet</code>	1
<code>\end@lemmas</code>	1
<code>\endashchar</code>	1
<code>\endline@num</code>	1
<code>\endlock</code>	1, 22
<code>\endminipage</code>	1
<code>\endnumbering</code>	1, 16
<code>\endpage@num</code>	1
<code>\endprint</code>	1
<code>\endquotation</code>	1
<code>\endquote</code>	1
<code>\endsub</code>	1, 22
<code>\endsubline@num</code>	1
environments:	
<code>edarrayc</code>	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

minipage .....	59
Euclid .....	13
\extensionchars .....	<u>1</u> , 73

**F**

\f@x@l@cks .....	<u>1</u>
Fairbairns, Robin .....	38
\first@linenum@out@false .....	<u>1</u>
\first@linenum@out@true .....	<u>1</u>
\firstfootnoteruleX@printed .....	<u>1</u>
\firsthalf@antilabe@wd .....	<u>1</u>
\firstlinenum .....	<u>1</u> , 21
\firstseriesX@ .....	<u>1</u>
\firstsublinenum .....	<u>1</u> , 21
\firstXseries@ .....	<u>1</u>
\flag@end .....	<u>1</u>
\flag@end@later .....	<u>1</u>
\flag@end@RTL .....	<u>1</u>
\flag@start .....	<u>1</u>
\flag@start@later .....	<u>1</u>
\flag@start@RTL .....	<u>1</u>
\flagstanza .....	<u>1</u> , 59
\flush@notes .....	<u>1</u>
\fnpos .....	<u>1</u> , 39
Folkerts, Menso .....	13
\footfootmarkA .....	38
\footfudgefiddle .....	<u>1</u> , 75
\footnote .....	<u>1</u>
\footnoteA .....	38
\footnoteB .....	38
\footnoteC .....	38
\footnoteD .....	38
\footnoteE .....	38
\footnotelang@lua .....	<u>1</u>
\footnotelang@poly .....	<u>1</u>
\footnoteoptions@ .....	<u>1</u>
\footnoteXmark .....	38
\footnoteXtext .....	38
\footplitskips .....	<u>1</u>
\fullstop .....	<u>1</u>

**G**

Gädeke, Nora .....	13
\get@edindex@hyperref .....	<u>1</u>
\get@edindex@ledinnote@command .....	<u>1</u>
\get@firsthalf@antilabe@wd .....	<u>1</u>
\get@fnmark .....	<u>1</u>
\get@fnmarkX .....	<u>1</u>
\get@index@command .....	<u>1</u>
\get@linelistfile .....	<u>1</u>
\get@pstartcounter .....	<u>1</u>
\get@stanzacounter .....	<u>1</u>
\get@sw@txt .....	<u>1</u>
\get@thisfootnote .....	<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> .....	49
<code>\hidenumbering</code> .....	<u>1</u> , 23
<code>\hidenumberingonleftpage</code> .....	<u>1</u> , 23
<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> .....	51
<code>\hsizetwocolX</code> .....	51
<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>\ifeledmaccomat@</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>\ifistanza</code>	1
<code>\ifistanzaL</code>	1
<code>\ifistanzaR</code>	1
<code>\ifl@d@dash</code>	1
<code>\ifl@d@elin</code>	1
<code>\ifl@d@esl</code>	1
<code>\ifl@d@pnum</code>	1
<code>\ifl@d@ssub</code>	1
<code>\ifl@d@Xendlinenumber</code>	1
<code>\ifl@d@Xlinenumber</code>	1
<code>\ifl@d@Xmorethantwoline</code>	1
<code>\ifl@d@Xtwoline</code>	1
<code>\ifl@dend@X</code>	1
<code>\ifl@dhiddenumber</code>	1
<code>\ifl@dmemoir</code>	1
<code>\ifl@dpaging</code>	1
<code>\ifl@dpairing</code>	1
<code>\ifl@dprintingcolumns</code>	1
<code>\ifl@dprintingpages</code>	1
<code>\ifl@dskipnumber</code>	1
<code>\ifl@dskipversenumber</code>	1
<code>\ifl@dstartendok</code>	1
<code>\ifl@footmisc</code>	1
<code>\ifl@imakeidx</code>	1
<code>\ifl@indextools</code>	1
<code>\ifledfinal</code>	1, 73
<code>\ifledgroupnotesL@</code>	1
<code>\ifledgroupnotesR@</code>	1
<code>\iflednopbinverse</code>	1, 1
<code>\ifledRcol</code>	1
<code>\ifledRcol@</code>	1
<code>\iflinenumannotationotherside</code>	1
<code>\ifnocritical@</code>	1
<code>\ifnoend@</code>	1
<code>\ifnofamiliar@</code>	1
<code>\ifnoledgroup@</code>	1
<code>\ifnoquotation@</code>	1
<code>\ifnoresetlinenumannotation@</code>	1
<code>\ifnoteschanged@</code>	1
<code>\ifnumberedpar@</code>	1
<code>\ifnumbering</code>	1
<code>\ifnumberingR</code>	1
<code>\ifnumberline</code>	1
<code>\ifnumberstanza</code>	1
<code>\ifparapparatus@</code>	1
<code>\ifparledgroup</code>	1
<code>\ifprevpgnotnumbered</code>	1
<code>\ifpst@rtedL</code>	1
<code>\ifresumenummering@start</code>	1
<code>\ifsameparallelpagenumber</code>	1
<code>\ifseriesbefore</code>	1
<code>\ifsidepstartnum</code>	1
<code>\ifstopmsdata@inserted@</code>	1
<code>\ifsublines@</code>	1



<code>\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> .....	66
<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>\insertparafotsepX</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@ddodoreinxtrafeet</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@denvbody</code>	1
<code>\l@desnote</code>	1
<code>\l@dfambeginmini</code>	1
<code>\l@dfamendmini</code>	1
<code>\l@dfetbeginmini</code>	1
<code>\l@dfetendmini</code>	1
<code>\l@dgetline@margin</code>	1
<code>\l@dgetlock@disp</code>	1
<code>\l@dgetref@num</code>	1
<code>\l@dgetsidenote@margin</code>	1
<code>\l@dobbleoptarg</code>	1
<code>\l@disnote</code>	1
<code>\l@dlabel@parse</code>	1
<code>\l@dld@ta</code>	1, 1
<code>\l@dlp@rbox</code>	1
<code>\l@dlsn@te</code>	1
<code>\l@dlsnote</code>	1
<code>\l@dmake@labels</code>	1
<code>\l@dmodforedtext</code>	1
<code>\l@dnullfills</code>	1
<code>\l@dnumpstartsL</code>	1
<code>\l@doldold@footnotetext</code>	1
<code>\l@dp@rsefootspec</code>	1
<code>\l@dparsedendline</code>	1
<code>\l@dparsedendpage</code>	1
<code>\l@dparsedendsub</code>	1
<code>\l@dparsedstartline</code>	1
<code>\l@dparsedstartpage</code>	1
<code>\l@dparsedstartsub</code>	1
<code>\l@dparsefootspec</code>	1
<code>\l@dpush@begins</code>	1
<code>\l@drd@ta</code>	1, 1
<code>\l@dref@undefined</code>	1
<code>\l@drestorefills</code>	1
<code>\l@drestoreforedtext</code>	1
<code>\l@drp@rbox</code>	1
<code>\l@drsn@te</code>	1
<code>\l@drsnote</code>	1
<code>\l@dsetmaxcolwidth</code>	1
<code>\l@dskipnumberfalse</code>	1
<code>\l@dskipnumbertrue</code>	1
<code>\l@dtabaddcols</code>	1
<code>\l@dtabnoexpands</code>	1
<code>\l@dunboxmpfoot</code>	1
<code>\l@dunhbox@line</code>	1
<code>\l@dzeropenalties</code>	1
<code>\l@noexpands@edgl</code>	1
<code>\l@secondmandarg</code>	1
<code>\l@wrap@ifnotemptybox</code>	1
<code>\l@wrapcs@ifnotemptybox</code>	1
<code>\label</code>	61
<code>\labelstartfalse</code>	1
<code>\labelstarttrue</code>	1, 19
<code>\labelref@list</code>	1

<code>\labelrefsparseline</code>	1
<code>\labelrefsparsesubline</code>	1
<code>\last@page@num</code>	1
Lavagnino, John	12
<code>\led@check@nopb</code>	1
<code>\led@check@pb</code>	1
<code>\led@err@AutoparNotNumbered</code>	1
<code>\led@err@BadAction</code>	1
<code>\led@err@edtextoutsidepstart</code>	1
<code>\led@err@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@doclearpage</code>	1
<code>\led@error@fail@patch@iiiminipage</code>	1
<code>\led@error@fail@patch@makecol</code>	1
<code>\led@error@fail@patch@reinserts</code>	1
<code>\led@error@fail@patch@addtocounter</code>	1
<code>\led@error@fail@patch@endminipage</code>	1, 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>	1
<code>\led@warn@BadLineation</code>	1
<code>\led@warn@BadLinenummargin</code>	1
<code>\led@warn@BadLockdisp</code>	1
<code>\led@warn@BadSetline</code>	1
<code>\led@warn@BadSetlinenum</code>	1
<code>\led@warn@BadSidenotemargin</code>	1
<code>\led@warn@BadSublockdisp</code>	1
<code>\led@warn@DuplicateLabel</code>	1
<code>\led@warn@edinde@outsidenumbering</code>	1
<code>\led@warn@LineFileObsolete</code>	1
<code>\led@warn@NoFile</code>	1
<code>\led@warn@NoIndexFile</code>	1
<code>\led@warn@NoMarginpars</code>	1
<code>\led@warn@RefUndefined</code>	1, 1
<code>\led@warn@SeriesStillExist</code>	1
<code>\led@warning@hsizeX@deprecated</code>	1
<code>\led@warning@mldatawithoutstop</code>	1
<code>\led@warning@preXnotes@deprecated</code>	1
<code>\led@warning@Xhsize@deprecated</code>	1
<code>ledgroup (env.)</code>	59
<code>ledgroupsize (env.)</code>	59
<code>\ledinnernote</code>	1, 64
<code>\ledinnote</code>	1
<code>\ledinnotemark</code>	1
<code>\ledleftnote</code>	1, 64
<code>\ledlinenum</code>	1
<code>\ledllfill</code>	1
<code>\ledlsnotefontsetup</code>	1, 65
<code>\ledlsnotesep</code>	1, 65
<code>\ledlsnotewidth</code>	1, 65
<code>\lednopb</code>	1, 73
<code>\lednopbinversetrue</code>	73
<code>\lednopbnum</code>	1
<code>\ledouternote</code>	1, 64
<code>\ledpb</code>	1, 73
<code>\ledpbnum</code>	1
<code>\ledpbsetting</code>	1, 73
<code>\ledrightnote</code>	1, 64
<code>\ledrlfill</code>	1
<code>\ledrsnotefontsetup</code>	1, 65
<code>\ledrsnotesep</code>	1, 65
<code>\ledrsnotewidth</code>	1, 65
<code>\ledsectnomark</code>	1
<code>\ledsectnotoc</code>	1
<code>\ledsetnormalparstuff@common</code>	1
<code>\ledsetnormalparstuffX</code>	1
<code>\ledsidenote</code>	1, 64
<code>\ledsidenote@</code>	1
<code>\leftctab</code>	1

<code>\leftlinenum</code> .....	<a href="#">1</a> , <a href="#">22</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> .....	<a href="#">1</a>
<code>\leftrtab</code> .....	<a href="#">1</a>
Leibniz .....	13
<code>\lemma</code> .....	<a href="#">1</a> , <a href="#">30</a>
<code>\letsforverteilen</code> .....	<a href="#">1</a>
<code>\line@list</code> .....	<a href="#">1</a>
<code>\line@list@stuff</code> .....	<a href="#">1</a>
<code>\line@list@version</code> .....	<a href="#">1</a>
<code>\line@margin</code> .....	<a href="#">1</a>
<code>\line@num</code> .....	<a href="#">1</a>
<code>\line@set</code> .....	<a href="#">1</a>
<code>\lineannot</code> .....	<a href="#">1</a> , <a href="#">25</a>
<code>\lineannot@set</code> .....	<a href="#">1</a>
<code>\lineation</code> .....	<a href="#">1</a> , <a href="#">21</a>
<code>\linenum</code> .....	<a href="#">1</a> , <a href="#">31</a>
<code>\linenum@out</code> .....	<a href="#">1</a>
<code>\linenumannotation</code> .....	<a href="#">1</a> , <a href="#">24</a>
<code>\linenumannotationothersidettrue</code> .....	<a href="#">26</a>
<code>\linenumberlist</code> .....	<a href="#">1</a> , <a href="#">21</a>
<code>\linenumberstyle</code> .....	<a href="#">1</a> , <a href="#">23</a>
<code>\linenumincrement</code> .....	<a href="#">1</a> , <a href="#">21</a>
<code>\linenummargin</code> .....	<a href="#">1</a> , <a href="#">21</a>
<code>\linenumr@p</code> .....	<a href="#">1</a>
<code>\linenumrep</code> .....	<a href="#">1</a>
<code>\linenumsep</code> .....	<a href="#">1</a> , <a href="#">22</a>
<code>\linerangesep@</code> .....	<a href="#">1</a>
<code>\list@clear</code> .....	<a href="#">1</a>
<code>\list@clearing@reg</code> .....	<a href="#">1</a>
<code>\list@create</code> .....	<a href="#">1</a>
<code>\lock@disp</code> .....	<a href="#">1</a>
<code>\lock@off</code> .....	<a href="#">1</a>
<code>\lock@on</code> .....	<a href="#">1</a>
<code>\lockdisp</code> .....	<a href="#">1</a> , <a href="#">22</a>
Lorch, Richard .....	13
<code>\ltab</code> .....	<a href="#">1</a>
<code>\ltabtext</code> .....	<a href="#">1</a>
Luecking, Dan .....	80

## M

<code>\m@mmf@check</code> .....	<a href="#">1</a>
<code>\m@mmf@prepare</code> .....	<a href="#">1</a>
<code>\M@sect</code> .....	<a href="#">1</a>
<code>\makehboxofhboxes</code> .....	<a href="#">1</a>
<code>\managestanza@modulo</code> .....	<a href="#">1</a>
<code>\maxhnotesX</code> .....	<a href="#">53</a>
Mayer, Gyula .....	13
<code>\measurebody</code> .....	<a href="#">1</a>
<code>\measurecell</code> .....	<a href="#">1</a>
<code>\measuremrow</code> .....	<a href="#">1</a>
<code>\measuretbody</code> .....	<a href="#">1</a>

<code>\measuretcell</code>	1
<code>\measuretrow</code>	1
Middleton, Thomas	13, 104
<code>minipage</code> (env.)	59
Mittelbach, Frank	12, 13
<code>\morenoexpands</code>	1, 75
<code>\mp@append@notesX</code>	1
<code>\mp@append@Xnotes</code>	1
<code>\mpfnpos</code>	1, 39
<code>\mpnormalfootgroup</code>	1
<code>\mpnormalfootgroupX</code>	1
<code>\mpnormalvfootnote</code>	1
<code>\mpnormalvfootnote@inserted</code>	1
<code>\mpnormalvfootnoteX</code>	1
<code>\mppara@footgroupX</code>	1
<code>\mppara@vfootnoteX</code>	1
<code>\mpparafootgroup</code>	1
<code>\mpparavfootnote</code>	1
<code>\mpthreecolfootgroup</code>	1
<code>\mpthreecolfootgroupX</code>	1
<code>\mpthreecolfootsetup</code>	1
<code>\mpthreecolfootsetupX</code>	1
<code>\mptwocolfootgroup</code>	1
<code>\mptwocolfootgroupX</code>	1
<code>\mptwocolfootsetup</code>	1
<code>\mptwocolfootsetupX</code>	1
<code>\msdata</code>	1, 37
<code>\msdata@c</code>	1
<code>\msdata@cR</code>	1
<code>\multfootsep</code>	1, 38
<code>\multiplefootnotemark</code>	1

## N

<code>\n@num</code>	1
<code>\n@num@stanza</code>	1
<code>\new@line</code>	1
<code>\newhookarg@specific</code>	1
<code>\newhookcommand@series</code>	1
<code>\newhookcommand@series@reload</code>	1
<code>\newhooktoggle@series</code>	1
<code>\newhooktoggle@series@reload</code>	1
<code>\newhooktoggle@specific</code>	1
<code>\newseries@</code>	1
<code>\newverse</code>	1
<code>\NEXT</code>	1, 1
<code>\next@line@list@stuff</code>	1
<code>\no@expands</code>	1
<code>\nomk@</code>	1
<code>\nonoteX</code>	41
<code>\normal@footnotemarkX</code>	1
<code>\normal@page@break</code>	1
<code>\normal@pars</code>	1
<code>\normalbfnoteX</code>	1
<code>\normalbodyfootmarkX</code>	1

<code>\normalfootfmt</code>	1
<code>\normalfootfmtX</code>	1
<code>\normalfootfootmarkX</code>	1
<code>\normalfootgroup</code>	1
<code>\normalfootgroupX</code>	1
<code>\normalfootnoterule</code>	1
<code>\normalfootnoteruleX</code>	1
<code>\normalfootstart</code>	1
<code>\normalfootstartX</code>	1
<code>\normalvfootnote</code>	1
<code>\normalvfootnote@inserted</code>	1
<code>\normalvfootnoteX</code>	1
<code>\notefontsizeX</code>	48
<code>\notenumfontX</code>	48
<code>\noteschanged@false</code>	1
<code>\noteschanged@true</code>	1
<code>\noteswidthliketwocolumnsX</code>	53
<code>\nulledindex</code>	1
<code>\nullsetzen</code>	1
<code>\num@lines</code>	1
<code>\numberedpar@false</code>	1
<code>\numberedpar@true</code>	1
<code>\numberingfalse</code>	1
<code>\numberingtrue</code>	1
<code>\numberlinefalse</code>	20
<code>\numberlinetrue</code>	20
<code>\numberpstartfalse</code>	1, 19
<code>\numberpstarttrue</code>	1, 19
<code>\numberstanzafalse</code>	58
<code>\numberstanzatrue</code>	58
<code>\numlabfont</code>	1, 54

## O

<code>\old@hsize</code>	1
<code>\one@line</code>	1
<code>\onlyoneruleX</code>	1, 52
<code>\onlyoneruleX@</code>	1
<code>optionafter</code>	26
<code>optionantilabe</code>	58, 330
<code>optionauxdir</code>	15, 455
<code>optionbefore</code>	26
<code>optioncontinuousnumberingwithcolumns</code>	90, 156, 162, 454, 458, 461
<code>optiondraft</code>	73, 74
<code>optionfinal</code>	73
<code>optioninnnote</code>	453
<code>optioninnote</code>	453
<code>optionlinerrangesep</code>	274
<code>optionminus</code>	454
<code>optionnocritical</code>	451, 453
<code>optionnoeledsec</code>	72, 368, 455
<code>optionnoend</code>	453
<code>optionnofamiliar</code>	451, 459
<code>optionnopenalties</code>	75, 455
<code>optionnoprefix</code>	64, 459

optionnoresetlinenumannotation	24, 44, 459
optionnotenumber	453, 456
optiononlyonerule	52, 462
optionparapparatus	461
optionplus	454
optionprefixmore	64, 459
optionsameparallelpagenumber	458
optionswcaseinsensitive	32, 457

## P

<code>\page@action</code>	1
<code>\page@num</code>	1
<code>\pagelinesep</code>	1, 67
<code>\pageref</code>	61
<code>\par@line</code>	1
<code>\par@patch@pagenumbering</code>	1
<code>\par@patch@thepage</code>	1
<code>\para@footgroupX</code>	1
<code>\para@footsetup</code>	1
<code>\para@footsetupX</code>	1
<code>\para@vfootnoteX</code>	1
<code>\parafootfmt</code>	1
<code>\parafootfmtX</code>	1
<code>\parafootgroup</code>	1
<code>\parafootsepX</code>	51
<code>\parafootstart</code>	1
<code>\parafootstartX</code>	1
<code>\paravfootnote</code>	1
<code>\parindentX</code>	49
<code>\parse@annot</code>	1
<code>\pausenumbering</code>	1, 20
<code>\pausenumbering@page@num</code>	1
<code>\pend</code>	1, 17
Plato of Tivoli	13
<code>\postbodyfootmark</code>	1
<code>\prebodyfootmark</code>	1
<code>\prenotesX</code>	52
<code>\prepare@edindex@fornote</code>	1
<code>\prepare@prenotesX</code>	1
<code>\prepare@Xgroupbyline</code>	1
<code>\prepare@Xprenotes</code>	1
<code>\prev@nopb</code>	1
<code>\prev@pb</code>	1
<code>\prevpage@num</code>	1
<code>\print@eledsection</code>	1
<code>\print@footnoteXrule</code>	1
<code>\print@leftmargin@eledsection</code>	1
<code>\print@lemma</code>	1
<code>\print@line</code>	1
<code>\print@notesX</code>	1
<code>\print@rightmargin@eledsection</code>	1
<code>\print@Xfootnoterule</code>	1
<code>\print@Xnotes</code>	1
<code>\printendlines</code>	1



<code>\printlineendnote</code> .....	<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>60</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@annot</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>20</u>
<code>\rightctab</code> .....	<u>1</u>
<code>\rightlinenum</code> .....	<u>1</u> , <u>22</u>
<code>\rightlinenumannotation</code> .....	<u>1</u>
<code>\rightltab</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 .....	13
<code>\sameword</code> .....	<u>1</u> , 32
<code>\sameword@inedtext</code> .....	<u>1</u>
Schöpf, Rainer .....	13
<code>\section@num</code> .....	<u>1</u>
<code>\select@lemmafонт</code> .....	<u>1</u>
<code>\select@lemmafонт</code> .....	<u>1</u> , 54
<code>\SEref</code> .....	<u>1</u> , 62
<code>\SErefonlypage</code> .....	62
<code>\SErefwithpage</code> .....	<u>1</u> , <u>1</u> , 62
<code>\series</code> .....	<u>1</u>
<code>\seriesatbegin</code> .....	<u>1</u> , 39
<code>\seriesatend</code> .....	<u>1</u> , 39
<code>\set@continuousnumberingforL</code> .....	<u>1</u>
<code>\set@line</code> .....	<u>1</u>
<code>\set@line@action</code> .....	<u>1</u>
<code>\set@this@c@page</code> .....	<u>1</u>
<code>\set@txtbeforenotesX</code> .....	<u>1</u>
<code>\set@Xtxtbeforenotes</code> .....	<u>1</u>
<code>\setapprefprefixmore</code> .....	62
<code>\setapprefprefixsingle</code> .....	62
<code>\setcommand@series</code> .....	<u>1</u>
<code>\sethangindentX</code> .....	<u>1</u>
<code>\sethangingsymbol</code> .....	<u>1</u> , 56
<code>\setistwofollowinglines</code> .....	<u>1</u>
<code>\setl@dlp@rbox</code> .....	<u>1</u>
<code>\setl@drpr@rbox</code> .....	<u>1</u>
<code>\setline</code> .....	<u>1</u> , 23
<code>\setlinenum</code> .....	<u>1</u> , 23
<code>\setlinenumannotationsep</code> .....	<u>1</u> , 26
<code>\setmcellcenter</code> .....	<u>1</u>
<code>\setmcellleft</code> .....	<u>1</u>
<code>\setmcellright</code> .....	<u>1</u>
<code>\setmrowcenter</code> .....	<u>1</u>
<code>\setmrowleft</code> .....	<u>1</u>
<code>\setmrowright</code> .....	<u>1</u>
<code>\setmsdatalabel</code> .....	<u>1</u> , 38
<code>\setmsdataposition</code> .....	<u>1</u> , 38
<code>\setmsdataseries</code> .....	<u>1</u> , 37
<code>\setnoteswidthliketwocolumnsX@</code> .....	<u>1</u>
<code>\setnotesXpositionliketwocolumns@</code> .....	<u>1</u>
<code>\setparindentX</code> .....	<u>1</u>
<code>\setprintendlines</code> .....	<u>1</u>
<code>\setprintlines</code> .....	<u>1</u>
<code>\setSErefonlypageprefixmore</code> .....	62
<code>\setSErefonlypageprefixsingle</code> .....	62
<code>\setSErefprefixmore</code> .....	62
<code>\setSErefprefixsingle</code> .....	62

<code>\setsidenotesep</code>	65
<code>\setstanzaindents</code>	1, 55
<code>\setstanzapenalties</code>	1, 56
<code>\setstanzavalues</code>	1
<code>\settcclcenter</code>	1
<code>\settcclleft</code>	1
<code>\settcclright</code>	1
<code>\settoggle@series</code>	1
<code>\setthrowcenter</code>	1
<code>\setthrowleft</code>	1
<code>\setthrowright</code>	1
<code>\setXnotespositionliketwocolumns@</code>	1
<code>\setXnoteswidthliketwocolumns@</code>	1
<code>\showlemma</code>	1, 73
<code>\showwordrank</code>	1, 35
<code>\sidenote@margin</code>	1
<code>\sidenotemargin</code>	1, 64
<code>\sidepstartnumtrue</code>	19
<code>\skip@lockoff</code>	1
<code>\skipnumbering</code>	1, 23
<code>\splitoff</code>	1
<code>\spreadmath</code>	1, 69
<code>\spreadtext</code>	1, 69
<code>\stanza</code>	1, 54
<code>\stanza@count</code>	1
<code>\stanza@hang</code>	1
<code>\stanza@indent</code>	1
<code>\stanza@line</code>	1
<code>\stanzaindent</code>	1, 56
<code>\stanzaindent*</code>	1, 56
<code>\stanzaindentbase</code>	1, 55
<code>\stanzanumwrapper</code>	1, 58
<code>\startlock</code>	1, 22
<code>\startsub</code>	1, 22
<code>\stepl@dcolcount</code>	1
<code>\stopmsdata</code>	1, 37
<code>\store@cannot@to@absline</code>	1
<code>\strip@szacnt</code>	1
<code>\sub@action</code>	1
<code>\sub@lock</code>	1
<code>\sub@off</code>	1
<code>\sub@on</code>	1
<code>\subline@num</code>	1
<code>\sublinenumberstyle</code>	1, 23
<code>\sublinenumincrement</code>	1, 21
<code>\sublinenumr@p</code>	1
<code>\sublinenumrep</code>	1
<code>\sublineref</code>	1, 60
<code>\sublines@false</code>	1
<code>\sublines@true</code>	1
<code>\sublock@disp</code>	1
<code>\sublockdisp</code>	1
Sullivan, Wayne	13, 75, 89, 94, 187, 188, 283, 323
<code>\sw@noexpand</code>	1

<code>\swnoexpands</code> .....	35
<code>\sza@penalty</code> .....	1

## T

<code>\tabHilfbox</code> .....	1
<code>\tabhilfbox</code> .....	1
<code>\theadcolcount</code> .....	1
<code>\theadtext</code> .....	1
<code>\theendpageline</code> .....	1
<code>\tfootnoteA</code> .....	38
Theodosius .....	13
<code>\thepageline</code> .....	1
<code>\thepar@page</code> .....	1
<code>\thepstart</code> .....	1, 19
<code>\thestanza</code> .....	1, 58
<code>\thestartpageline</code> .....	1
<code>\this@line@list@version</code> .....	1
<code>\this@section@last@page@num</code> .....	1
<code>\this@section@page@num</code> .....	1
<code>\threecol@begin@insert</code> .....	1
<code>\threecolfootfmt</code> .....	1
<code>\threecolfootfmtX</code> .....	1
<code>\threecolfootgroup</code> .....	1
<code>\threecolfootgroupX</code> .....	1
<code>\threecolfootsetup</code> .....	1
<code>\threecolfootsetupX</code> .....	1
<code>\threecolvfootnote</code> .....	1
<code>\threecolvfootnote@inserted</code> .....	1
<code>\threecolvfootnoteX</code> .....	1
<code>\toendnotes</code> .....	1, 30
<code>\toendnotes*</code> .....	1
<code>\twocolfootfmt</code> .....	1
<code>\twocolfootfmtX</code> .....	1
<code>\twocolfootgroup</code> .....	1
<code>\twocolfootgroupX</code> .....	1
<code>\twocolfootsetup</code> .....	1
<code>\twocolfootsetupX</code> .....	1
<code>\twocolvfootnote</code> .....	1
<code>\twocolvfootnote@inserted</code> .....	1
<code>\twocolvfootnoteX</code> .....	1
<code>\txtbeforenotesonlyonceX</code> .....	52
<code>\txtbeforenotesX</code> .....	51

## U

<code>\unvxhX</code> .....	1
----------------------------	---

## V

Vamana .....	13
<code>\variab</code> .....	1
<code>\vbfnoteX</code> .....	1
<code>\vl@dbfnote</code> .....	1, 1
<code>\vl@dcsnote</code> .....	1
<code>\vl@disnote</code> .....	1
<code>\vl@dlsnote</code> .....	1

<code>\vl@dosnote</code> .....	<a href="#">1</a>
<code>\vl@drsnote</code> .....	<a href="#">1</a>
<code>\vnumfootnoteX</code> .....	<a href="#">1</a>

## W

Whitney, Ron .....	13
<code>\widthX</code> .....	<a href="#">53</a>
<code>\wrap@edcrossref</code> .....	<a href="#">1</a>
<code>\wrapcontentX</code> .....	<a href="#">49</a>
<code>\wrapped@bodyfootmarkX</code> .....	<a href="#">1</a>
<code>\wrapped@footfootmarkX</code> .....	<a href="#">1</a>
<code>\write@lemma</code> .....	<a href="#">1</a>
Wujastyk, Dominik .....	12

## X

<code>\X@atbegininsertion</code> .....	<a href="#">1</a>
<code>\X@beforeinsertion</code> .....	<a href="#">1</a>
<code>\X@doreinfeet</code> .....	<a href="#">1</a>
<code>\xabslineref</code> .....	<a href="#">1</a>
<code>\Xafterlemmaseparator</code> .....	<a href="#">47</a>
<code>\Xafternote</code> .....	<a href="#">51</a>
<code>\Xafternumber</code> .....	<a href="#">45</a>
<code>\Xafterrule</code> .....	<a href="#">52</a>
<code>\Xaftersymlinenum</code> .....	<a href="#">46</a>
<code>\xannotationref</code> .....	<a href="#">1</a> , <a href="#">60</a>
<code>\Xarrangement</code> .....	<a href="#">1</a> , <a href="#">41</a>
<code>\Xarrangement@normal</code> .....	<a href="#">1</a>
<code>\Xarrangement@paragraph</code> .....	<a href="#">1</a>
<code>\Xarrangement@threecol</code> .....	<a href="#">1</a>
<code>\Xarrangement@twocol</code> .....	<a href="#">1</a>
<code>\Xbeforeinserting</code> .....	<a href="#">50</a>
<code>\Xbeforelemmaseparator</code> .....	<a href="#">47</a>
<code>\Xbeforenotes</code> .....	<a href="#">52</a>
<code>\Xbeforenumber</code> .....	<a href="#">45</a>
<code>\Xbeforepagenumber</code> .....	<a href="#">42</a>
<code>\Xbeforesymlinenum</code> .....	<a href="#">46</a>
<code>\Xbhookgroup</code> .....	<a href="#">52</a>
<code>\Xbhooknote</code> .....	<a href="#">50</a>
<code>\Xboxlinenum</code> .....	<a href="#">46</a>
<code>\Xboxlinenumalign</code> .....	<a href="#">46</a>
<code>\Xboxsymlinenum</code> .....	<a href="#">46</a>
<code>\Xcolalign</code> .....	<a href="#">50</a>
<code>\xedindex</code> .....	<a href="#">1</a>
<code>\xedtext</code> .....	<a href="#">1</a>
<code>\Xendafterlemmaseparator</code> .....	<a href="#">48</a>
<code>\Xendafternote</code> .....	<a href="#">54</a>
<code>\Xendafterpagenumber</code> .....	<a href="#">43</a>
<code>\Xendaftersymlinenum</code> .....	<a href="#">46</a>
<code>\Xendaftrenumber</code> .....	<a href="#">45</a>
<code>\Xendahookinplaceofnumber</code> .....	<a href="#">49</a>
<code>\Xendahooklinenumber</code> .....	<a href="#">49</a>
<code>\Xendbeforelemmaseparator</code> .....	<a href="#">47</a>
<code>\Xendbeforenumber</code> .....	<a href="#">45</a>
<code>\Xendbeforepagenumber</code> .....	<a href="#">43</a>

<code>\Xendbeforesymlinenum</code> .....	46
<code>\Xendbhookinplaceofnumber</code> .....	49
<code>\Xendbhooklinenum</code> .....	49
<code>\Xendbhooknote</code> .....	50
<code>\Xendboxendlinum</code> .....	47
<code>\Xendboxlinenum</code> .....	47
<code>\Xendboxlinenumalign</code> .....	47
<code>\Xendboxstartline</code> .....	47
<code>\Xendboxsymlinenum</code> .....	46
<code>\Xendhangindent</code> .....	49
<code>\Xendinplaceoflemmaseparator</code> .....	48
<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> .....	43
<code>\Xendlineprefixsingle</code> .....	43
<code>\Xendlinerangeseparator</code> .....	43
<code>\Xendmorethantwolines</code> .....	43
<code>\Xendnoidenticallylinenumannotation</code> .....	26
<code>\Xendnolinenumberifannotation</code> .....	45
<code>\Xendnonumber</code> .....	44
<code>\Xendnotefontsize</code> .....	48
<code>\Xendnotenumfont</code> .....	48
<code>\Xendnumberonlyfirstinline</code> .....	42
<code>\Xendnumberonlyfirstintwolines</code> .....	42
<code>\Xendpagenumberonlyfirst</code> .....	42
<code>\Xendparagraph</code> .....	54
<code>\Xendsep</code> .....	54
<code>\Xendstorelineinfo</code> .....	1
<code>\Xendsublinesep</code> .....	45
<code>\Xendsymlinenum</code> .....	42
<code>\Xendsymlinenumannotation</code> .....	45
<code>\Xendsympagenum</code> .....	42
<code>\Xendtwolines</code> .....	43
<code>\Xendtwolinesbutnotmore</code> .....	43
<code>\Xendtxtbeforenotes</code> .....	51
<code>\Xendwrapcontent</code> .....	49
<code>\Xfirstfootnoterule@printed</code> .....	1
<code>\xflagref</code> .....	1
<code>\Xgroupbyline</code> .....	51
<code>\Xgroupbylineseparetwolines</code> .....	51
<code>\Xhangindent</code> .....	49
<code>\Xsizethreecol</code> .....	50
<code>\Xsizetwocol</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> .....	<u>1</u>
<code>\Xlemmadisablefontselection</code> .....	48
<code>\Xlemmafont</code> .....	48
<code>\Xlemmaseparator</code> .....	47
<code>\Xlinenumannotationonlyfirst</code> .....	44
<code>\Xlinenumannotationonlyfirstintwo</code> .....	45
<code>\Xlinenumannotationposition</code> .....	25
<code>\Xlinenumannotationposition@side</code> .....	<u>1</u>
<code>\Xlinenumannotationpositionside</code> .....	25
<code>\Xlinrangeseparator</code> .....	43
<code>\xlineref</code> .....	<u>1</u> , 60
<code>\Xmaxhnotes</code> .....	53
<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> .....	44
<code>\Xnotboxingsubline</code> .....	47
<code>\Xnotefontsize</code> .....	48
<code>\Xnotenumentfont</code> .....	48
<code>\Xnoteswidthliketwocolumns</code> .....	53
<code>\Xnumberonlyfirstinline</code> .....	41
<code>\Xnumberonlyfirstintwolines</code> .....	41
<code>\Xonlyonerule</code> .....	<u>1</u> , 52
<code>\Xonlyonerule@</code> .....	<u>1</u>
<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>\Xsublinesepside</code>	22
<code>\Xsymlinenum</code>	42
<code>\Xsymlinenumannotation</code>	45
<code>\Xtoendnotes</code>	30
<code>\Xtwolines</code>	43
<code>\Xtwolinesonlyinsamepage</code>	43
<code>\Xtxtbeforenotes</code>	51
<code>\Xtxtbeforenotesonlyonce</code>	52
<code>\Xtxtbeforenumber</code>	42
<code>\Xunvxh</code>	1
<code>\Xwidth</code>	53
<code>\Xwrapcontent</code>	49
<code>\Xwrapendlemma</code>	49
<code>\Xwraplemma</code>	49
<code>\Xwraplinenumannotation@ref</code>	1
<code>\Xwraplinenumannotation@side</code>	1
<code>\xxref</code>	1, 61

**Z**

<code>\zz@@@</code>	1
<code>\zz@@@</code>	1



## 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	.....	278
\@lab: Removed page setting from \@lab	.....	286
\edlabel: Tweaked \edlabel to get correct page numbers	.....	283
\l@ddodoreintrafeet: Renamed \dodoreintrafeet to \l@ddodoreintrafeet	.....	277
\l@ddoxtrafeet: Renamed \doxtrafeet to \l@ddoxtrafeet	.....	275
\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	.....	286
\@nl@reg: Added a bunch of code to \@nl for handling \setlinenum	.....	111
\ledlinenum: Added \linenumr@p and \sublinenum@rep to \leftlinenum and \rightlinenum	.....	102
\linenumberlist: Added \linenumberlist mechanism	.....	89
\printendlines: Added \linenumr@p and \sublinenumr@p to \printendlines	.....	248
\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	.....	310
Added ledgroupsize environment	.....	310
Added minipage, etc., support	.....	1
\@iiiminipage: Modified kernel \@iiiminipage and \endminipage to cater for critical footnotes	.....	309
\edtext: Added \showlemma to \edtext	.....	134
\l@dfeetendmini: Added \l@dfeetbeginmini, \l@dfeetendmini and all their supporting code	.....	307
\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 \doclearpage	279
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	276
\edindex: Let eledmac take advantage of memoir's indexing	315
\print@Xnotes: Added \op@Xfeet	276
v0.5.0.	
General: Added left and right sidenotes	299
Added sidenotes, familiar footnotes in numbered text	1
\@footnotetext: Enabled regular \footnote in numbered text	215
\@xympar: Eliminated \marginpar disturbance	299
v0.5.1.	
General: Added moveable sidenote	299
Fixed right line numbers killed in v0.5	1
Only change \hsize in ledgroupsize environment otherwise page number can be in wrong place	310
\affixline@num: Changed \affixline@num to cater for sidenotes	166
\l@dgetsidenote@margin: Added \sidenotemargin and \sidenote@margin	300
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 \v@l@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	244
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
\fo@x@l@cks: Added \ch@cksub@l@ck, \ch@ck@l@ck and \f@x@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	309
\l@dcsnotetext@r: Added \l@emptyd@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> . . . . .	300
<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. . . . .	325
v0.13.1.	
General: <code>\thepstartL</code> and <code>\thepstartR</code> use now <code>\bfseries</code> and not <code>\bf</code> , which is deprecated and makes conflicts with memoir class. . . . .	1
v0.14.0.	
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. . . . .	283
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. . . . .	323
\printlines@count: Line numbering can be reset at each pstart. . . . .	210
v0.17.0.	
\ifinserthangingsymbol: New new management of hangingsymbol insertion, preventing undesirable insertions. . . . .	323
v1.0.0.	
General: \lemma can contain commands. . . . .	30
Debug in lineation command . . . . .	21
New generic commands to customize footnote display. . . . .	40, 265
Options nonum and nosepp in \Xfootnote. . . . .	29
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 \labelstarttrue. . . . .	19
Add \Xnumberonlyfirstintwolines . . . . .	41
Add \Xpstart and \Xonlypstart . . . . .	44
New hook to add arbitrary code at the beginning of the notes . . . . .	50
New options for block of notes. . . . .	51
New package option: parapparatus. . . . .	1
New tools to change order of series . . . . .	264
Sectioning commands. . . . .	71
\settoggle@series: \settoggle@series switch the global value of the toggle, not only the local value. . . . .	266
\Xprenotes: New skip \Xprenotes@ . . . . .	237
v1.2.0.	
\endquote: Compatibility of \ledchapter with the memoir class. . . . .	361
\Xprenotes: Debug in familiar footnotes (bug introduced by v1.1). . . . .	237
v1.3.0.	
\endquote: <i>Quotation</i> and quote environment inside numbered sections. . . . .	361
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. . . . .	305
\endquote: New option <i>noquotation</i> . . . . .	361
\get@thisfootnote: Compatibility of standard footnotes with eledmac when these footnotes contain any commands. . . . .	216
\labelrefsparsesubline: Fix a bug with \edlabel. . . . .	285

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</code> + <code>eledpar</code> + <code>chapter</code> . . . . .	1
v1.4.6.		
	General: Bug with <code>memoir</code> class introduced by 1.4.5. . . . .	1
v1.4.7.		
	<code>\endquote</code> : Compatibility of sectioning commands with <code>\autopar</code> . . . . .	361
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. . . . .	311
	<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> . . . . .	315
v1.5.1.		
	<code>\managestanza@modulo</code> : Correct <code>stanzaindentsrepetition</code> counter . . . . .	325
	<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. . . . .	328
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. . . . .	323
	<code>\l@dunhbox@oline</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. . . . .	311
	<code>\endquote</code> : Correction of sectioning commands in parallel texts. . . . .	361
	<code>\get@index@command</code> : Debug <code>\get@index@command</code> and compatibility with <code>hyperref</code> package. . . . .	314
	<code>\newhookcommand@series@reload</code> : Debug <code>\beforenotesX</code> and <code>\maxhnotesX</code> which did not work. . . . .	268
	<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). . . . .	238
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 <code>eledpar</code> .	311
v1.9.0.	
<code>\l@ddoxtrafeet</code> : Add <code>\fnpos</code> to choice the order of footnotes.	275
<code>\l@dfeetendmini</code> : Add <code>\mpfnpos</code> to choice the order of footnotes in <code>minipage / ledgroup</code> .	307
v1.10.0.	
General: Add <code>\pstartref</code> and <code>\xpstartref</code> to refer to a <code>pstart</code> number (extension of <code>\edlabel</code> ).	1
<code>\endquote</code> : Correction of sectioning commands in parallel texts.	361
v1.10.1.	
General: Compatibility with <code>cleveref</code> .	1
v1.10.2.	
General: Compatibility of stanza with v1.8a of <code>babel-greek</code> .	1
v1.10.3.	
General: Debug of cross-referencing.	1
v1.10.4.	
General: Debug of critical notes in <code>edtabular</code> 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.	54
Add hyperlink to <code>crossref</code> (needs <code>hyperref</code> package).	59
Compatibility with <code>musixtex</code> .	1
Debug <code>eledmac</code> sectioning command after using <code>\resumenummering</code> .	1
Ensure that <code>imakeidx</code> is loaded <i>before</i> <code>eledmac</code> .	311
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.	315
<code>\endquote</code> : <code>\ledxxx</code> sectioning commands are deprecated and replaced by <code>\eledxxx</code> commands.	361
<code>\initnumbering@reg</code> : <code>\beginnumbering</code> is defined only on <code>eledmac</code> , not on <code>eledpar</code> .	92
<code>\l@dgetsidenote@margin</code> : <code>\sidenotemargin</code> is now directly defined in <code>eledmac</code> to be able to manage <code>eledpar</code> .	300
<code>\l@disnote</code> : <code>\l@dlsnote</code> , <code>\l@drsnote</code> and <code>\l@dcsnote</code> defined only one time, in <code>eledmac</code> , including needs for <code>eledpar</code> case.	301
<code>\l@dnumpstartsL</code> : Add <code>\ifledRcol@</code> for <code>eledpar</code> .	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>\maxnotesX</code> which did not work when called after <code>\footparagraphX</code> .	268

Debug <code>\Xbeforenotes</code> and <code>\Xmaxhnotes</code> which did not work when called after <code>\footparagraph</code> .	268
<code>\pend</code> : New optional argument for <code>\pend</code> , to execute code after it.	150
<code>\stanza</code> : <code>&amp;</code> can have an optional argument: content to be printed after.	328
<code>\Stanza</code> can have an optional argument: content to be printed before.	328
Add <code>\newverse</code> macro, <code>\falseverse</code> deprecated.	328
v1.12.1.	
<code>\wrap@edcrossref</code> : Fix spurious spaces.	288
v1.12.2.	
<code>\l@dunhbox@oline</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	317
<code>\@ssect</code> : Debug <code>\eledchapter</code> in parallel with <code>memoir</code>	365
<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@dgobbleoptarg</code> : <code>\l@dgobblearg</code> becomes <code>\l@dgobbelloptarg</code>	346
<code>\l@drestoreforedtext</code> : Debug optional arguments of <code>\Xfootnote</code> in tabular context	346
<code>\pausenumbering@page@num</code> : Debug <code>\resumenumbering</code>	95
v1.12.7.	
<code>\wrap@edcrossref</code> : <code>\wrap@edcrossref</code> is now robust	288
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 <code>endnote</code>	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>	268
<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.	266
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@dpaging</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. . . . .	238
Delete previous override of <code>\l@d@@wrindexhyp</code> at the beginning of a document	
when hyperref is not loaded. . . . .	319
Move gobbling command . . . . .	89
Provide <code>\@gobblefour</code> . . . . .	89
<code>\edindex</code> : Let eledmac take advantage of imakeidx even when memoir class is used . . . . .	315
v1.14.1.	
<code>\@sssect</code> : Debug sectioning commands when using both <code>handout</code> and <code>hyperref</code>	
package. . . . .	367
v1.14.2.	
<code>\@sssect</code> : Debug <code>\edtext</code> after starred sectioning commands when using memoir class. . . . .	365
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> . . . . .	19
New tools to prevent ambiguous references in lemma . . . . .	32
<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 eledmac, not in eledpar. 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> .	
Eledmac provides <code>\edlineref</code> instead. . . . .	289
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 $\TeX$ 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> . . . . .	317
<code>\edlabel</code> : <code>\edlabel</code> is now defined only one time for both eledmac and eledpar . . . . .	283
<code>\l@d@section</code> : Option <code>parapparatus</code> works for endnotes. . . . .	239
<code>\l@dnumpstartsL</code> : Add <code>\ifl@dprintingpages</code> and <code>\@dprintingcolumns</code> for	
eledpar . . . . .	90
<code>\print@line</code> : Compatibility with Lua $\TeX$ 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 (eledpar) . . . . .	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. . . . .	291
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<sup>®</sup>TeX</code> . . . . .	1
Fix issues with RTL text in notes when using <code>Lua<sup>®</sup>TeX</code> . . . . .	1
Options <code>fulllines</code> in <code>\Xfootnote</code> . . . . .	29
The <code>\newifs</code> are not followed by boolean values set to false, because it is the <code>TeX</code> 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. . . . .	328
<code>\xxref</code> : Debug <code>\xxref</code> when not loading <code>eledpar</code> (Fix a bug added in 1.19.0). . . . .	291
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 . . . . .	264
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>noeledsec</code> package option . . . . .	1
Debug <code>\beforenotesX</code> <code>\maxhnotesX</code> <code>\noteswidthliketwocolumnsX</code> and <code>\afterruleX</code> with footnotes set in two and three columns. . . . .	1
Fix a bug when a <code>\Xfootnote</code> follows a <code>\Xendnote</code> in the second argument of <code>\edtext</code> (bug added in <code>eledmac</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>oldprintnpnumspace</code> option allows to come back to previous setting	1
<code>parapparatus</code> option works now with familiar footnotes.	1
Provide <code>\@gobblefive</code>	89
<code>\@edindex@hyperref</code> : Look at the <code>hyperindex</code> option of <code>hyperref</code> before inserting <code>hyperref</code>	317
<code>firstfootnoteruleX@printed</code> : Code refactoring: the spaces after the footnote rules are directly managed in <code>\print@Xfootnoterule</code> and <code>\print@footnoterule</code>	235
<code>\l@d@section</code> : <code>\endnotes</code> take five arguments.	239
<code>\ledinnotemark</code> : Add <code>\ledinnotemark</code> .	314
<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.	267
<code>\newhooktoggle@series</code> : <code>\newhooktoggle@series</code> can take an optional argument.	268
<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.	314
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 inputenc 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.	32
Fix a bug with <code>\sameword</code> when the same lemma is used for multiple notes or for nested <code>\edtexts</code> .	32
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.	32
<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, \Xendmoreethantwolines, \Xendtwolinesbutnotmore and \Xendtwolinesonlyinsamepage when using \apprefwithpage.	295
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 nofamiliar 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	278
\doxtrafeeti becomes \Xdo@feet; \doxtrafeetii becomes \do@Xfeet;	
\@opxtrafeeti becomes \@opfeetX; \doreinxtrafeetii becomes	
\X@doreinfeet; \doreinxtrafeeti becomes \@doreinfeetX.	278
Add \Xendinplaceofnumber 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.	315
\@iiiminipage: Patch \@iiiminipage instead of redefining it.	309
\@xympar: Patching \@xympar instead of redefining it	299
\endminipage: Patch \endminipage instead of redefining it.	309
\initnumbering@quote: \initnumbering@sectcmd becomes	
\initnumbering@quote	361
\l@advance@parledgroup@beforenormalnotes: Some conde of \l@dumboxmpfoot moved to \l@advance@parledgroup@beforenormalnotes	309
\newseries@: One endnotes file by series.	261
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 tabular 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 <code>Lua<math>\TeX</math></code> .	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>\Xendsymlinenum</code> , <code>\Xendbeforenumber</code> , <code>\Xendafternumber</code> , <code>\Xendbeforesymlinenum</code> , <code>\Xendaftersymlinenum</code> , <code>\Xendboxsymlinenum</code> , <code>\Xendhangindent</code> , <code>\Xendbhooklinenumber</code> , <code>\Xendahooklinenumber</code> , <code>\Xendbhookinplaceofnumber</code> , <code>\Xendahookinplaceofnumber</code> .	1
v2.2.1.	
General: Compatibility with $\TeX$ 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>\Xlinarangeseperator</code> and <code>\Xendlinarangeseperator</code> . . . . .	43
Option <code>linarangesep</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 <code>\romannumeral</code> command, like Greek language. . . . .	1
Fix a bug with tabular environments when using <code>babel</code> or <code>polyglossia</code> languages that override <code>\romannumeral</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>linarangesep</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	62
<code>\edlabel</code> : Fix a bug when calling <code>\edlabel</code> in a footnotes of the right side . . . . .	283
<code>\l@d@section: \endnotes</code> take six arguments. . . . .	239
<code>\printlines@count: \printlines</code> takes an eighth argument: the line flag . . . . .	210
<code>\SErefwithpage: Debug \setapprefprefixsingle</code> . . . . .	295
<code>\xlineref: \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. . . . .	289
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 linenummer 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. . . . .	239
v2.7.1.	
General: Debug <code>\Xbhookgroup</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. . . . .	239
v2.8.1.	
General: Warnings for undefined labels are really parsable by <code>latexmk</code> . . . . .	1
v2.8.2.	
General: Fix a bug concerning indent in a paragraph immediately following a sectioning command (bug NOT fixed on <code>reledpar</code> ) . . . . .	1
Fix a bug with <code>\AtEveryPstart</code> added in version 2.0.0. . . . .	1
Fix a bug with vertical space after the between-sectioning command as optional argument of a <code>\pstart</code> and <code>\pstart</code> content . . . . .	1
v2.9.0.	
General: Allow continuing line numbering between normal text and parallel text, using <code>\pausenumbering</code> and <code>\resumenumbering</code> and the <code>continuousnumberingwithcolumns</code> option. . . . .	1
Fix a bug when using <code>\lineneation{page}</code> and <code>\pausenumbering... \resumenumbering</code> . . . . .	1
Fix a bug with three- and two-column footnote setting (added in v.2.4.0). . . . .	1
Fix spurious space inside three-column familiar footnote. . . . .	1
Write correct metadata in numbered files when using <code>\pausenumbering... \resumenumbering</code> . . . . .	1
v2.9.1.	
General: Fix a bug when notes start with plus or minus. . . . .	1
v2.9.2.	
General: Fix a bug with <code>hyperref</code> package when a lemma starts with plus or “minus” (bug introduced in v. 2.7.0). . . . .	1
v2.9.3.	
General: Fix a bug with line number position and reset added by v. 2.9.0 . . . . .	1
v2.10.0.	
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> . . . . .	49

\print@lemma: Code refactoring between \parafootfmt, \twocolfootfmt, \threecolfootfmt and \normalfootfmt. . . . .	200
v2.10.1.	
General: Add nopenalties option. . . . .	1
Fix a bug introduced in v. 1.4: not paragraphed critical footnotes could prevent marginal line number from being displayed . . . . .	1
v2.11.0.	
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 noeledsec option. . . . .	1
Fix a bug with \ledinnernote and \ledouternote in parallel typesetting . . . . .	1
Fix a bug with familiar footnote number in optional argument of \pstart or \pend in parallel typesetting . . . . .	1
Fix spurious vertical space in \chapter when used as optional argument of \pstart in parallel typesetting. . . . .	1
Make endnote compatible with \sameword mechanism . . . . .	1
More accurate message to control the position of \Xfootnote and \applabel in the L <sup>A</sup> T <sub>E</sub> X 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>	278
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>notenumber</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.	278
New options for <code>\fnpos</code> and <code>\mpfnpos</code> to set a customized order for familiar and critical footnotes.	1
When <code>\edindex</code> is called outside of a <code>\beginnumbering... \endnumbering</code> structure, it is automatically switched to <code>\index</code> , with a warning message.	1
When indexing texts in familiar footnotes with <code>\edtext</code> , refer to the line number where the footnote is called.	1
When indexing texts in sidenotes with <code>\edtext</code> , refer to the line number where the sidenote is called.	1
v2.16.1.	
General: Fix a bug with redefinition of the style of the footnote number (bug added in v2.12.0)	1
v2.16.2.	
General: Error message if <code>footmisc</code> is loaded after <code>reledmac</code> .	1
Fix a bug introduced by v2.16.1 when using non-expandable control sequence, like <code>\normalfont</code> , in the footnote number style.	1
v2.16.3.	
General: Fix a bug with <code>\SEref</code> (bug added in v2.7.0).	1
v2.16.4.	
General: Fix a bug with vertical space before sectioning command in optional argument of <code>\pstart</code> (bug added in v2.15.3).	1
v2.16.5.	
General: Fix potential spurious spaces in endnotes.	1
v2.16.6.	
General: Fix a bug with the line number style in <code>\doenotes</code> when referring to right side line in parallel typesetting.	1
Take into account <code>\linenumberstyle</code> when using <code>\edlineref</code> .	1
v2.16.7.	
General: Fix a bug with <code>\msdata</code> when using multiple <code>\beginnumbering... \endnumbering</code> .	1
Fix a bug with <code>\numberpstarttrue</code> when using multiple <code>\beginnumbering... \endnumbering</code> .	1
v2.16.8.	
General: Fix a bug with <code>\edindex</code> in footnotes in parallel typesetting.	1
v2.17.0.	
General: Add <code>\edglsadd</code> command.	1



Add <code>\setmsdataposition</code> setting. . . . .	1
v2.17.1.	
General: Fix spurious space in paragraphed footnotes when using 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 $\X_{\mathbb{E}}\TeX$ . . . . .	1
Fix a bug when using both <code>\Xnumberonlyfirstintwolines</code> or <code>\Xnumberonlyfirstinline</code> and <code>\Xparafootsep</code> and <code>\Xsymlinenum</code> . . . . .	1
v2.18.1.	
General: Fix a bug when using <code>\msdata</code> with Lua $\TeX$ or with the <code>hyperref</code> package. . . . .	1
v2.19.0.	
General: Add <code>\footnoteXmark</code> and <code>\footnoteXtext</code> commands. . . . .	1
Add better compatibility with the <code>csquotes</code> package when using familiar footnotes. . . . .	1
Fix a bug with paragraph indent after sectioning command. . . . .	1
v2.20.0.	
General: Add <code>\AtStartEveryStanza</code> , <code>\BeforeEveryStopStanza</code> , <code>\AtEndEveryPend</code> , <code>\AtStartEveryPstart</code> . . . . .	1
Add second optional argument to <code>\pstart</code> , <code>\pend</code> and <code>\stanza</code> . . . . .	1
Add starred version of <code>\AtEveryPstart</code> , <code>\AtEveryPend</code> , <code>\AtEveryStanza</code> and <code>\AtEveryStopStanza</code> . . . . .	1
Add third and fourth optional argument to <code>\newverse</code> . . . . .	1
Fix a bug when using familiar footnotes in <code>\eledsection</code> and related. . . . .	1
Reset font specification at the beginning of familiar footnotes. . . . .	1
v2.21.0.	
General: Add the possibility of nested <code>\sameword</code> commands. . . . .	1
Fix a bug when using formatting command in the argument of <code>\edindex</code> inside <code>\edtext</code> . . . . .	1
Now, as explained in the handbook, a <code>\edindex</code> inside <code>\edtext</code> only creates an index reference to the main text, and not to the critical footnote. . . . .	1
v2.22.0.	
General: Add <code>\txtbeforenotesonlyonceX</code> and <code>\Xtxtbeforenotesonlyonce</code> hooks. . . . .	1
Add <code>\txtbeforenotesX</code> hook. . . . .	1
Fix a bug added in v2.16.0 when using <code>\Xtxtbeforenotes</code> with paragraphed or normal footnotes. . . . .	1
Fix a bug with three and two columns critical footnotes, broken in v. 2.17.6. . . . .	1
v2.22.1.	
General: Compatibility with new version of <code>reledpar</code> . . . . .	1
Fix a bug with some commands inside <code>\sameword</code> . . . . .	1
v2.22.2.	
General: Do not print footnotes at the first run. . . . .	1
v2.23.0.	
General: Add <code>swcaseinsensitive</code> option . . . . .	1
Optimisation of the code added on v2.22.2 to not print footnotes at the first run. . . . .	1

v2.24.0.	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</code> 2.20.2 . . . . .	1
v2.24.2.	General: Fix bug between tabular environments and endnotes . . . . .	1
v2.25.0.	General: Add <code>\Xendtxtbeforenotes</code> hook . . . . .	1
v2.26.0.	General: Add a mechanism for annotation of stanza line numbering . . . . .	1
	Fix bug in <code>\xpstartref</code> . . . . .	1
v2.26.1.	General: Fix bug with <code>\linenumannotation</code> . . . . .	1
v2.26.10.	General: Add error message when insertion a <code>\pstart</code> or <code>\pend</code> inside <code>\edtext</code> arguments . . . . .	1
	Really fix bug with page number in footnote after <code>\mainmatter</code> (bug introduced in v2.26.7) . . . . .	1
v2.26.11.	General: Allow to use <code>\edtext</code> inside <code>\sameword</code> . . . . .	1
v2.26.12.	General: Precise minimal version of <code>xparse</code> package required. . . . .	1
v2.26.2.	General: Fix bug with <code>\linenumannotation</code> in parallel typesetting . . . . .	1
	Fix bug with <code>\Xgroupbyline</code> and <code>\Xparafootsep</code> . . . . .	1
	Fix bug with <code>\Xgroupbyline</code> and optional argument of critical footnotes. . . . .	1
	Fix bug with <code>\Xgroupbyline</code> for the series after the first one. . . . .	1
v2.26.3.	General: Fix bug added by v2.26.0 with notes in parallel typesetting . . . . .	1
v2.26.4.	General: Fix bug when using <code>\numberpstarttrue</code> and alternating between manual <code>\pstart</code> and automatic <code>\pstart</code> produced by <code>\autopar</code> . . . . .	1
v2.26.5.	General: Check that users have a enough recent version of <code>xparse</code> . . . . .	1
	Fix bug when using <code>\eledsection</code> with the <code>continuousnumberingwithcolumns</code> of <code>reledpar</code> . . . . .	1
	Fix bug with <code>continuousnumberingwithcolumns</code> . . . . .	1
	Fix bug with using <code>\eledsection</code> and related when alternate normal typesetting and parallel typesetting . . . . .	1
v2.26.6.	General: Fix bug with <code>\lineation</code> by page and nexted <code>\edtext</code> between two pages (added in v. 2.24.0) . . . . .	1
v2.26.7.	General: Fix page number in footnote and endnotes when using <code>sameparallelpagenumber</code> option of <code>reledpar</code> and typesetting critical text in single mode after critical text in double page mode . . . . .	1
v2.26.8.	General: Fix bug in some specific cases, introduced by v.2.27.7 . . . . .	1

v2.26.9.		
General: Fix bug with critical footnotes line number when alternating between <code>\numberlinefalse</code> and <code>\numberlinetrue</code> .	.....	1
Fix bug with page number in footnote after <code>\mainmatter</code> (bug introduced in v2.26.7)		1
Workaround with a bug of L <sup>A</sup> T <sub>E</sub> X 2018/04/01 patch level 3 to generate the .sty file from a .ins file	.....	1
v2.27.0.		
General: Add <code>\doinsidethislinehook</code> .	.....	1
v2.27.1.		
General: Fix spurious space in <code>\hidenumbering</code> .	.....	1
v2.28.0.		
General: Familiar footnotes can take an optional argument to manually define the footnote mark.	.....	1
v2.28.1.		
General: Fix bug with the alignment of the columns of two and three columns critical footnotes.	.....	1
v2.29.0.		
General: Add <code>noresetlinenumannotation</code>	.....	1
Fix bug with <code>\linenumannotation</code> and <code>nofamiliar</code> option.	.....	1
Fix little bug with <code>\msdata</code>	.....	1
Really reset line number annotation at each line	.....	1
v2.30.0.		
General: Add <code>\Xnoidenticallinenumannotation</code> and <code>\Xendnoidenticallinenumannotation</code> hooks	.....	1
Add separator between line number annotations in margin. Add <code>\setlinenumannotationsep</code>	.....	1
v2.31.0.		
General: Add <code>noprefix</code> and <code>prefixmore</code> to <code>\appref</code> and <code>\SEref</code> .	.....	1
v2.31.1.		
General: Fix bug in vertical spacing with <code>\Xgroupbyline</code> .	.....	1
Fix bug with <code>\Xgroupbyline</code> , which could produce spurious blank lines.	.....	1
Fix bug with <code>\Xhangindent</code> and <code>\hangindentX</code> for right-to-left texts.	.....	1
Fix bug with indentation with <code>\Xgroupbyline</code> .	.....	1
Fix bug with multiple footnote layer and <code>\Xgroupbyline</code> .	.....	1
Fix incompatibility between <code>\Xgroupbyline</code> and <code>hyperref</code> package.	.....	1
Fix incompatibility between <code>\Xgroupbyline</code> and optional argument of critical footnotes.	.....	1
v2.31.2.		
General: Fix bug with familiar footnote containing non-default font (introduced with v. 2.20.0).	.....	1
v2.31.3.		
General: Fix bug with paragraphed footnote in RTL context.	.....	1
v2.32.0.		
General: Add <code>\Xlinenumannotationonlyfirst</code> , <code>\Xlinenumannotationonlyfirstintwo</code> , <code>\Xsymlinenumannotation</code> , <code>\Xendlinenumannotationonlyfirst</code> , <code>\Xendlinenumannotationonlyfirstintwo</code> and <code>\Xendsymlinenumannotation</code> .	.....	1
v2.32.1.		
General: Do not print <code>\Xendsep</code> if we print <code>\Xendsymlinenum</code> .	.....	1
When using <code>\parindentX</code> or <code>\Xparindent</code> and stanza, use the normal indentation in the footnote, not the verse indentation.	.....	1
v2.32.2.		
General: Fix spurious vertical space in familiar footnotes with LuaL <sup>A</sup> T <sub>E</sub> X.	.....	1

v2.32.3.	General: Fix bug with critical footnotes when typesetting Arabic text with <code>polyglossia</code> .	1
v2.32.4.	General: <code>\footnoteXmark</code> and <code>\footnoteXtext</code> can now have an optional argument, setting the footnote number.	1
	Fix bug <code>\footnoteXmark</code> and <code>\footnoteXtext</code> .	1
	Fix bug when using <code>\Xgroupbyline</code> and <code>twocol</code> 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- $X_{\text{L}}\text{T}_{\text{E}}\text{X}$ 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 <code>antilabe</code> 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.	314
	<code>\newhookcommand@series</code> : <code>\newhookcommand@series</code> can take a second optional argument (third argument).	267
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 <code>tabular</code> 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 <code>(a)stanza</code> .	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 <code>calc</code> package (added with v2.34.4).	1
v2.34.7.	
General: Fix bug with <code>\parafootsepX</code> in some circumstances (added with v2.34.4).	1
v2.35.0.	
General: Add <code>\linenumannotationothersidettrue</code> .	1
Add new hooks: <code>\Xnolinenum</code> , <code>\Xendnolinenum</code> , <code>\Xnolinenumberifannotation</code> , <code>\Xendnolinenumberifannotation</code> .	1
v2.35.1.	
General: Fix a bug when printing ending page number in endnotes and using the <code>hyperref</code> package.	1
v2.36.0.	
General: Add <code>\Xnonote</code> and <code>\nonoteX</code> .	1
Fix bug with <code>sublineation</code> 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 <code>\Xnonote</code>	1
Fix incompatibility between <code>\eledsection</code> (and related) and <code>hyperref</code> . Incompatibility was introduced by some changes in <code>hyperref</code> package.	1
Improve syntax check.	1
v2.37.1.	
General: Fix bug with <code>\edgls</code> and related in <code>\edtext</code> argument. Require <code>glossaries-extra</code> is <code>glossaries</code> is loaded.	1
v2.37.2.	
General: Fix bug with <code>\Xgroupbyline</code> and two or three columns notes.	1
Fix bug with <code>\Xstanzaonlyfirst</code> .	1
Fix bug with <code>parapparat</code> option and nested <code>\edtext</code> .	1
v2.38.0.	
General: Add <code>\Xbeforepagenumber</code> hook.	1
Add <code>\Xnopagenumberifcurrent</code> hook.	1
Add hyperlinks between footnote and lemma when using abbreviated line range.	1
Fix bug with <code>\lemma</code> and right-to-left typesetting with <code>X<sub>2</sub>TeX</code> .	1
Fix indent bugs with stanza and right-to-left typesetting.	1
v2.38.1.	
General: Fix incompatibility between <code>reledpar</code> 's option <code>continuousnumberingwithcolumns</code> and <code>lineation</code> by page.	1
v2.38.2.	
General: Fix bug when using <code>\applabel</code> in conjunction with <code>hyperref</code> package.	1
Fix some trouble in indentation in stanza, when switching to some language with <code>polyglossia</code> .	1
v2.39.0.	
General: Add <code>\Xnotboxingsubline</code> and <code>\Xendnotboxingsubline</code> hooks.	1
v2.39.1.	
General: Fix bug with <code>index</code> in familiar and critical footnotes.	1
v2.39.2.	
General: Fix bug with <code>\edlabel</code> in <code>\eledsection</code> and related.	1
Fix bug with <code>continuousnumberingwithcolumns</code> when right column starts a new page and we use <code>lineation</code> by page.	1
Tools for <code>reledpar</code> v2.25.4	1
v2.39.3.	
General: Fix bug with <code>\edlabel</code> after tabular environment (added by v2.39.2).	1
v2.39.4.	
General: Fix bug with hyperlink when using <code>\footnoteXmark</code> and <code>\footnoteXtext</code> .	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
v2.40.0.	
General: Add\Xonlyonerule and \onlyoneruleX hooks . . . . .	1
Addonlyonerule option . . . . .	1
v2.40.1.	
General: Dealing with direction in \footnotetext when using LuaLaTeX . . . . .	1
Fix bug when using \edlabel in edtabular and related . . . . .	1
Reset \parfillskip after paragraphed and columned footnote . . . . .	1
v2.40.2.	
General: Bug with right pstart number with continuousnumberingwithcolumns . . . . .	1