

reledmac

Typeset scholarly editions with L^AT_EX*

Maïeul Rouquette[†]

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_EX 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

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

*This file (`reledmac.dtx`) has version number v2.39.1, last revised 2022/02/04.

[†]maieul at maieul dot net

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

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

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

11 Cross referencing	59
11.1 Basic use	59
11.2 Cross-referencing to a critical note	60
11.3 Cross-referencing which return a number in any case	60
11.3.1 Cross-referencing in order to define line number of a critical note	60
11.4 Not automatic cross-referencing	61
11.5 Normal \LaTeX cross-referencing	61
11.6 References to start and end lines	61
11.6.1 Reference to main text lines	61
11.6.2 References to lines that are commented on in the apparatus	62
11.6.3 Settings	62
11.6.4 Combining multiple references	64
11.7 Compatibility with xr package	64
12 Sidenotes	64
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
13 Indexing	65
13.1 Basics	65
13.2 Use with imakeidx or indextools	66
13.3 Referring to critical notes	66
13.4 Separator between page and line numbers	66
13.5 Using xindy	67
13.6 Advanced setting	67
14 Glossary	68
14.1 Preamble setting	68
14.2 Commands	68
15 Tabular material	68
16 Sectioning commands	71
16.1 Sectioning commands without line numbers or critical notes	71
16.2 Sectioning commands with line numbering and critical notes	71
16.3 Optimization	72
17 Quotation environments	72
18 Page breaks	73
18.1 Control page breaking	73
18.2 Prevent page break in a long verses	73

19 Miscellaneous	73
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	74
19.1.6 Paragraphed footnotes	75
19.1.7 Use with other packages	75
19.2 Parallel typesetting	76
I Implementation overview	77
II Preliminaries	77
II.1 Links with original EDMAC	77
II.2 Package declaration	77
II.3 Package options	78
II.3.1 Options of reledpar	78
II.3.2 Options of reledmac	78
II.4 Loading packages	80
II.5 Compatibility with Lua _T _E X	80
II.6 Boolean flags	81
II.7 Messages	82
II.8 Gobbling	88
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	90
III Sectioning commands	91
IV List macros	96
V Line counting	97
V.1 Choosing the system of lineation	97
V.2 Line number margin	98
V.3 Line number initialization and increment	99
V.4 Line number locking	100
V.5 Line number style	101
V.6 Line number printing	102
V.7 Line number counters and lists	103
V.8 Line number locking counter	104
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	122

VI Marking text for notes	130
VI.1 <code>\edtext</code> itself	131
VI.2 Substitute lemma	139
VI.3 Substitute line numbers	139
VI.4 Lemma disambiguation	141
VII Paragraph decomposition and reassembly	147
VII.1 Boxes, counters, <code>\pstart</code> and <code>\pend</code>	147
VII.2 Processing one line	153
VII.2.1 General process	153
VII.2.2 Process for “normal” line	154
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	157
VIII Line and page number computation	158
VIII.1 Continuous line numbering between parallel typesetting and normal typesetting	162
IX Line number annotation	162
X Line number printing	165
XI Pstart number printing inside	169
XII Restoring footnotes and penalties	170
XII.1 Add insertions to the vertical list	170
XII.2 Penalties	172
XII.3 Printing leftover notes	173
XII.4 Text before notes	174
XIII Critical footnotes	175
XIII.1 Fonts	176
XIII.2 Individual note options	176
XIII.3 Notes language	176
XIII.4 General survey of the way we manage notes	177
XIII.5 General setup	178
XIII.6 Footnotes arrangement	178
XIII.6.1 User level macro	178
XIII.6.2 Normal footnote	179
XIII.6.3 Paragraphed footnotes	184
XIII.6.4 Columnar footnotes	190
XIII.7 Footnote paragraph indent	197
XIII.8 Footnote hanging indent	197
XIII.9 Critical notes presentation	198
XIII.9.1 Font tools	198
XIII.9.2 Pstart number in footnote	199
XIII.9.3 Lemma printing	199
XIII.9.4 Line number printing	200
XIII.9.5 Footnote grouped by line	212

XIV Familiar footnotes	213
XIV.1 Adjacent footnotes	213
XIV.2 Regular footnotes for numbered texts	215
XIV.3 Footnote formats	217
XIV.4 Footnote arrangement	217
XIV.4.1 User level macro	217
XIV.4.2 Normal footnotes	217
XIV.4.3 Two columns footnotes	223
XIV.4.4 Three columns footnotes	225
XIV.4.5 Paragraphed footnotes	227
XIV.5 Wrapping footnote marks in hyperlink	231
XV Code common to both critical and familiar footnote in normal arrangement	232
XVI Footnotes' width for two columns	233
XVII Footnotes' order	234
XVIII Footnotes' rule	234
XIX Specific skip for first series of footnotes	235
XIX.1 Overview	235
XIX.2 User level command	235
XIX.3 Internal commands	236
XX Endnotes	237
XX.1 Internal commands	237
XX.2 User level commands	241
XX.2.1 Inserting contents to endnotes	241
XX.2.2 Printing endnotes	241
XXI Generate series of notes	250
XXI.1 Test if series is still existing	250
XXI.2 Init specific to reledpar	250
XXI.3 For critical footnotes	251
XXI.3.1 Options	251
XXI.3.2 Create inserts, needed to add notes in foot	253
XXI.3.3 Create commands for critical apparatus, \Afootnote, \Bfootnote etc.	253
XXI.3.4 Set standard display	255
XXI.4 For familiar footnotes	256
XXI.4.1 Options	256
XXI.4.2 Create tools for familiar footnotes (\footnoteX)	256
XXI.5 The endnotes	259
XXI.5.1 The auxiliary file	259
XXI.5.2 The main macro	259
XXI.5.3 Tools	260
XXI.5.4 Internal commands	260
XXI.5.5 The options	260
XXI.6 Init standards series (A,B,C,D,E)	262

XXII Setting series display	262
XXII.1 Change series order	262
XXII.2 Test series order	263
XXII.2.1 Get the first series	263
XXII.3 Series setting	263
XXII.3.1 General way of working	263
XXII.3.2 Tools to set options	264
XXII.3.3 Tools to generate options commands	265
XXII.3.4 Options for critical notes	267
XXII.3.5 Options for familiar notes	269
XXII.3.6 Options for endnotes	269
XXII.4 Hooks for a particular footnote	271
XXII.5 Alias	273
XXIII Output routine	273
XXIII.1 Extra footnotes output	273
XXIII.2 Patching standard output's commands	276
XXIV Page numbering in parallel typesetting	278
XXV Cross referencing	281
XXV.1 Compatibility with xref	296
XXVI Sidenotes	297
XXVII Minipages and such	305
XXVIII Indexing	309
XXVIII.1 Looking on package order	309
XXVIII.2 Auxiliary macros for \edindex	310
XXVIII.3 Code specific to \edindexin critical footnotes	311
XXVIII.4 Analysis of command in indexed text	312
XXVIII.5 Code for the formatted index	312
XXVIII.6 Main code	313
XXVIII.7 Hyperlink	315
XXVIII.8 "innote" and "notenumber" option of indextols package	317
XXIX Glossaries	318
XXX Verse	321
XXX.1 Hanging symbol management	321
XXX.2 Using & character	321
XXX.3 Code category setting	322
XXX.4 Stanza count and indent	322
XXX.5 Numbering stanza	323
XXX.6 Stanza number in note	324
XXX.7 Main work	325
XXX.8 Restore catcode and penalties	328
XXX.9 Dealing with antilabe	328

XXXI Apparatus of Manuscripts	329
XXXI.1 User level macro	329
XXXI.2 Setting macro	330
XXXI.3 Counters and lists	331
XXXI.4 Auxiliary file macros	331
XXXI.5 Action macro	332
XXXI.6 Inserting footnote	337
XXXI.7 Other	337
XXXII Arrays and tables	338
XXXII.1 Preamble: macro as environment	338
XXXII.2 Tabular environments	341
XXXII.2.1 Disabling and restoring commands	341
XXXII.2.2 Counters, boxes and lengths	345
XXXII.2.3 Tabular typesetting	348
XXXII.2.4 Environments	359
XXXIII Quotation's commands	359
XXXIV Section's title commands	360
XXXIV.1 Commands to disable some feature	360
XXXIV.2 General overview	360
XXXIV.3 <code>\beforeeledchapter</code> command	361
XXXIV.4 Auxiliary commands	361
XXXIV.5 Patching standard commands	362
XXXIV.6 Main code of <code>\eledxxx</code> commands	367
XXXIV.7 Macros written in the auxiliary file	369
XXXV Page breaking or no page breaking depending on specific lines	371
XXXVI Long verse: prevents being separated by a page break	373
XXXVII Tools for hyperref package	373
XXXVIII Compatibility with <code>eledmac</code>	374
Appendix A Things to do when changing versions	377
A.1 Migrating from EDMAC to <code>ledmac</code>	377
A.2 Migration from <code>ledmac</code> to <code>eledmac</code>	378
A.3 Migration to <code>eledmac</code> 1.5.1	379
A.4 Migration to <code>eledmac</code> 1.12.0	379
A.5 Migration to <code>eledmac</code> 17.1	379
A.6 Migration to <code>eledmac</code> 1.21.0	380
A.6.1 <code>\Xledsetnormalparstuff</code> and <code>\ledsetnormalparstuffX</code> . . .	380
A.6.2 Endnotes	380
A.7 Migration to <code>eledmac</code> 1.22.0	380
A.8 Migration to <code>eledmac</code> 1.23.0	380
A.9 Migration from <code>eledmac</code> to <code>reledmac</code>	380
A.9.1 Risk of 'no room for a new'	380
A.9.2 Multiple indices with memoir	381
A.9.3 Deprecated commands and options	381

A.9.4 <code>\renewcommand</code> replaced by <code>command</code>	381
A.9.5 Commands the names of which have been changed	382
A.9.6 Endnotes	383
A.9.7 Z Series	383
A.9.8 Internal commands	383
A.10 Migration to <code>reledmac</code> 2.1.0	383
A.11 Migration to <code>reledmac</code> 2.1.3	384
A.12 Migration to <code>reledmac</code> 2.3.0	384
A.13 Migration to <code>reledmac</code> 2.4.0	384
A.14 Migration to <code>reledmac</code> 2.5.0	384
A.15 Migration to <code>reledmac</code> 2.7.0	384
A.16 Migration to <code>reledmac</code> 2.7.2	385
A.17 Migration to <code>reledmac</code> 2.8.0	385
A.18 Migration to <code>reledmac</code> 2.13.1	385
A.19 Migration to <code>reledmac</code> 2.18.0	385
A.20 Migration to <code>reledmac</code> 2.21.0	385
A.21 Migration to <code>reledmac</code> 2.24.0	385
A.22 Migration to <code>reledmac</code> 2.26.0	386
A.23 Migration to <code>reledmac</code> 2.27.1	386
A.24 Migration to <code>reledmac</code> 2.30.0	386
A.25 Migration to <code>reledmac</code> 2.31.1	386
A.26 Migration to <code>reledmac</code> 2.32.7	386
A.27 Migration to <code>reledmac</code> 2.33.0	386
Appendix B Auxiliary softwares	386
B.1 <i>samewords</i>	386
B.2 critical-keys for <i>Emacs</i>	387
B.3 critical-marks for <i>Emacs</i>	387
B.4 Import from TEI	387
B.5 Import from TEI	387
References	388
Index	388
Change History	438

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.¹ A description by John and Dominik of this version of EDMAC was published as “An overview of EDMAC: a PLAIN \TeX format for critical editions”, *TUGboat* 11 (1990), pp. 623–643.

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

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

edmac@mailbase.ac.uk discussion group who helped us with smoothing out the bugs and infelicities in the macros. Ron Whitney and our anonymous reviewer at the TUG were both of great help in ironing out last-minute wrinkles, while Ron made some important suggestions which may help to make future versions of EDMAC even more efficient. Wayne Sullivan, in particular, provided several important fixes and contributions, including adapting the Mittelbach/Schöpf “New Font Selection Scheme” for use with PLAIN T_EX 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*,² an edition of the letters of Nicolaus Copernicus,³ Simon Bredon’s *Arithmetica*,⁴ a Latin translation by Plato of Tivoli of an Arabic astrolabe text,⁵ a Latin translation of part II of the Arabic *Algebra* by Abū Kāmil Shujā’ b. Aslam,⁶ the Latin *Rithmarchia* of Werinher von Tegernsee,⁷ a middle-Dutch romance epic on the Crusades,⁸ a seventeenth-century Hungarian politico-philosophical tract,⁹ an anonymous Latin compilation from Hungary entitled *Sermones Compilati in Studio Generali Quinqueecclesiensi in Regno Ungarie*,¹⁰ the collected letters and papers of Leibniz,¹¹ Theodosius’s *Spherics*, the German *Algorismus* of Sacrobosco, the Sanskrit text of the *Kāśikāvṛtti* of Vāmana and Jayāditya,¹² 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 T_EX to L^AT_EX. The starting point was EDMAC version 3.16 as documented on 19 July 1994 (available from CTAN). In August 2003 the `tabmac` functions were added; the starting point for these being version 1.0 of October 1996. The `edstanza` (v0.01) functions were added in February 2004. Sidenotes and regular footnotes in numbered text were added in April 2004. This port was called `ledmac` (L^AT_EX EDMAC).

²Gerhard 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).

³Being prepared at the German Copernicus Research Institute, Munich.

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

⁵Richard Lorch, Gerhard Brey *et al.*, at the same Institute.

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

⁷Menso Folkerts, “Die *Rithmarchia* des Werinher von Tegernsee”, *ibid.*

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

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

¹⁰Being produced, as was the previous book, by Gyula Mayer in Budapest.

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

¹²Being prepared at Poona and Lausanne Universities.

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

1.2.3 `eledmac`

Important changes were put in version 1.0, to make `ledmac` more easily extensible (see 7 p. 40). These changes can trigger small problems with the old customization. That is why a new name was selected: `eledmac` (extended `ledmac`).

To migrate from `ledmac` to `eledmac`, please read A.2 p. 378.

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. 380.

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/. 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 \XeTeX . If you make any subsequent changes altering the number of lines or notes, the input file may similarly require three passes to get everything to the right place. `reledmac` will tell you that you need to make more runs when it detects changes, but it does not expend the labor to check this thoroughly. If you have problems with a line or two misnumbered at the top of a page, try running \LaTeX once or twice more.

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

3 Compatibility warning

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

Please read 7.12.6 p. 52.

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

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

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

4 Options

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

4.1 Specific features

draft underlines lemmas in the main text.

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

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

antilabe enable `\antilabe` feature (see 9.11 p. 58);

eledmac-compat help to migrate from `eledmac` to `reledmac` (see A.9.5 p. 382);

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

nopbinverse prevents page break within verse environment;

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

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

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

swcaseinsensitive make `\sameword` command case insensitive;

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

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

4.2 Optimizing package performance

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

noeledsec disables tools for `\eledsection` and related commands (16.2 p. 71).

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

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

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

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

5 Text lines and paragraphs numbering

5.1 Text lines numbering

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

```
\beginnumbering
Text
\endnumbering
```

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

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

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

5.2 Paragraphs

5.2.1 Basics

`\pstart` `\pend` Within a numbered section, each paragraph of numbered text must be marked using the `\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.¹³

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

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

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

```

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

```

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

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

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

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

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

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

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

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

```

\AtStartEveryPstart{*}

```

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

Instead of manually doing

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

5.2.5 Numbering paragraphs (\pstart)

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

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

On each \beginnumbering the numbering restarts.

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.

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

5.2.6 Languages written in Right to Left

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

5.2.7 Memory limits

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

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

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

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

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

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

```
\numberpstarttrue
\numberpstartfalse
\thepstart
```

```
\sidepstartnumtrue
```

```
\labelpstarttrue
```

```
\pausenumbering
\resumenumbering
```

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

```
\newcommand{\memorybreak}{\pausenumbering\resumenumbering}
```

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

5.3 Lineation commands

5.3.1 Disabling lineation

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

This feature must be used with caution.

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

A successor to `reledmac` may disable this feature.

5.3.2 Setting lineation start and step

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

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

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

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

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

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

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

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

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

5.3.3 Setting lineation reset

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

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

5.3.4 Setting line number margin

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

```
\linenummargin{left}
```

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

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

5.3.5 Other settings

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

5.4 Changing the line numbers

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

5.4.1 Sublineation

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

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

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

`\Xsublinesep`
`\Xsublinesepside`

You can change the separator between the line number and the subline number either by using `\Xsublinesep` without any optional argument (7.4.11 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`
`\endlock`

The `\startlock` command, used in running text, locks the line number at its current value, until you insert `\endlock`. It can tell for itself whether you are in a patch of line or sub-line numbering. One use for line-number locking is in printing poetry: there the line numbers should be those of verse lines rather than of printed lines, even when a verse line requires several printed lines. But in this case you may use the `\stanza` mechanism, see 9 p. 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.¹⁴

`\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.

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

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

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

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

In the critical notes, the line number will be followed

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

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

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

will use A as starting annotation, while:

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

won't.

5.5.1 Resetting annotation

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

5.5.2 Modifying annotation associated with a specific note

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

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

5.5.3 Changing the position of the annotation

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

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

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

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

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

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

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

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

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

5.5.4 Changing the macro in which annotations are wrapped

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

- `\Xwraplinenumannotation[⟨s⟩]{⟨cmd⟩}` for critical footnotes; `⟨s⟩` indicates the series of notes to which it applies and may be a comma-separated list of series. When `⟨s⟩` is empty, the change applies to the annotations in the margins also—around the line number—as well as to the annotations referenced using the `\annotationref` command of the crossref mechanism (11 p. 59).

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

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

¹⁵As for all \TeX macro with no argument.

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

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

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

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

For example, if you do not wish to have annotations in the margins, but do want to have them in bold in the critical footnotes and endnotes, you say:¹⁶

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

5.5.5 Not printing the second identical annotations

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

5.5.6 Separator between annotations

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

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
\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

¹⁶The `\@gobble` command takes one argument, and returns nothing.

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

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 \LaTeX .

See the file `examples/2-subdivision-number-in-header.tex` for an example. Especially, the use of this command with \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).
- `linangesep=<c>` to change to `<c>` the separator between start line and end line for this particular note.

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

6.2.3 Endnotes

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

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

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

`\Aendnote[⟨option⟩]{⟨text⟩}`.

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

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

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

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

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

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

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

\toendnotes **Code between endnotes** Sometimes, it is useful to insert content between endnotes of the same series: for example to separate endnotes of different sections of the same text. In this case, you could use *inside numbered text* the command:
\Xtoendnotes `\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-expanded¹⁷ version of the `⟨content⟩`, that is the version where the commands are expanded on the first level. This can be, for example, to get a counter value. Use the starred version in this case. For example:

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

6.2.4 Paragraph in critical apparatus

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

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

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

6.2.5 Change lemma and line number

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

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

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

1 Smith] Jones C, D.

1–2 I ... Tuesday.] The date was July 16, 1954.

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

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

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

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

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

This command does not change the marginal line numbers in any way; it just changes the numbers passed to the notes. Its use comes in situations that `\edtext` has trouble dealing with for whatever reason. If you need notes for overlapping passages that aren't nested, for instance, you can use `\lemma` and `\linenum` to generate such notes despite the limitations of `\edtext`. If the *lemma* argument to `\edtext` is extremely long, you may run out of memory; here again you can specify a note with an abbreviated lemma using `\lemma` and `\linenum`. The numbers used in `\linenum` need not be entered manually; you can use the “x-” symbolic cross-referencing commands below (11 p. 59) to compute them automatically.

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

6.2.6 Changing the names of commands for critical apparatus

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

We recommend that you create a series of such aliases and use them instead of the names chosen here; all you have to do is put commands of this form at the start of your file:¹⁸

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

6.3 Disambiguation of identical words in the apparatus

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

6.3.1 Basic use

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

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

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

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

6.3.2 Case setting

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

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

6.3.3 Notes about input encoding with UTF-8 processor

If you use UTF-8 processor, like \XeTeX or \LuaTeX , there should not be any glitches. However, pay attention to how characters are encoded. Similar-looking characters may be represented differently in unicode numbering.

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

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

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

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

- If you use \XeTeX , add this line in your preamble: `\XeTeXinputnormalization 1`.
- If you use \LuaTeX , use the `uninormalize` package of Michal Hoftich¹⁹ with the `buffer` option set to true.

With these tools, \XeTeX / \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 \XeTeX , `\sameword` does not work with right-to-left text with this engine.

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

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


```

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

```

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

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

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

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

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

```

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

```

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

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

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

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

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

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

Here, the corresponding code:

```

\edtext{%
  \edtext{%
    \sameword[inlemma]{A} (1)
    \edtext{%
      \sameword{A} (2)
    }%
  }%
  \Afootnote{level~3}%
}%
\sameword[2]{A} (3)
}%
{%
  \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 A¹...A⁵] level 1

1 A¹...A³] level 2

1 A² (2)] level 3

6.3.6 Sameword for a group of words

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

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

```

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

```

1 per causam tamen scire causam est per causam cognoscere causam

1 causam²] fnote
 1 per² causam³] causam rei B
 1 causam⁴] fnote

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

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

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

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

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

```
1 causam2] fnote
```

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

```
1 causam4] fnote
```

6.3.7 Customizing

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

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

6.3.8 Problems with some macros

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

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

This will drop the content of \somemacro during comparison.

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

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

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

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

6.3.9 Automatic sameword annotation

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

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

6.4 Apparatus of manuscripts

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

To produce an apparatus of manuscripts with `reledmac`, you have to insert specific commands that are used to mark the sections for which only part of the manuscripts are relevant. These commands will be processed, and **after the second \TeX 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 \TeX two or more times. We strongly recommend to use tools like *latexmk* to do that.

6.4.1 Marking sections of text

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

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

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

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

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

Which means that the part of text containing “Lorem ipsum dolor [...] amet” is witnessed by manuscripts A, B and C, while the part of text starting “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 have the familiar marker in the text, and the marked text at the foot of the page can be

`\footnoteB`

`\footnoteC`

`\footnoteD`

`\footnoteE`

formatted using any of the styles described for the critical footnotes. Note that the “regular” footnotes have the series letter at the end of the macro name whereas the critical footnotes have the series letter at the start of the name.

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

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

6.5.2 Customizing mark

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

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

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

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

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

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

There are similar command triples for the other series.

6.5.3 Separator for multiple footnotes

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

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

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

and can be changed if necessary.

6.6 Printing the footnote mark without printing the footnote text

`\footnoteXmark` In certain cases, you can’t directly use `\footnoteX`; for example, when using `\uline` command of the `ulem` package. You need to print the footnote mark first, then call the footnote text to be inserted.

`\footnoteXtext` For all $\langle X \rangle$ command, `reledmac` provides a `\footnote\langle X \rangle mark` command and a `\footnote\langle X \rangle text` command, equivalent to standard \TeX ’s command `\footnotemark` and `\footnotetext`. For example, to use with `\uline`, do:

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

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

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

6.7 Changing series

6.7.1 Create a new series

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

6.7.2 Delete series

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

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

6.7.3 Series order

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

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

6.8 Position of critical and familiar footnotes

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

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

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

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

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

`<series><type>`

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

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

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

```

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

```

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

7 Apparatus customization

7.1 Introduction

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

When a length, noted $\langle l \rangle$, is used, it can be stretchable: a plus b minus c . The final length m is calculated by \LaTeX to have: $a - c \leq m \leq a + b$. If you use some relative unit²¹, it will be relative to font size of the footnote, except for commands concerning the place kept by the notes – including blank space.

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

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

There is also name convention:

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

7.2 Notes arrangement in a series

`\Xarrangement`
`\arrangementX`

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

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

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

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

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

- `paragraph` formats all of the footnotes of a series as a single paragraph; if you use this arrangement, you are strongly encouraged to read 19.1.6 p. 75.
- `twocol` formats them as separate paragraphs, but in two columns;
- `threecol`, in three columns.
- `normal`, restore normal arrangement.

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

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

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

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

7.3 Disabling footnote

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

`nonoteX`

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

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

7.4 Control number printing

7.4.1 Print line number only at first time

`\Xnumberonlyfirstinline`

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

`\numberonlyfirstintwolines`

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

`\Xsymlinenum`

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

`\endnumberonlyfirstinline`

`\numberonlyfirstintwolines`

`\Xendsymlinenum`

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

7.4.2 Print page number only at first time

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

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

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

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

In any case, you can use:

`\Xendsympagenum`

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

`\Xendinplaceofpagenumber`

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

7.4.3 Arbitrary text before line number

`\Xtxtbeforenumber`

For critical notes `\Xtxtbeforenumber[⟨s⟩]{⟨txt⟩}` allows the insertion of `⟨txt⟩` before the line number only when the line number is printed, so taking into account

`\Xbeforepagenumber`

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

`\Xendbeforepagenumber`

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

`\Xendafterpagenumber`

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

`\Xendlineprefixsingle`

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

`\Xendlineprefixmore`

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 `\Xendlinerrangeseparator` `[\langle s \rangle]{\langle text \rangle}`, and with `\Xendlinerrangeseparator` `[\langle s \rangle]{\langle text \rangle}` 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` `[\langle s \rangle]{\langle text \rangle}` and `\Xmorethantwolines` `[\langle s \rangle]{\langle text \rangle}`. The `\langle text \rangle` argument of `\Xtwolines` will be printed if the lemma is on two lines, and the `\langle text \rangle` argument of `\Xmorethantwolines` will be printed if the lemma is on three or more lines. For example:

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

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

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

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

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

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

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

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

7.4.6 Disabling line number printing

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

7.4.8 Printing pstart number

<code>\Xpstart</code>	You can use <code>\Xpstart[⟨s⟩]</code> 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: <ul style="list-style-type: none"> • If you use lineation by pstart, the option is enabled. • If you use lineation by section or by page, the option is disabled.
<code>\Xpstarteverytime</code>	By default, the pstart number is printed only in the part of text where you have called <code>\numberpstarttrue</code> . 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 <code>\Xpstarteverytime[⟨s⟩]</code> . In this case, the pstart number will be printed every time in footnote.
<code>\Xonlypstart</code>	In combination with <code>\Xpstart</code> , you can use <code>\Xonlypstart[⟨s⟩]</code> if you want to print only the pstart number in the footnote, and not the line and subline number.
<code>Xpstartonlyfirst</code>	You can use <code>\Xpstartonlyfirst[⟨s⟩]</code> to print the pstart number only the first time it appears in the apparatus. When using <code>\Xpstart</code> , you can use <code>\Xpstartseparator[⟨s⟩]{⟨text⟩}</code> to print <code>⟨text⟩</code> after the pstart number. Default value is empty.
<code>\Xpstartseparator</code>	

7.4.9 Printing stanza number

<code>\Xstanza</code>	You can use <code>\Xstanza[⟨s⟩]</code> 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 <code>\numberstanza</code>
<code>\Xstanzaseparator</code>	When using <code>\Xstanza</code> , you can use <code>\Xstanzaseparator[⟨s⟩]{⟨text⟩}</code> to print <code>⟨text⟩</code> after the stanza number. Default value is empty.
<code>Xstanzaonlyfirst</code>	You can use <code>\Xstanzaonlyfirst[⟨s⟩]</code> to print the stanza number only the first time it appears in the apparatus.

7.4.10 Options for line number annotations

<code>\Xlinenumannotationonlyfirst</code>	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 <code>\Xlinenumannotationonlyfirst[⟨s⟩]</code> . Note the two following points: <ul style="list-style-type: none"> • If you use the <code>noresetlinenumannotation</code> option of <code>reledmac</code>, the <code>\Xlinenumannotationonly</code> won't take account of the official line number. • The <code>\Xlinenumannotationonlyfirst</code> 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.
<code>\Xlinenumannotationonlyfirstintwo</code>	Suppose you have a lemma on annotation 1 and a lemma between annotation 1 and annotation 3. With <code>\Xlinenumannotationonlyfirst</code> , the second lemma is considered to have the same annotation as the first lemma. But if you use both <code>\Xlinenumannotationonlyfirst[⟨s⟩]</code> and <code>\Xlinenumannotationonlyfirstintwo[⟨s⟩]</code> , a distinction is made.
<code>\Xsymlinenumannotation</code>	To use a particular symbol in place of the line number annotation, you can use <code>\Xsymlinenumannotation[⟨s⟩]{⟨symbol⟩}</code> in combination with <code>\Xlinenumannotationonlyfirst[⟨s⟩]</code> . From the second lemma with the same line number annotation, the symbol will be used

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

`\endnumannotationonlyfirst` For endnotes, `\Xendlinenumannotationonlyfirst`; `\Xendlinenumannotationonlyfirstintwo` and `\Xendsymmlinenumannotation` are the equivalents of `\Xlinenumannotationonlyfirst`; `\Xlinenumannotationonlyfirstintwo` and `\Xsymmlinenumannotation`.
`\olinenumberifannotation` The `\Xnolinenumberifannotation` disable line number printing only for critical
`\nolinenumberifannotation` notes when there are some annotations to the line number.
 The `\Xendnolinenumberifannotation` is the same for endnotes.

7.4.11 Separator between line and subline numbers

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

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

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

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

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

7.4.12 Separator between page and line numbers

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

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

7.4.13 Space around number

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

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

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

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

7.4.14 Space around line symbol

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

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

`\Xendbeforemsymmlinenum` `\Xendbeforemsymmlinenum` and `\Xendaftersymmlinenum` are the equivalents of `\Xbeforemsymmlinenum` and `\Xaftersymmlinenum` for the endnotes.

7.4.15 Space in place of number

<code>\Xinplaceofnumber</code>	If no number or symbolic line number is printed, you can add a space, with <code>\Xinplaceofnumber[\langle s \rangle]{\langle l \rangle}</code> . The default value is 1 em.
<code>\Xendingplaceofnumber</code>	<code>\Xendingplaceofnumber[\langle s \rangle]{\langle l \rangle}</code> is the same, for critical endnotes.

7.4.16 Boxing line number and line symbol

<code>\Xboxlinenum</code>	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 <code>\Xboxlinenum[\langle s \rangle]{\langle l \rangle}</code> to do that. To subsequently disable this feature, use <code>\Xboxlinenum</code> 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: <pre> \Xhangindent{1em} \Xafternumber{0em} \Xboxlinenum{1em} </pre>
<code>\Xboxsymlinenum</code>	<code>\Xboxsymlinenum[\langle s \rangle]{\langle l \rangle}</code> is the same as <code>\Xboxlinenum</code> but for the line number symbol.
<code>\Xendboxsymlinenum</code>	<code>\Xendboxsymlinenum[\langle s \rangle]{\langle l \rangle}</code> is the same as <code>\Xboxsymlinenum</code> but for endnotes.
<code>\Xboxlinenumalign</code>	If you put line number in box, it will be aligned left inside the box. However, you can change it using <code>\Xboxlinenumalign[\langle s \rangle]{\langle text \rangle}</code> where <code>\langle text \rangle</code> 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[\langle s \rangle]{\langle l \rangle}` will box the start line number in a box of length `\langle l \rangle`. The content will be put at the right of the box.
- `\Xboxendlinenum[\langle s \rangle]{\langle l \rangle}` will box the dash plus the end line number or the range symbol in a box of length `\langle l \rangle`. The content will be put at the left of the box.

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

```

1
12-23
24ff.

```

<code>\Xnotboxingsubline</code>	By default, for the start line number, these things are boxed: <ul style="list-style-type: none"> • (start) line number annotation, if before line number; • (start) page and (start) line number;
---------------------------------	--

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

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

7.5 Separator between the lemma and the note

7.5.1 For footnotes

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

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.

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.

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

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

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.

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

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.

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

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

7.6 Font style

7.6.1 For line number

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

7.6.2 For the lemma

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

7.6.3 For all notes

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

7.7 Wrapping notes

7.7.1 Wrapping lemmas

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

7.7.2 Wrapping contents

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

7.8 Indent of notes content

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

7.9 Arbitrary code

7.9.1 Arbitrary code around line number

<code>\Xendbhooklinenumber</code>	<code>\Xendbhooklinenumber[\langle s \rangle]{\langle code \rangle}</code> is used to execute code before line numbers in endnotes. The code is executed before the <code>\Xendbeforelinenumber</code> space and before the <code>\Xendnotenumfont</code> font setting.
<code>\Xendahooklinenumber</code>	<code>\Xendahooklinenumber[\langle s \rangle]{\langle code \rangle}</code> is used to execute code after line number in endnotes. The code is executed after the <code>\Xendafternumber</code> space.
<code>\Xendbhookinplaceofnumber</code>	<code>\Xendbhookinplaceofnumber[\langle s \rangle]{\langle code \rangle}</code> is used to execute code before a space or symbol which replaces line number in endnotes. The code is executed before the <code>\Xendbeforestylinenum</code> space and before the <code>\Xendnotenumfont</code> font setting.
<code>\Xendahookinplaceofnumber</code>	<code>\Xendahookinplaceofnumber[\langle s \rangle]{\langle code \rangle}</code> is used to execute code after a space or symbol which replaces line number in endnotes. The code is executed after the <code>\Xendaftersylinenum</code> 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

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

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

Note that the changes are local to the footnote.

7.10 Options for footnotes in columns

7.10.1 Alignment

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

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

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.

`\afternoteX`

`\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}`.

`\parafootsepX` 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).

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⟩]`.

`\Xgroupbylineseparetwolines` 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.

`\txtbeforenotesonlyonceX`

7.12.3 Code before notes

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

7.12.4 Spacing

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

Be careful, the standard L^AT_EX footnote rule used by reledmac decreases by 3pt. This 3pt decrease is not changed by this command.

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

Be careful, the standard L^AT_EX footnote rule, which is used by reledmac, decreases 3pt. These 3pt are not changed by this command.

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

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

7.12.5 Rule

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

Be careful, the standard L^AT_EX 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^AT_EX footnote rule, which is used by reledmac, adds 2.6pt. These 2.6pt are not changed by this command.

7.12.6 Maximum height

`\Xmaxhnotes` By default, one series of critical notes can take up to 80% of `\vsize`, before being broken to the next page. If you want to change the size use `\Xmaxhnotes[⟨s⟩]{⟨l⟩}`. Be careful: the length must be fixed (no stretch), and is relative to the the current font. For example, if you want the note to take, at most, 33% of the text height, do `\Xmaxhnotes{.33\textheight}`.

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

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

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

Be careful with the two previous commands. Actually, for technical purposes, one paragraphed note is considered as one block. Consequently, it cannot be broken between two pages, even if you used these commands. The debug is in the *todolist*.

7.12.7 Width

`\Xwidth` `\Xwidth[⟨s⟩]{⟨l⟩}` sets the total width of critical footnotes. `\widthX[⟨s⟩]{⟨l⟩}` does the same for familiar footnotes.
`⟨l⟩` can be a length expression, parsable with `\dimexpr`. For example:

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

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

7.13 Footnotes and the *reledpar* columns

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

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

7.14 Endnotes in one paragraph

`\Xendparagraph` By default, any new endnote starts a new paragraph. Use `\Xendparagraph[⟨s⟩]` to have all end notes of one given series set in one paragraph.
`\Xendafternote` You can add some space after a endnote series by using `\Xendafternote[⟨s⟩]{⟨l⟩}`. The default value is `1em plus .4em minus .4em`.
`\Xendsep` You can choose the separator between each note by `\Xendsep[⟨s⟩]{⟨text⟩}`. A common separator is the double pipe (`||`), which you can set by using `\Xendsep{\parallel}`.

8 Fonts

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

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

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

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

`\select@lemmafонт` We will briefly discuss `\select@lemmafонт` here because it is important to know

about it now, although it is not one of the macros you would expect to change in the course of a simple job. Hence it is “protected” by having the @-sign in its name.

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

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

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

9 Verse

9.1 Basic

`\stanza` Use `\stanza` at the start of a stanza. Each line in a stanza is ended by an ampersand (&), and the stanza itself is ended by putting `\&` at the end of the last line.

If you need to add brackets directly after `\stanza`, `&` or `\&`, add `\norelax`. Otherwise, the brackets will be interpreted as delimitation of an optional argument (cf. 9.9 p. 56)

9.2 Define stanza indents

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

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

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

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

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

If you want the hanging verse to be flush right, you can use `\sethanginsymbol:` see p. 9.7 p. 56.

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

9.3 Repeating stanza indents

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

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

is like

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

Be careful: the feature is changed in `eledmac` 1.5.1. See A.3 p. 379.

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

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

```
\setcounter{stanzaindentsrepetition}{0}
```

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

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

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

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

9.4 Notes about the scope of the indent settings

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

9.5 Manual stanza indent

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

`\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{<h>}` with this code. In the example of French typography, do

`\sethangingsymbol`

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

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

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

9.8 Long verse and page break

If you want to prevent page breaks inside long verses, use the option `nopbinverse` when loading package, or use `\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 \TeX .

`\stanzanumwrapper`

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

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

```
\flagstanza{#1}%
}
```

```
\antilabe
```

9.11 Antilabe

In dramatic text, it may happen that a line of verse is split between two speakers. That's 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 expands argument, you can use one of 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. 56).

10 Grouping

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

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

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

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

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

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

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

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

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

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

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

11 Cross referencing

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

11.1 Basic use

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

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

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

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

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

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

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

11.2 Cross-referencing to a critical note

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

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

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

11.3 Cross-referencing which return a number in any case

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

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

For this situation, `reledmac` supplies variants of the reference commands, with the `x` prefix: `\xpageref`, `\xlineref`, `\xsublineref`, `\xpstartref` and `\xannotationref`. They have the following limitations:

- They will not tell you if the label is undefined.
- They must be preceded in the file by at least one of the four other cross-reference commands—e.g., a `\edlabel{foo}` command, even if you never refer to that label—since those commands can all do the necessary processing of the `.aux` file, and the `\x...` ones cannot.
- When `hyperref` is loaded, the `hyperref` link will not be added. (Indeed, it is not a limitation, but a feature.)
- With `reledpar`, the `\xlineref` does not insert the right side flag, in order to obtain a line number. Use `\xflagref` to obtain the side flag, depending on your flag.

11.3.1 Cross-referencing in order to define line number of a critical note

`\xxref`

The `\xxref{<lab1>}{<lab2>}` command generates a reference to a sequence of lines, for use in the second argument of `\edtext`. It takes two arguments, both of which are labels: e.g., `\xxref{mouse}{elephant}`.

It automatically calls `\linenum` (6.2.5 p. 30) and `\lineannot` (5.5.2 p. 25) and sets the beginning page, line, subtitle numbers and line annotations to those of the place where `\edlabel{mouse}` was placed, and the ending ones to those where `\edlabel{elephant}` occurs.

For example, one might use the following:

```
\beginnumbering

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

\pstart
\edtext{Et videtur quod non\edlabel{non}.}{\xxref{Queritur}{non}\lemma{queritur \dots{} non}\Afootnot
\pend

\endnumbering
```

11.4 Not automatic cross-referencing

`\edmakelabel` Sometimes the `\edlabel` command cannot be used to specify exactly the page and line desired—for example, if you want to refer to a page and line number in another volume of your edition. In such cases, you can use the `\edmakelabel{<lab>}{<numbers>}` macro so that you can “roll your own” label.

For example, if you type `\edmakelabel{elephant}{10|25|0}` you will create a new label, and a later call to `\edpageref{elephant}` would print “10” and `\lineref{elephant}` would print “25”. The sub-line number here is zero. It is usually best to collect your `\edmakelabel` statements near the top of your document, so that you can see them at a glance.

11.5 Normal L^AT_EX cross-referencing

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

11.6 References to start and end lines

11.6.1 Reference to main text lines

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

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

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

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

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

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

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

11.6.2 References to lines that are commented on in the apparatus

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

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

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

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

11.6.3 Settings

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

For example, you may use:

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

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

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

`\setSerefonlypageprefixsingle` Use `\setSerefonlypageprefixsingle{<prefix>}` to set the page prefix for `\Serefonlypage` when there is only one page. Use `\setSerefonlypageprefixmore{<prefix>}` to set it when there is more than one page. For example:

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

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

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

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

These commands are the following:

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

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

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

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

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

For one specific command When calling `\appref` and `\SEref`, you can use as a first optional argument, in brackets (`[]`), any optional argument which can be used for critical footnotes (6.2.2 p. 28).

When calling `\apprefwithpage`, `\SErefwithpage` or `\SErefonlypage` you can use as a first optional argument, in brackets (`[]`), any optional argument which can be used for critical endnotes (6.2.3 p. 29).

11.6.4 Combining multiple references

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

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

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

11.7 Compatibility with `xr` package

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

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

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

12 Sidenotes

12.1 Basics

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

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

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

`\ledrightnote` The package’s default setting is `\sidenotemargin{right}`

`\ledsidenote` 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.

`\sidenotemargin`

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 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) margin. These lengths are initially set to the value of `\linenumsep`.

12.2.4 Font

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

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

These can of course be changed to suit.

12.2.5 Separator between notes

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

13 Indexing

13.1 Basics

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

Note that the file `.idx` will contain the right reference only after the third run, 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`
`\innotemarkX`

You can redefine the way the reference is made using:

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

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

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

For example:

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

13.4 Separator between page and line numbers

`\pagelinesep`

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

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

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

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

However, you also have to configure your `.ist` index style file. For example, if you use `:` as separator²⁴.

```
page_compositor ":"
```

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

13.5 Using xindy

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

```
pagenumber-linenummer
```

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

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.²⁶

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.

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

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

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

14 Glossary

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

14.1 Preamble setting

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

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

- Or set `.glossaries`:
`\glsSetCompositor{-}`
- Or set reledmac:
`\renewcommand{\pagelinesep}{.}`
 In this case, the above will have consequences for your use of `\edindex` and you should set your `.ist` file (13.4 p. 66).

14.2 Commands

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

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

15 Tabular material

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

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

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

1	2	3
a	bb	ccc
AAA	BB	C

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

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

For example:

```
\beginnumbering
```


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

You can also define your own “fill”. For example:

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

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

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

1	2	3	4
a	└─		d
A	B	C	D

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

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

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

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

For example:

```
\begin{edarrayl}
```

```

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

```

Before	<i>A</i>	1	2	3	After
	<i>B</i>	1	3	6	
	<i>C</i>	1	4	8	
	<i>D</i>	1	5	0	

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

```

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

```

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

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

16 Sectioning commands

16.1 Sectioning commands without line numbers or critical notes

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

```

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

```

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

16.2 Sectioning commands with line numbering and critical notes

You have to use the following commands:

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

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

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

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

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

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

16.3 Optimization

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

17 Quotation environments

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

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

18 Page breaks

18.1 Control page breaking

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

`\ledpb`
`\lednopb`

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

These commands have effect only at the second run.

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

`\ledpbsetting`

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

18.2 Prevent page break in a long verses

`\lednopbinversetrue`

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

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

19 Miscellaneous

`\extensionchars`

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

`\ifledfinal`

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

`\showlemma`

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

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

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

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

so that its argument is bold, which may make it easier to check that all lemmas have been treated.

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

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

19.1 Known and suspected limitations

19.1.1 Non-standard geometry

If you use classes other than `article` or `book`, or if you use the `geometry` package, you should use `maxhnotesX` and/or `\Xmaxhnotes` as explained in 7.12.6 p. 52 in order to prevent footnotes from overlapping the bottom margin.

19.1.2 floatrow package compatibility

The `floatrow` package must be loaded before the `reledmac`.

19.1.3 “No room for a new”

Sometimes, especially when using `reledmac` with other packages, you could obtain warning messages such ‘no room for a new count’ or ‘no room for a new write’.

In order to prevent such problems, the first thing is to use the options to optimize `reledmac`. For example, if you need only two series of notes, use the `series={A,B}` option. Read 16.3 p. 72 in order to know which are the available options.

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

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

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

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

19.1.4 Marginal notes

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

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

19.1.5 Paragraph shape

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

`\ballast`

\LaTeX is a three-pass system, but even after a document has been processed three times, there are some tricky situations in which the page breaks decided by \TeX never settle down. At each successive run, `reledmac` may oscillate between two different sets of page decisions. To stop this happening, should it arise, Wayne Sullivan suggested the

inclusion of the quantity `\ballast`. The amount of `\ballast` will be subtracted from the penalties which apply to the page breaks calculated on the *previous* run through \TeX , thus reinforcing these breaks. So if you find your page breaks oscillating, insert `\setcounter{ballast}{100}` or some such figure, and with any luck the page breaks will settle down. Luckily, this problem does not crop up at all often.

19.1.6 Paragraphed footnotes

The restriction on explicit line-breaking in paragraphed footnotes, mentioned on 7.2 p. 41, and described in more detail on XIII.6.3 p. 186, 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 `\Xmaxnotes` and / or `\maxnotesX` (7.12.6 p. 52), in order to define the maximum height relative to `\textheight` and not to `\vsize`, because the `\vsize` value is not the same inside and outside of the preamble.

19.1.7 Use with other packages

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

`\morenoexpands` You can define the macro `\morenoexpands` to modify macros that you call within `\edtext`. Because of the way `reledmac` numbers the lines the arguments to `\edtext` can be processed more than once and in some cases a macro should only be processed once. One example is the `\colorbox` macro from the `color` package, which you might use like this:

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

If you actually try this²⁷ you will find \TeX whinging “Missing { inserted”, and then things start to fall apart. The trick in this case is to specify either:

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

²⁷Reported by Dirk-Jan Dekker in the CTT thread “Incompatibility of ‘color’ package” on 2003/08/28.

or

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

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

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

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

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

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

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

19.2 Parallel typesetting

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

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

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}[2022/02/04 v2.39.1 typesetting critical editions
4 ]%
```

II.3 Package options

<code>\ifparledgroup</code>	Use this to remember which option is used, set and execute the options with final as the
<code>\ifsameparallelepagenumber</code>	default. We use xkeyval in order to manage options with argument.
<code>\ifprevpgnotnumbered</code>	
<code>\ifledfinal</code>	<code>\RequirePackage{xkeyval}</code>
<code>\ifnocritical@</code>	<code>%</code>
<code>\if@noeled@sec</code>	
<code>\ifnoend@</code>	
<code>\ifnofamiliar@</code>	
<code>\ifnoledgroup@</code>	
<code>\ifparapparatus@</code>	
<code>\ifnoquotation@</code>	<code>\newif\ifparledgroup</code>
<code>\iflednopbinverse</code>	<code>\newif\ifsameparallelepagenumber%</code>
<code>\ifwidthliketwocolumns</code>	<code>\newif\ifprevpgnotnumbered%%</code>
<code>\ifxindy@</code>	<code>%</code>
<code>\ifxindyhyperref@</code>	
<code>\ifeledmaccompat@</code>	
<code>\ifnoresetlinenumannotation@</code>	
<code>\ifantilabe@</code>	

II.3.1 Options of reledpar

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

II.3.2 Options of reledmac

```

11 \DeclareOptionX{series}[A,B,C,D,E]{\xdef\default@series{#1}}
12 \ExecuteOptionsX{series}%
13
14 \newif\if@noeled@sec%
15 \DeclareOptionX{noeledsec}{\@noeled@sectrue}
16
17 \newif\ifnocritical@%
18 \DeclareOptionX{nocritical}{\nocritical@true}%
19
20 \newif\ifnofamiliar@%
21 \DeclareOptionX{nofamiliar}{\nofamiliar@true}%
22
23 \newif\ifnoledgroup@%
24 \DeclareOptionX{noledgroup}{\noledgroup@true}%
25
26 \newif\ifnoend@%
27 \DeclareOptionX{noend}{%
28   \let\l@dend@open\@gobble%
29   \let\l@dend@close\relax%
30   \global\let\l@dend@stuff=\relax%
31   \noend@true%
32 }%
33
34 \newif\ifnoquotation@
35 \DeclareOptionX{noquotation}{\noquotation@true}
36
37 \newif\ifledfinal
38 \DeclareOptionX{final}{\ledfinaltrue}
39 \DeclareOptionX{draft}{\ledfinalfalse}
40 \ExecuteOptionsX{final}
41
42 \newif\ifparapparatus@
43 \DeclareOptionX{parapparatus}{\parapparatus@true}
44
```

```

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

```

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

```

93 \ProcessOptionsX*\relax
94
95 %

```

II.4 Loading packages

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

```

96 \RequirePackage{xargs}
97 \RequirePackage{xparse}[2017/03/07]%
98 \RequirePackage{etoolbox}
99 \ifl@t@r\fmtversion{2015/10/01}
100 {\ifboolexpr{not test{\@ifl@t@r\fmtversion{2016/03/31}}} or (test{\
101   ifdefstring{\fmtversion}{2016/03/31}} and test {\ifnumless{\patch@level
102   }{3}})}}%
103   {\PackageWarning{reledmac}{You are using a LaTeX version older than
104   2016/03/31 patch 3.%
105   \MessageBreak You are strongly encouraged to use a newer version.}}%
106   }%
107 }%
108 {\RequirePackage{etex}%
109 \csname reserveinserts\endcsname{32}%
110 }%
111 \RequirePackage{suffix}
112 \RequirePackage{xstring}
113 \RequirePackage{ifluatex}
114 \RequirePackage{ragged2e}
115 \RequirePackage{ifxetex}%
116 %

```

II.5 Compatibility with LuaTeX

Here, we enable some primitives for LuaTeX.

```

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

```


II.6 Boolean flags

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

II.7 Messages

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

`\reledmac@warning` Write a warning message.

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

`\reledmac@error` Write an error message.

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

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

```
\led@err@samewordRTL58 \newcommand*{\led@err@samewordRTL}{%
159 \reledmac@error{You can't use \string\sameword \MessageBreak with XeLaTeX
\MessageBreak when typesetting RTL text.\MessageBreak Please use LuaTeX
instead.}{\@ehc}%
160 }%
161 %
```

```
\led@err@edtextoutsidepstart62 \newcommand*{\led@err@edtextoutsidepstart}{%
\led@err@PstartInEdtext63 \reledmac@error{\string\edtext\space outside numbered paragraph (\pstart\
\led@err@PendInEdtext space ... \space\pend)}{\@ehc}}%
164
165 \newcommand{\led@err@PstartInEdtext}[1]{%
166 \reledmac@error{\string\pstart\space in \string\edtext\space #1 argument
}{\@ehc}%
167 }%
168
169 \newcommand{\led@err@PendInEdtext}[1]{%
170 \reledmac@error{\string\pend\space in \string\edtext\space #1 argument}{\@ehc}%
171 }%
172 %
```

```
\led@mess@NotesChanged73 \newcommand*{\led@mess@NotesChanged}{%
174 \typeout{reledmac reminder: }%
175 \typeout{ The number of the footnotes in this section
176 has changed since the last run.}%
177 \typeout{ You will need to run LaTeX two more times
```

```

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

```

```

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

```

```

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

```

```

\led@warn@BadLineation89 \newcommand*\led@warn@BadLineation){%
\led@warn@BadLinenummargin90 \reledmac@warning{Bad \string\lineation\space argument}}
\led@warn@BadLockdisp91 \newcommand*\led@warn@BadLinenummargin){%
\led@warn@BadSubblockdisp92 \reledmac@warning{Bad \string\linenummargin\space argument}}
193 \newcommand*\led@warn@BadLockdisp){%
194     \reledmac@warning{Bad \string\lockdisp\space argument}}
195 \newcommand*\led@warn@BadSubblockdisp){%
196     \reledmac@warning{Bad \string\subblockdisp\space argument}}
197 %

```

```

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

```

```

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

```

```

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

```

```

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

```

```

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

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

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

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

```

```

\led@warn@BadHookSeries 259 \newcommand{\led@warn@BadHookSeries}[2]{%
260   \reledmac@warning{Bad series #1 for #2 hook. Skipped.}%
261 }%
262 %

\led@warn@BadSidenotemargin 263 \newcommand*\led@warn@BadSidenotemargin{%
264   \reledmac@warning{Bad \string\sidenotemmargin\space argument}}
265 %

\led@warn@NoIndexFile 266 \newcommand*\led@warn@NoIndexFile}[1]{%
267   \reledmac@warning{Undefined index file #1}}
268 %

\led@warn@SeriesStillExist 269 \newcommand{\led@warn@SeriesStillExist}[1]{%
270   \reledmac@warning{Series #1 is still existing !}%
271 }%
272 %

\led@err@BadAction 273 \newcommand*\led@err@StanzaIndentNotDefined{%
274   \reledmac@error{You have not defined the indentation for the line \number
\stanza@count}{\@ehc}}%
275 %

\led@err@ManySidenotes 276 \newcommand{\led@err@ManySidenotes}{%
\led@err@ManyLeftnotes 277   \ifledRcol{%
\led@err@ManyRightnotes 278     \reledmac@warning{\itemcount@\space sidenotes on line \the\line@numR\
space p. \the\page@numR}%
279     \else%
280     \reledmac@warning{\itemcount@\space sidenotes on line \the\line@num\
space p. \the\page@num}%
281     \fi%
282   }%
283   \newcommand{\led@err@ManyLeftnotes}{%
284     \ifledRcol{%
285       \reledmac@warning{\itemcount@\space leftnotes on line \the\line@numR\
space p. \the\page@numR}%
286       \else%
287       \reledmac@warning{\itemcount@\space leftnotes on line \the\line@num\
space p. \the\page@num}%
288       \fi%
289     }%
290   \newcommand{\led@err@ManyRightnotes}{%
291     \ifledRcol{%
292       \reledmac@warning{\itemcount@\space rightnotes on line \the\line@numR\
space p. \the\page@numR}%
293       \else%
294       \reledmac@warning{\itemcount@\space rightnotes on line \the\line@num\
space p. \the\page@num}%

```

```

295 \fi%
296 }%
297 %

```

```

\led@err@TooManyColumns 98 \newcommand*{\led@err@TooManyColumns}{%
\led@err@UnequalColumns 99 \reledmac@error{Too many columns}{\@ehc}}
\led@err@LowStartColumn 00 \newcommand*{\led@err@UnequalColumns}{%
\led@err@HighEndColumn 01 \reledmac@error{Number of columns is not equal to the number
\led@err@ReverseColumns 202 in the previous row (or \protect\\ \space forgotten?)}{\@ehc}}
203 \newcommand*{\led@err@LowStartColumn}{%
204 \reledmac@error{Start column is too low}{\@ehc}}
205 \newcommand*{\led@err@HighEndColumn}{%
206 \reledmac@error{End column is too high}{\@ehc}}
207 \newcommand*{\led@err@ReverseColumns}{%
208 \reledmac@error{Start column is greater than end column}{\@ehc}}
209 %

```

```

\led@err@toendnotes@outsidenumbering 10 \newcommand{\led@err@toendnotes@outsidenumbering}{%
311 \reledmac@error{\string\toendnotes\space and related commands must be
called inside a numbered text (\string\beginnumbering\space ... \space\string\endnumbering)}{\@ehc}%
312 }%
313 %

```

```

\led@err@EdtextWithoutNote 14 \newcommand{\led@err@EdtextWithoutNote}[1]{%
315 \reledmac@error{edtext ``\unexpanded{#1}` without Xfootnote or Xendnote.
Check syntax.}{\@ehc}%
316 }%
317 %

```

```

\led@err@FootnoteNotInSecondArgEdtext 18 \newcommand{\led@err@FootnoteNotInSecondArgEdtext}[1]{%
319 \reledmac@error{#1 footnote outside of the second argument of an edtext.
Check syntax.}{\@ehc}%
320 }%
321 %

```

```

\led@error@PackageAfterEledmac 22 \newcommand{\led@error@PackageAfterEledmac}[1]{%
323 \reledmac@error{#1 must be loaded before reledmac}{\@ehc}%
324 }%
325 %

```

```

\led@error@require@glossariesextra 26 \newcommand{\led@error@require@glossariesextra}{%
327 \reledmac@error{When using both reledmac and glossaries, you must also
use glossaries-extra}{\@ehc}%
328 }%
329 %

```

```

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

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

\newcommand{\led@error@fail@patch@@doclearpage}{%
\reledmac@error{Fail to patch \string\@doclearpage\space command}{\@ehc}%
}%
%

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

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

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

\newcommand{\led@error@fail@patch@setcounter}{%
\reledmac@error{Failed to patch the \string\setcounter\space command}{\@ehc}%
}%
%

\newcommand{\led@error@fail@patch@addtocounter}{%
\reledmac@error{Failed to patch the \string\addtocounter\space command}{\@ehc}%
}%
%
```

```

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

```

```

\led@warn@edinde@outsidenumbering66 \newcommand{\led@warn@edinde@outsidenumbering}{%
367   \reledmac@warning{\string\edindex\space called outside of \string\
beginnumbering\space ... \space \string\endnumbering. \MessageBreak
Automatically switched to \string\index.}%
368 }%
369 %

```

```

\led@warning@hsizeX@deprecated70 \newcommand{\led@warning@hsizeX@deprecated}{%
371   \reledmac@warning{\string\hsizeX\space command deprecated, use \string\
widthX\space instead.}%
372 }%
373 %

```

```

\led@warning@Xhsize@deprecated74 \newcommand{\led@warning@Xhsize@deprecated}{%
375   \reledmac@warning{\string\Xhsize\space command deprecated, use \string\
Xwidth\space instead.}%
376 }%
377 %

```

```

\led@warning@msdatawithoutstop78 \newcommand{\led@warning@msdatawithoutstop}{%
379   \reledmac@warning{\string\msdata\space without corresponding \string\
stopmsdata}%
380 }%
381 %

```

```

\led@warning@preXnotes@deprecated82 \newcommand{\led@warning@preXnotes@deprecated}{%
383   \reledmac@warning@preXnotes@deprecated%
384 }%
385 %

```

II.8 Gobbling

Here, we define some commands which gobble their arguments.

```

\@gobblethree86 \providecommand*\@gobblethree}[3]{}
\@gobblefour87 \providecommand*\@gobblefour}[4]{}
\@gobbleseven88 \providecommand*\@gobbleseven}[7]{}
389 %

```


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 `ab` does not produce the same typographical result as `ab`.²⁸ The `\l@wrapcs@ifnotemptybox` does the same, but take as argument a control sequence name, so without backslash.

```

390 \newcommand{\l@wrap@ifnotemptybox}[2]{%
391   \setbox0=\hbox{#2}%
392   \ifdim\wd0=\z@ \else%
393     #1{#2}%
394   \fi%
395 }%
396 \newcommand{\l@wrapcs@ifnotemptybox}[2]{%
397   \l@wrap@ifnotemptybox{\csname #1\endcsname}{#2}%
398 }%
399 %

```

II.10 Miscellaneous commands

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

```

400 \ifldefinal
401   \global\notbool{parapparatus@}{\newcommand*}{\newcommand}{\showlemma
402   }[1]{#1}%
403 \else
404   \global\notbool{parapparatus@}{\newcommand*}{\newcommand}{\showlemma
405   }[1]{\textbf{#1}}%
406 \fi
407 %
408 %
409 %

```

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

Initialize it as `\empty`.

```

407 \let\linenumberlist=\empty
408
409 %

```

`\@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).

```

410 \newcount\@l@tempcnta \newcount\@l@tempcntb
411 %

```

²⁸Compare the distance between letters in the first case (ab) and in the second one (ab).

II.11 Prepare reledpar

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

`\ifl@dpaging`

`\ifl@dprintingpages`

`\ifl@dprintingcolumns`

`\ifpst@rtedL`

`\l@dnumpstartsL`

```

412 \newif\ifl@dpairing
413 \newif\ifl@dpaging%
414 \newif\ifl@dprintingpages%
415 \newif\ifl@dprintingcolumns%
416 \newif\ifpst@rtedL
417 \newcount\l@dnumpstartsL
418 %

```

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

```

419 \newif\ifledRcol
420 \newif\ifledRcol@
421 %

```

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

```

422 \newif\ifnumberingR
423 %

```

`\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`.

```

424 \newif\ifafterendnumberingR%
425 %

```

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

```

426 \newif\ifXnote@%
427 %

```

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` is not used.

`\ifindtl@notenumber`

```

428 \providebool{indtl@innote}%
429 \providebool{indtl@notenumber}%
430 %

```

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

`ifinstanzaR`

```

431 \newif\ifistanzaL%
432 \newif\ifistanzaR%
433 %

```

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

```

434 \newif\ifistanzaL%
435 \newif\ifistanzaR%
436 %

```

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.

```

437 \newcount\section@num
438 \section@num=0
439 \let\extensionchars=\empty
440 %

```

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

```

441 \newif\ifnumbering
442 %

```

`\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.

```

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

```

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.

```

471 \else
472   \begingroup
473   \global\@afterindenttrue%In order to reestablish normal feature if the \
474   \initnumbering@quote
475   \ifwidthliketwocolumns%
476     \setwidthliketwocolumns%
477     \csuse{setpositionliketwocolumns@\columns@position}%
478   \fi%
479 \fi
480 \gdef\eled@sections@@{}%

```

```

481 \if@noeled@sec\else%
482 \makeatletter%
483 \InputIfFileExists%
484 {\l@auxdir\jobname.eledsec\the\section@num}%
485 {}%
486 {\led@warn@NoFile{\l@auxdir\jobname.eledsec\the\section@num}}%
487 \makeatother%
488 \immediate\openout\eled@sectioning@out=\l@auxdir\jobname.eledsec\the\
section@num\relax%
489 \fi%
490 }
491 \newcommand*{\initnumbering@reg}{%
492 \global\pst@rtedLfalse
493 \global\l@dnumpststartsL \z@
494 \global\absline@num \z@
495 \gdef\normal@page@break{}
496 \gdef\l@prev@pb{}
497 \gdef\l@prev@nopb{}
498 \global\line@num \z@
499 \global\subline@num \z@
500 \global\@lock \z@
501 \global\sub@lock \z@
502 \global\sublines@false
503 \global\let\next@page@num=\relax
504 \global\let\this@section@next@page@num=\relax%
505 \global\let\sub@change=\relax
506 \global\last@page@num=-10000%
507 \ifdefined\line@numR%
508 \line@numR=\z@%
509 \last@page@numR=\z@%
510 \fi%
511 \resetprevline@
512 \resetprevpage@num
513 \global\stopmsdata@inserted@true%
514 \global\let\@msdata@list\relax%
515 \global\csundef{\@msdata@\add@msd@c @data}%
516 }
517
518 %

```

\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.

```

519 \def\endnumbering{%
520 \ifnumbering
521 \global\numberingfalse
522 \normal@pars
523 \ifnum\l@dnumpststartsL=0%
524 \led@err@NumberingWithoutPstart%
525 \fi%
526 %

```

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`.

```

527 \global\page@num=\this@section@page@num%
528 \global\last@page@num=\this@section@last@page@num%
529 \global\let\next@page@num\this@section@next@page@num%
530 \ifl@dpairing
531 \global\pst@rtedLfalse
532 \else
533 \ifx\insertlines@list\empty\else
534 \global\noteschanged@true
535 \fi
536 \ifx\line@list\empty\else
537 \global\noteschanged@true
538 \fi
539 \fi
540 \ifnoteschanged@
541 \led@mess@NotesChanged
542 \fi
543 \else
544 \led@err@NumberingNotStarted
545 \fi
546 \autoparfalse
547 \if@noeled@sec\else%
548 \immediate\closeout\eled@sectioning@out%
549 \fi%
550 \ifl@dpairing\else
551 \global\l@dnumpstartsL=\z@%
552 \endgroup
553 \fi
554 }
555 %

```

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

```

556 \newcount\pausenumbering@page@num%
557 \newcommand{\pausenumbering}{%
558 \ifx\this@section@next@page@num\relax%
559 \global\pausenumbering@page@num=0%
560 \else%
561 \global\pausenumbering@page@num=\this@section@next@page@num%
562 \fi%
563 \ifautopar\global\autopar@pausettrue\fi%
564 \endnumbering\global\numberingtrue}
565 %

```

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

²⁹Peter Wilson's thanks to Wayne Sullivan, who suggested the idea behind these macros.

invoked. The boolean `\ifresumenummering@start` is set to true. That allows us to avoid resetting the line number at the first line of `\resumenummering` if the lineation is by page. This boolean is set to false after the first action.

```

566 \newif\ifresumenummering@start%
567 \newcommand*{\resumenummering}{%
568   \ifnummering
569     \ifautopar@pause\autopar\fi
570     \global\pst@rtedLtrue
571     \global\advance\section@num \@ne
572     \global\resumenummering@starttrue%
573     \led@mess@SectionContinued{\the\section@num}%
574     \set@continuousnumberingforL%
575     \line@list@stuff{\jobname.\extensionchars\the\section@num}%
576     \ifcontinuousnumberingwithcolumns%
577       \unless\ifafterendnumberingR%
578         \unless\ifl@dpairing%
579           \ledRcoltrue%
580           \global\advance\section@numR \@ne%
581           \message{Section \the\section@numR R (continuoousnumbering)}%
582           \line@list@stuffR{\jobname.\extensionchars\the\section@numR R}%
583           \ledRcolfalse%
584         \fi%
585       \fi%
586     \fi%
587     \l@dend@stuff
588     \ifl@dpairing\else%
589       \begingroup%
590       \initnumbering@quote%
591       \ifwidthliketwocolumns%
592         \setwidthliketwocolumns%
593         \csuse{setpositionliketwocolumns@\columns@position}%
594       \fi%
595     \fi%
596   \else
597     \led@err@NumberingShouldHaveStarted
598     \endnumbering
599     \beginnumbering
600   \fi}
601
602
603 %

```

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

```

604 \newcommand{\set@continuousnumberingforL}{%
605   \ifcontinuousnumberingwithcolumns%
606     \ifl@dpairing%
607       \unless\ifl@dpaging%
608         \global\c@pstartL=\c@pstart%
609       \fi%
610     \fi%
611   \fi%
612 }%

```

613 %

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 \LaTeX 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.

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

`\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 .

```
618 \newcommand*{\list@clear}[1]{%
619   \global\let#1=\empty%
620 }
621 %
```

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

```
622 \newtoks\led@toksa \newtoks\led@toksb
623 \global\led@toksa={\}
624 \long\def\xright@appenditem#1\to#2{%
625   \global\led@toksb=\expandafter{#2}%
626   \xdef#2{\the\led@toksb\the\led@toksa\expandafter{#1}}%
627   \global\led@toksb={}}
628 %
```

`\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`.


```

629 \long\def\xleft@appenditem#1\to#2{%
630   \global\led@toksb=\expandafter{#2}%
631   \xdef#2{\the\led@toksa\expandafter{#1}\the\led@toksb}%
632   \global\led@toksb={}}
633 %

```

\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.

```

634 \def\gl@p#1\to#2{\expandafter\gl@poff#1\gl@poff#1#2}
635 \long\def\gl@poff\#1#2\gl@poff#3#4{\gdef#4{#1}\gdef#3{#2}}
636
637 %

```

V Line counting

V.1 Choosing the system of lineation

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

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

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

`\ifbypage@`

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

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

`\bypage@false`

```

638 \newif\ifbypage@
639 \newif\ifbypstart@
640 %

```

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@R41 \newif\ifbypage@R
\ifbypstart@R42 \newif\ifbypstart@R
643 %

```

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

```

644 \newcommand*{\lineation}[1]{
645 %

```

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

```

646 \ifnumbering
647   \led@err@LineationInNumbered
648 \else
649 %

```

If the argument is page.

```

650 \def\@tempa{#1}\def\@tempb{page}%
651 \ifx\@tempa\@tempb
652   \global\bypage@true
653   \global\bystart@false
654   \unless\ifnocritical@%
655     \Xpstart[] [false]%
656   \fi%
657 %

```

If the argument is pstart.

```

658 \else
659   \def\@tempb{pstart}%
660   \ifx\@tempa\@tempb
661     \global\bypage@false
662     \global\bystart@true
663     \unless\ifnocritical@%
664       \Xpstart%
665     \fi%
666 %

```

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

```

667 \else
668   \def\@tempb{section}
669   \ifx\@tempa\@tempb
670     \global\bypage@false
671     \global\bystart@false
672     \unless\ifnocritical@%
673       \Xpstart[] [false]%
674     \fi%
675 %

```

In other case, it is an error.

```

676 \else
677   \led@warn@BadLineation
678 \fi
679 \fi
680 \fi
681 \fi}}
682 %

```

V.2 Line number margin

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

```

683 \newcount\line@margin%
684 \newcount\line@margin@columns%Only for parallel typesetting
685 \line@margin@columns=\m@ne%
686
687 \newcommand*{\linenummargin}[1]{\{

```

```

688 \l@dgetline@margin{#1}%
689 \ifnum\l@dtempcntb>\m@ne
690 \ifledRcol
691 \global\line@marginR=\l@dtempcntb
692 \led@warn@setting@in@rightside{\linenummargin}%
693 \else
694 \global\line@margin=\l@dtempcntb
695 \fi
696 \fi}}
697
698 \newcommand*{\l@dgetline@margin}[1]{%
699 \def\@tempa{#1}\def\@tempb{left}%
700 \ifx\@tempa\@tempb
701 \l@dtempcntb \z@
702 \else
703 \def\@tempb{right}%
704 \ifx\@tempa\@tempb
705 \l@dtempcntb \@ne
706 \else
707 \def\@tempb{outer}%
708 \ifx\@tempa\@tempb
709 \l@dtempcntb \tw@
710 \else
711 \def\@tempb{inner}%
712 \ifx\@tempa\@tempb
713 \l@dtempcntb \thr@@
714 \else
715 \led@warn@BadLinenummargin
716 \l@dtempcntb \m@ne
717 \fi
718 \fi
719 \fi
720 \fi}
721
722 %

```

V.3 Line number initialization and increment

`\c@firstlinenum` The following counters tell `reledmac` which lines should be printed with line numbers. `firstlinenum` is the number of the first line in each section that gets a number; `linenumincrement` is the difference between successive numbered lines. The initial values of these counters produce labels on lines 5, 10, 15, etc. `linenumincrement` must be at least 1.

```

723 \newcounter{firstlinenum}
724 \setcounter{firstlinenum}{5}
725 \newcounter{linenumincrement}
726 \setcounter{linenumincrement}{5}
727 %

```

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

```

728 \newcounter{firstsublinenum}
729 \setcounter{firstsublinenum}{5}
730 \newcounter{sublinenumincrement}
731 \setcounter{sublinenumincrement}{5}
732
733 %

```

These macros can be used to set the corresponding counters.

```

\firstlinenum
\linenumincrement
\firstsublinenum
\sublinenumincrement
734
735 \newcommand*{\firstlinenum}[1]{%
736   \ifledRcol%
737     \setcounter{firstlinenumR}{#1}%
738     \led@warn@setting@in@rightside{\firstlinenum}%
739   \else%
740     \setcounter{firstlinenum}{#1}%
741   \fi%
742 }
743 \newcommand*{\linenumincrement}[1]{%
744   \ifledRcol%
745     \setcounter{linenumincrementR}{#1}%
746     \led@warn@setting@in@rightside{\linenumincrement}%
747   \else%
748     \setcounter{linenumincrement}{#1}%
749   \fi%
750 }
751 \newcommand*{\firstsublinenum}[1]{%
752   \ifledRcol%
753     \setcounter{firstsublinenumR}{#1}%
754     \led@warn@setting@in@rightside{\firstsublinenum}%
755   \else%
756     \setcounter{firstsublinenum}{#1}%
757   \fi%
758 }
759 \newcommand*{\sublinenumincrement}[1]{%
760   \ifledRcol%
761     \setcounter{sublinenumincrementR}{#1}%
762     \led@warn@setting@in@rightside{\sublinenumincrement}%
763   \else%
764     \setcounter{sublinenumincrement}{#1}%
765   \fi%
766 }
767
768 %

```

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.

```

769 \newcount\lock@disp
770 \newcommand{\lockdisp}[1]{\{%
771   \l@getlock@disp{#1}%
772   \ifnum\@l@tempcntb>\m@ne
773     \global\lock@disp=\@l@tempcntb
774   \else
775     \led@warn@BadLockdisp
776   \fi}}
777 \newcommand*{\l@getlock@disp}[1]{
778   \def\@tempa{#1}\def\@tempb{first}%
779   \ifx\@tempa\@tempb
780     \@l@tempcntb \z@
781   \else
782     \def\@tempb{last}%
783     \ifx\@tempa\@tempb
784       \@l@tempcntb \@ne
785     \else
786       \def\@tempb{all}%
787       \ifx\@tempa\@tempb
788         \@l@tempcntb \tw@
789       \else
790         \@l@tempcntb \m@ne
791       \fi
792     \fi
793   \fi}
794
795 %

```

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

```

796 \newcount\sublock@disp
797 \newcommand{\sublockdisp}[1]{\{%
798   \l@getlock@disp{#1}%
799   \ifnum\@l@tempcntb>\m@ne
800     \global\sublock@disp=\@l@tempcntb
801   \else
802     \led@warn@BadSublockdisp
803   \fi}}
804
805 %

```

V.5 Line number style

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

```

806 \newcommand*{\linenumberstyle}[1]{\%

```

```

807 \def\linenumrep##1{\@nameuse{@#1}{##1}}
808 \newcommand*{\sublinenumberstyle}[1]{%
809 \def\sublinenumrep##1{\@nameuse{@#1}{##1}}
810 %
811
812 Initialise the number styles to arabic.
813
814 \linenumberstyle{arabic}
815 \let\linenumr@p\linenumrep
816 \sublinenumberstyle{arabic}
817 \let\sublinenumr@p\sublinenumrep
818 %

```

V.6 Line number printing

`\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.

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

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

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

```

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

```

```

840 \fi%
841 \unless\iflinenumannotationotherside%
842 \ifdefstring{\Xlinenumannotationposition@side}{after}{%
843 \l@wrap@ifnotemptybox{\Xwraplinenumannotation@side}{%
844 \csuse{annot@the\absline@num @the\section@num}%
845 }%
846 }%
847 \fi%
848 \egroup%
849 }%
850
851 \newcommand*{\leftlinenum}{%
852 \ledlinenum%
853 \kern\linenumsep%
854 }%
855 \newcommand*{\rightlinenum}{%
856 \kern\linenumsep%
857 \ledlinenum%
858 }%
859
860 %

```

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

```

861 \newcommand*{\leftlinenumannotation}{%
862 \l@wrap@ifnotemptybox{\Xwraplinenumannotation@side}{%
863 \csuse{annot@the\absline@num @the\section@num}%
864 }%
865 \kern\linenumsep%
866 }%
867 \newcommand*{\rightlinenumannotation}{%
868 \kern\linenumsep%
869 \l@wrap@ifnotemptybox{\Xwraplinenumannotation@side}{%
870 \csuse{annot@the\absline@num @the\section@num}%
871 }%
872 }%
873 %

```

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 \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`.

```
874 \newcount\line@num
875 %
```

\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.

```
876 \newcount\subline@num
877 %
```

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

\sublines@true

\sublines@false

```
878 \newif\ifsublines@
879 \newif\ifsublines@R%
880 %
```

\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.

```
881 \newcount\absline@num
882 %
```

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.

\sub@lock

```
883 \newcount\@lock
884 \newcount\sub@lock
885 %
```


V.9 Line number associated to lemma

```
\line@list
\insertlines@list
\actionlines@list
\actions@list
\annot@list
```

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

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

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

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

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

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

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

Any number greater than -1000 is a page-start action: the line number associated with it is the first line on a page, and the action number is the page number. (The cutoff of -1000 is chosen because negative page-number values are used by some

macro packages; we assume that page-number values less than -1000 are not common.) Page-start action codes are added to the list by the `\page@action` macro, which is (indirectly) triggered by the workings of the `\page@start` macro; that macro should always be called in the output routine, just before the page contents are assembled. Eledmac calls it in `\pagecontents`.

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

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

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

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

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

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

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

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

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

Here are the commands to create these lists:

```

886 \list@create{\line@list}
887 \list@create{\insertlines@list}
888 \list@create{\actionlines@list}
889 \list@create{\actions@list}
890 \list@create{\annot@list}%
891 %

```

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

```

892 \newcount\page@num
893 \newcount\endpage@num
894 \newcount\endline@num
895 \newcount\endsubline@num
896 %

```

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

```

897 \newcount\this@section@page@num%
898 %

```

`\ifnoteschanged@` If the number of the footnotes in a section is different from what it was during the last
`\noteschanged@true` run, or if this is the very first time you've run \LaTeX , on this file, the information from
`\noteschanged@false` 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.

```

899 \newif\ifnoteschanged@
900 %

```

`\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@ 901 \newcommand*\resetprevline@{%
902 \def\do##1{\global\csundef{prevline##1}}%
903 \dolistloop{\@series}%
904 }
905 %

```

`\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@ 906 \newcommand*\resetprevpage@num{%
907 \def\do##1{%
908 \ifcsdef{prevpage##1@num}{%
909 \global\csname prevpage##1@num\endcsname=\z@%
910 \global\csname prevpage##1@numR\endcsname=\z@%
911 }%

```

```

912     {}%
913     \ifcsdef{##1prevpage@num}{%
914         \global\csname ##1prevpage@num\endcsname=\z@%
915         \global\csname ##1prevpage@numR\endcsname=\z@%
916     }%
917     {}%
918     }%
919     \dolistloop{\@series}%
920 }
921 %

```

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.

```

922 \newread\@inputcheck
923 \newcommand*{\read@linelist}[1]{%
924     \ifledRcol%
925         \list@clearing@regR%
926     \else%
927         \list@clearing@reg%
928     \fi%
929 %

```

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).

```

930     \list@clear{\maxlinesinpar@list}
931 %

```

Now get the file and interpret it. When the file is there we start a new group and make some special definitions we will need to process it. It is a sequence of \TeX commands, but they require a few special settings. We make `[` and `]` become grouping characters: they are used that way in the line-list file, because we need to write them out one at a time rather than in balanced pairs, and it is easier to just use something other than real braces. `@` must become a letter, since this is run in the ordinary \LaTeX context. We ignore carriage returns, since if we are in horizontal mode they can get interpreted as spaces to be printed.

Our line, page, and line-locking counters were already zeroed by `\line@list@stuff` if this is being called from within `\beginnumbering`; sub-lineation will be turned off as well in that case. On the other hand, if this is being called from `\resumenumbers`, those things should still have the values they had when `\pausenumbering` was executed.

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

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

```

932 \get@linelistfile{#1}%
933 \@stopmsd%Security if last \endms{} is forgotten
934 \unless\ifledRcol%Get the last line of the last page
935     \csnumgdef{\lastabsline@forpage@the\page@num}{\the\absline@num}%
936     \csnumgdef{\lastline@forpage@the\page@num}{\the\line@num}%
937     \csnumgdef{\lastsubline@forpage@the\page@num}{\the\subline@num}%

```

```

938 \else%
939 \csnumgdef{@lastabsline@forpageR@the\page@numR}{\the\absline@numR}%
940 \csnumgdef{@lastline@forpageR@the\page@numR}{\the\line@numR}%
941 \csnumgdef{@lastsubline@forpageR@the\page@numR}{\the\subline@numR}%
942 \fi%
943 \endgroup
944 %

```

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.

```

945 \ifledRcol
946 \global\page@numR=\m@ne
947 \ifx\actionlines@listR\empty
948 \gdef\next@actionlineR{1000000}%
949 \else
950 \gl@p\actionlines@listR\to\next@actionlineR
951 \gl@p\actions@listR\to\next@actionR
952 \fi
953 \else
954 \global\page@num=\m@ne
955 \ifx\actionlines@list\empty
956 \gdef\next@actionline{1000000}%
957 \else
958 \gl@p\actionlines@list\to\next@actionline
959 \gl@p\actions@list\to\next@action
960 \fi
961 \fi
962 }
963 %

```

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

```

964 \newcommand*{\list@clearing@reg}{%
965 \list@clear{\line@list}%
966 \list@clear{\insertlines@list}%
967 \list@clear{\actionlines@list}%
968 \list@clear{\actions@list}%
969 \list@clear{\linesinpar@listL}%
970 \list@clear{\linesonpage@listL}%
971 }%
972 %

```

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

```

973 \newcommand*{\get@linelistfile}[1]{%
974 \InputIfFileExists{\l@auxdir#1}{%
975 \global\noteschanged@false
976 \begingroup
977 \catcode`\[=1 \catcode`\]=2
978 \makeatletter \catcode`\^^M=9}{%
979 \led@warn@NoFile{\l@auxdir#1}%

```

```

980 \global\noteschanged@true
981 \begingroup}%
982 }
983
984 %

```

This version of `\read@linelist` creates list macros containing data for the entire section, so they could get rather large. It would be no more difficult to read the line-list file incrementally rather than all at once: we could read, at the start of each paragraph, only the commands relating to that paragraph. But this would require that we have two line-lists open at once, one for reading, one for writing, and on systems without version numbers we would have to do some file renaming outside of \TeX for that to work. We have retained this slower approach to avoid that sort of hacking about, but have provided the `\pausenumbers` and `\resumenumbers` macros to help you if you run into macro memory limitations (see 5.2.7 p. 19 above).

V.11 Commands within the line-list file

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

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

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

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

```

985 \newcommand{\line@list@version}[1]{%
986   \ifStrEq{#1}{\this@line@list@version}%
987   }{%
988     {\ifledRcol%
989       \led@warn@Obsolete{\jobname.\extensionchars\the\section@num}%
990       \else%
991       \led@warn@Obsolete{\jobname.\extensionchars\the\section@num}%
992       \fi%
993       \endinput%
994     }%
995   }%
996   %

```

`\@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`.

```

997
998 \newcommand*{\@nl}[2]{%
999   \@page{#1}%
1000   \ifledRcol%
1001     \@nl@regR%
1002   \else%
1003     \@nl@reg%
1004   \fi%
1005 }
1006 \newcommand*{\@nl@reg}{%
1007   \ifx\l@dchset@num\relax \else
1008     \advance\absline@num \@ne
1009     \csgdef{l@dchset@num@the\absline@num}{}%To remember this line have
1010     been marked by a \setlinenum
1011     \set@line@action
1012     \let\l@dchset@num=\relax
1013     \advance\absline@num \m@ne
1014     \advance\line@num \m@ne
1015   \fi
1016 }

```

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

```

1016 \reset@current@annot%
1017 \advance\absline@num \@ne
1018   \ifx\next@page@num\relax \else
1019     \page@action
1020     \let\next@page@num=\relax
1021   \fi
1022   \ifx\sub@change\relax \else
1023     \ifnum\sub@change>\z@
1024       \sublines@true
1025     \else
1026       \sublines@false
1027     \fi
1028     \sub@action
1029     \let\sub@change=\relax
1030   \fi
1031 }

```

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

```

1032 \ifcase\@lock

```

```

1033         \or
1034         \@lock \tw@
1035         \or \or
1036         \@lock \z@
1037     \fi
1038     \ifcase\sub@lock
1039         \or
1040         \sub@lock \tw@
1041         \or \or
1042         \sub@lock \z@
1043     \fi
1044 %

```

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

```

1045     \ifsublines@
1046         \ifnum\sub@lock<\tw@
1047             \advance\subline@num \@ne
1048         \fi
1049     \else
1050         \ifnum\@lock<\tw@
1051             \advance\line@num \@ne \subline@num \z@
1052         \fi
1053     \fi}
1054 %
1055 %

```

`\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
1056 \newcount\last@page@num
1057 \last@page@num=-10000
1058
1059 \newcount\this@section@last@page@num%
1060 \this@section@last@page@num=-10000%
1061
1062 \newcommand*{\@page}[1]{%
1063     \ifledRcol
1064         \ifnum #1=\last@page@numR
1065         \else
1066             \csnumgdef{\lastabsline@forpageR@the\page@numR}{\the\absline@numR}%
1067             \csnumgdef{\lastline@forpageR@the\page@numR}{\the\line@numR}%
1068             \csnumgdef{\lastsubline@forpageR@the\page@numR}{\the\subline@numR}%
1069             \ifbypage@R
1070                 \ifx\l@dchset@num\relax%Not resetting if preceded by a \setlinenum
1071                     \ifboolexpr{%
1072                         (bool{continuousnumberingwithcolumns} or bool{
1073 resumenumberingR@start})}%
1074                         and test {\ifnumequal{\last@page@numR}{-10000}}}%
1075                     }%
1076                     {%
1077                         \line@numR \z@%
1078                         \subline@numR \z@%
1079                         \global\csdef{reset@lineR\the\numexpr\absline@numR+1\relax @\
the\section@numR}{}%

```



```

1080     }%
1081     \fi%
1082     \fi
1083     \page@numR=#1%
1084     \global\this@section@page@numR=#1%
1085     \global\last@page@numR=#1%
1086     \global\this@section@last@page@numR=#1%
1087     \def\next@page@numR{#1}%
1088     \gdef\this@section@next@page@numR{#1}%
1089     \fi
1090     \else
1091     \ifnum #1=\last@page@num
1092     \else
1093     \csnumgdef{\lastabsline@forpage@the\page@num}{\the\absline@num}%
1094     \csnumgdef{\lastline@forpage@the\page@num}{\the\line@num}%
1095     \csnumgdef{\lastsubline@forpage@the\page@num}{\the\subline@num}%
1096     \ifbypage@
1097     \ifx\ldchset@num\relax%Not resetting if preceded by a \setlinenum
1098     \line@num \z@%
1099     \subline@num \z@%
1100     \global\csdef{reset@line\the\numexpr\absline@num+1\relax @the\
section@num}{}%
1101     \fi%
1102     \fi
1103     \page@num=#1%
1104     \global\this@section@page@num=#1%
1105     \last@page@num=#1%
1106     \global\this@section@last@page@num=#1%
1107     \def\next@page@num{#1}%
1108 %

```

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.

```

1109     \gdef\this@section@next@page@num{#1}%
1110     \listxadd{\normal@page@break}{\the\absline@num}
1111     \fi
1112 \fi}
1113 %

```

\@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.

```

1114 \@lopR \newcommand*{\@pend}[1]{}
1115 \newcommand*{\@pendR}[1]{}
1116 \newcommand*{\@lopL}[1]{}
1117 \newcommand*{\@lopR}[1]{}
1118
1119 %

```

\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.

```

1120 \newcommand*{\sub@on}{%
1121   \ifboolexpr{%
1122     bool{sublines@}%
1123     or%
1124     (bool{sublines@R} and bool{ledRcol})}%
1125   }{%
1126     \let\sub@change=\relax
1127   }{%
1128     \def\sub@change{1}%
1129   }%
1130 }%
1131 \newcommand*{\sub@off}{%
1132   \ifboolexpr{%
1133     bool{sublines@}%
1134     or%
1135     (bool{sublines@R} and bool{ledRcol})}%
1136   }{%
1137     \def\sub@change{-1}%
1138   }{%
1139     \let\sub@change=\relax
1140   }%
1141 }%
1142 %
1143 %

```

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

```

1144 \newcommand*{\@adv}[1]{%
1145   \ifsublines@
1146     \ifledRcol
1147       \advance\subline@numR by #1\relax
1148       \ifnum\subline@numR<\z@
1149         \led@warn@BadAdvancelineSubline
1150         \subline@numR \z@
1151       \fi
1152     \else
1153       \advance\subline@num by #1\relax
1154       \ifnum\subline@num<\z@
1155         \led@warn@BadAdvancelineSubline
1156         \subline@num \z@
1157       \fi
1158     \fi
1159   \fi
1160 \else
1161   \ifledRcol
1162     \advance\line@numR by #1\relax
1163     \ifnum\line@numR<\z@
1164       \led@warn@BadAdvancelineLine
1165       \line@numR \z@
1166     \fi
1167   \else
1168     \advance\line@num by #1\relax
1169     \ifnum\line@num<\z@

```

```

1170     \led@warn@BadAdvancelineLine
1171     \line@num \z@
1172     \fi
1173   \fi
1174 \fi
1175 \set@line@action}
1176
1177 %

```

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

```

1178
1179 \newcommand*{\@set}[1]{%
1180   \ifledRcol
1181     \ifsublines@R%
1182       \subline@numR=#1\relax
1183     \else
1184       \line@numR=#1\relax
1185     \fi
1186     \set@line@action
1187   \else
1188     \ifsublines@
1189       \subline@num=#1\relax
1190     \else
1191       \line@num=#1\relax
1192     \fi
1193     \set@line@action
1194   \fi}
1195
1196 %

```

\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.

```

1197
1198 \newcommand*{\l@d@set}[1]{%
1199   \ifledRcol
1200     \line@numR=#1\relax
1201     \advance\line@numR \@ne
1202     \def\l@dchset@num{#1}
1203   \else
1204     \line@num=#1\relax
1205     \advance\line@num \@ne
1206     \def\l@dchset@num{#1}
1207   \fi}
1208 \let\l@dchset@num\relax
1209
1210 %

```

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

```

1211
1212 \newcommand*{\page@action}{%
1213   \ifledRcol
1214     \xright@appenditem{\the\absline@numR}\to\actionlines@listR
1215     \xright@appenditem{\next@page@numR}\to\actions@listR
1216   \else
1217     \xright@appenditem{\the\absline@num}\to\actionlines@list
1218     \xright@appenditem{\next@page@num}\to\actions@list
1219   \fi}
1220 %

```

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

```

1221
1222 \newcommand*{\set@line@action}{%
1223   \ifledRcol
1224     \xright@appenditem{\the\absline@numR}\to\actionlines@listR
1225     \ifsublines@R%
1226       \@l@tempcnta=-\subline@numR
1227     \else
1228       \@l@tempcnta=-\line@numR
1229     \fi
1230     \advance\@l@tempcnta by -5000\relax
1231     \xright@appenditem{\the\@l@tempcnta}\to\actions@listR
1232   \else
1233     \xright@appenditem{\the\absline@num}\to\actionlines@list
1234     \ifsublines@
1235       \@l@tempcnta=-\subline@num
1236     \else
1237       \@l@tempcnta=-\line@num
1238     \fi
1239     \advance\@l@tempcnta by -5000\relax
1240     \xright@appenditem{\the\@l@tempcnta}\to\actions@list
1241   \fi}
1242 %

```

\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.

```

1243
1244 \newcommand*{\sub@action}{%
1245   \ifledRcol
1246     \xright@appenditem{\the\absline@numR}\to\actionlines@listR
1247     \ifsublines@R%
1248       \xright@appenditem{-1001}\to\actions@listR
1249     \else
1250       \xright@appenditem{-1002}\to\actions@listR
1251     \fi
1252   \else
1253     \xright@appenditem{\the\absline@num}\to\actionlines@list
1254     \ifsublines@
1255       \xright@appenditem{-1001}\to\actions@list
1256     \else

```

```

1257 \xright@appenditem{-1002}\to\actions@list
1258 \fi
1259 \fi}
1260 %

```

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

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

```

1261 \newcommand*{\lock@on}{\futurelet\next\do@lockon}
1262
1263 \newcommand*{\do@lockon}{%
1264 \ifx\next\lock@off
1265 \global\let\lock@off=\skip@lockoff
1266 \else
1267 \ifledRcol
1268 \do@lockonR
1269 \else
1270 \do@lockonL
1271 \fi
1272 \fi}
1273
1274
1275 \newcommand*{\do@lockonL}{%
1276 \xright@appenditem{\the\absline@num}\to\actionlines@list
1277 \ifsublines@
1278 \xright@appenditem{-1005}\to\actions@list
1279 \ifnum\sub@lock=\z@
1280 \sub@lock \@ne
1281 \else
1282 \ifnum\sub@lock=\thr@@
1283 \sub@lock \@ne
1284 \fi
1285 \fi
1286 \else
1287 \xright@appenditem{-1003}\to\actions@list
1288 \ifnum\@lock=\z@
1289 \@lock \@ne
1290 \else
1291 \ifnum\@lock=\thr@@
1292 \@lock \@ne
1293 \fi
1294 \fi
1295 \fi}
1296
1297 %

```

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

\do@lockoff
\do@lockoffL \newcommand*{\do@lockoffL}{%
\skip@lockoff

```

1299 \xright@appenditem{\the\absline@num}\to\actionlines@list
1300 \ifsublines@
1301 \xright@appenditem{-1006}\to\actions@list
1302 \ifnum\sub@lock=\tw@
1303 \sub@lock \thr@@
1304 \else
1305 \sub@lock \z@
1306 \fi
1307 \else
1308 \xright@appenditem{-1004}\to\actions@list
1309 \ifnum\@lock=\tw@
1310 \@lock \thr@@
1311 \else
1312 \@lock \z@
1313 \fi
1314 \fi}
1315
1316 \newcommand*{\do@lockoff}{%
1317 \reset@current@annot%
1318 \ifledRcol
1319 \do@lockoffR
1320 \else
1321 \do@lockoffL
1322 \fi}
1323 \newcommand*{\skip@lockoff}{\global\let\lock@off=\do@lockoff}
1324 \global\let\lock@off=\do@lockoff
1325
1326 %

```

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

```

1327 \newcommand*{\n@num}{%
1328 \ifledRcol%
1329 \xright@appenditem{\the\absline@numR}\to\actionlines@listR
1330 \xright@appenditem{-1007}\to\actions@listR
1331 \else%
1332 \xright@appenditem{\the\absline@num}\to\actionlines@list%
1333 \xright@appenditem{-1007}\to\actions@list%
1334 \fi%
1335 }%
1336
1337 %

```

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

```

1338 \newcommand*{\n@num@stanza}{%
1339 \ifledRcol%
1340 \xright@appenditem{\the\absline@numR}\to\actionlines@listR%
1341 \xright@appenditem{-1008}\to\actions@listR%
1342 \else%
1343 \xright@appenditem{\the\absline@num}\to\actionlines@list%%
1344 \xright@appenditem{-1008}\to\actions@list%

```

```

1345 \fi%
1346 }
1347 %

```

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

```

\hidenumberingonleftpage
\hidenumberingonrightpage
\hidenumberingonleftpage
\hidenumberingonrightpage
1348 \newif\ifl@dhiddenumber
1349 \newcommand*{\hidenumbering}{%
1350 \ifl@dhiddenumber
1351 \write\linenum@outR{\string\hide@num}%
1352 \else%
1353 \write\linenum@out{\string\hide@num}%
1354 \fi%
1355 }%
1356 \newcommand*{\hide@num}{%
1357 \ifl@dhiddenumber
1358 \xright@appenditem{\the\absline@numR}\to\actionlines@listR%
1359 \xright@appenditem{-1009}\to\actions@listR%
1360 \else%
1361 \xright@appenditem{\the\absline@num}\to\actionlines@listR%
1362 \xright@appenditem{-1009}\to\actions@listR%
1363 \fi%
1364 }
1365 \newcommand*{\hidenumberingonleftpage}{%
1366 \ifl@dhiddenumber
1367 \write\linenum@outR{\string\hide@num@left}%
1368 \else%
1369 \write\linenum@out{\string\hide@num@left}%
1370 \fi%
1371 }%
1372
1373 \newcommand*{\hide@num@left}{%
1374 \ifl@dhiddenumber
1375 \ifodd\page@numR\else%
1376 \xright@appenditem{\the\absline@numR}\to\actionlines@listR%
1377 \xright@appenditem{-1009}\to\actions@listR%
1378 \fi%
1379 \else%
1380 \ifodd\page@num\else%
1381 \xright@appenditem{\the\absline@num}\to\actionlines@listR%
1382 \xright@appenditem{-1009}\to\actions@listR%
1383 \fi%
1384 \fi%
1385 }%
1386
1387 \newcommand*{\hidenumberingonrightpage}{%
1388 \ifl@dhiddenumber
1389 \write\linenum@outR{\string\hide@num@right}%
1390 \else%
1391 \write\linenum@out{\string\hide@num@right}%
1392 \fi%
1393 }%
1394

```

```

1395 \newcommand*{\hide@num@right}{%
1396   \ifledRcol%
1397     \ifodd\page@numR%
1398       \xright@appenditem{\the\absline@numR}\to\actionlines@listR%
1399       \xright@appenditem{-1009}\to\actions@listR%
1400     \fi%
1401   \else%
1402     \ifodd\page@num%
1403       \xright@appenditem{\the\absline@num}\to\actionlines@list%%
1404       \xright@appenditem{-1009}\to\actions@list%
1405     \fi%
1406   \fi%
1407 }%
1408
1409 %

```

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

\insert@count

- #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.

```

1410 \newcount\insert@count
1411 %

```

- #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.

```

1412 \newcommand*{\dummy@ref}[2]{#2}
1413 %

```

\@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.

```

1414 \newcommand*{\@ref}[2]{%
1415   \ifledRcol%
1416     \@ref@regR{#1}{#2}%
1417   \else%
1418     \@ref@reg{#1}{#2}%
1419   \fi%
1420 }%
1421 \newcommand*{\@ref@reg}[2]{%
1422   \global\insert@count=#1\relax
1423   \global\advance\@edtext@level by 1%
1424   \loop\ifnum\insert@count>\z@
1425     \xright@appenditem{\the\absline@num}\to\insertlines@list
1426     \global\advance\insert@count \m@ne
1427   \repeat
1428 %

```


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.

```

1429 \beginingroup
1430   \let\@ref=\dummy@ref
1431   \let\@lopL\@gobble
1432   \let\page@action=\relax
1433   \let\sub@action=\relax
1434   \let\set@line@action=\relax
1435   \let\@lab=\relax
1436   \let\@lemma=\relax%
1437   \let\@sw\@gobblethree%
1438   \let\store@annot@to@absline\@gobble%
1439   #2
1440   \global\endpage@num=\page@num
1441   \global\endline@num=\line@num
1442   \global\endsubline@num=\subline@num
1443   \global\let\endcurrent@annot=\current@annot%
1444 \endgroup
1445 %

```

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

```

1446 \xright@appenditem%
1447   {\the\page@num|\the\line@num|%
1448   \ifsublines@ \the\subline@num \else 0\fi}%
1449   \the\endpage@num|\the\endline@num|%
1450   \ifsublines@ \the\endsubline@num \else 0\fi}\to\line@list
1451 \xright@appenditem%
1452   {\current@annot|\endcurrent@annot}\to\annot@list%
1453 %

```

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

```

1454   \@ref@reg@parse{#2}%
1455 %

```

Decrease edtext level counter.

```

1456   \global\advance\@edtext@level by -1%
1457 }
1458 %

```

`\@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.

```

1459 \newcommand{\@ref@reg@parse}[1]{%
1460   \expandafter\list@create\expandafter{\csname sw@list@edtext@tmp@\the\
@edtext@level\endcsname}%
1461   \providebool{lemmacommand@\the\@edtext@level}%
1462   \boolfalse{lemmacommand@\the\@edtext@level}%
1463 %

```

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

```
1464 #1%
1465 %
```

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.

```
1466 \ifnum\@edtext@level>0%
1467 \def\create@this@edtext@level{\expandafter\list@create\expandafter{\
csname sw@list@edtext@the\@edtext@level\endcsname}}%
1468 \ifcsundef{sw@list@edtext@the\@edtext@level}{\create@this@edtext@level
}{}%
1469 \letcs{\@tmp}{sw@list@edtext@the\@edtext@level}%
1470 \letcs{\@tmpp}{sw@list@edtext@tmp@the\@edtext@level}
1471 \xright@appenditem{\expandonce\@tmpp}{to\@tmp%
1472 \global\cslet{sw@list@edtext@the\@edtext@level}{\@tmp}%
1473 \fi%
1474 %
1475 }
1476 %
1477 %
```

`\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.

```
1478 \newcommand{\@ref@later}[1]{%
1479 \global\advance\@edtext@level by \@ne%
1480 \ifledRcol%
1481 \@ref@reg@parseR{#1}%
1482 \else%
1483 \@ref@reg@parse{#1}%
1484 \fi%
1485 \global\advance\@edtext@level by -\@ne%
1486 }%
1487 %
```

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`.

```
1488 \newwrite\linenum@out
1489 %
```

`\iffirst@linenum@out@` Once any file is opened on this stream, we keep it open forever, or else switch to another
`\first@linenum@out@true` file that we keep open. The reason is that we want the output routine to write the page
`\first@linenum@out@false` number for every page to this file; otherwise we would have to write it at the start of every line. But it is not very easy for the output routine to tell whether an output stream

is open or not. There is no way to test the status of a particular output stream directly, and the asynchronous nature of output routines makes the status hard to determine by other means.

We can manage pretty well by means of the `\iffirst@linenum@out@` flag; its inelegant name suggests the nature of the problem that made its creation necessary. It is set to be true before any `\linenum@out` file is opened. When such a file is opened for the first time, it is done using `\immediate`, so that it will at once be safe for the output routine to write to it; we then set this flag to false.

```
1490 \newif\iffirst@linenum@out@
1491 \first@linenum@out@true
1492 %
```

`\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.

```
1493 \newcommand{\this@line@list@version}{8}%
1494 %
```

`\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.

```
1495 \let\next@line@list@stuff\relax%
1496 \newcommand*{\line@list@stuff}[1]{%
1497 %
```

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

```
1498 \global\newtoggle{notfirstrun@#1}%
1499 \IfFileExists{\l@auxdir#1}%
1500 {\global\toggletrue{notfirstrun@#1}}%
1501 {\global\togglefalse{notfirstrun@#1}}%
1502 %
```

A internal hook (not used yet).

```
1503 \next@line@list@stuff%
1504 \global\let\next@line@list@stuff\relax%
1505 %
```

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

```
1506 \read@linelist{#1}%
1507 %
```

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 .

```

1508 \iffirst@linenum@out@
1509 \global\first@linenum@out@false%
1510 \immediate\openout\linenum@out=\l@auxdir#1\relax%
1511 \immediate\write\linenum@out{\string\line@list@version{\
this@line@list@version}}}%
1512 \ifl@dpaging%
1513 \immediate\write\linenum@out{\string\@par@sync@option{\
@par@this@sync@option}}}%
1514 \fi%
1515 \else
1516 %

```

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.

```

1517 \if@minipage%
1518 \leavevmode%
1519 \fi%
1520 \closeout\linenum@out%
1521 \openout\linenum@out=\l@auxdir#1\relax%
1522 \write\linenum@out{\string\line@list@version{\this@line@list@version}}
%
1523 \ifl@dpaging%
1524 \write\linenum@out{\string\@par@sync@option{\@par@this@sync@option}}
%
1525 \fi%
1526 \fi%
1527 }%
1528 %
1529 %

```

`\new@line`
`\set@this@c@page`

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 `\sameparallelpagnumber` 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.

```

1530 \newcommand*{\new@line}{%
1531 \ifnumberline%
1532 \IfStrEq{\led@pb@setting}{after}%
1533 {\xifinlist{\the\absline@num}{\l@prev@nopb}%
1534 {\xifinlist{\the\absline@num}{\normal@page@break}%
1535 {\numgdef{\@next@page}{\c@par@page+\@ne}%

```

```

1536 \write\linenum@out{\string\@nl[\@next@page][\@next@page]}}%
1537 }%
1538 {\write\linenum@out{\string\@nl[\the\@this@c@page][\thepage]}}%
1539 }%
1540 {\write\linenum@out{\string\@nl[\the\@this@c@page][\thepage]}}}%
1541 {}%
1542 \IfStrEq{\led@pb@setting}{before}%
1543 {\numdef{\next@absline}{\the\absline@num+\@ne}%
1544 \xifinlist{\next@absline}{\l@prev@nopb}%
1545 {\xifinlist{\the\absline@num}{\normal@page@break}%
1546 {\numgdef{\nc@page}{\c@par@page+\@ne}%
1547 \write\linenum@out{\string\@nl[\nc@page][\nc@page]}}%
1548 }%
1549 {\write\linenum@out{\string\@nl[\the\@this@c@page][\thepage]}}%
1550 }%
1551 {\write\linenum@out{\string\@nl[\the\@this@c@page][\thepage]}}%
1552 }%
1553 {}%
1554 \IfStrEqCase{\led@pb@setting}%
1555 {%
1556 {before}{\relax}%
1557 {after}{\relax}%
1558 }%
1559 \write\linenum@out{\string\@nl[\the\@this@c@page][\thepage]}}%
1560 ]%
1561 \fi%
1562 }
1563
1564 \newcount\@this@c@page%%
1565 \newcommand{\set@this@c@page}{%
1566 \ifboolexpr{%
1567 bool{sameparallelpagenumber}%
1568 or bool{prevpgnotnumbered}%
1569 }%
1570 {\global\@this@c@page=\c@par@page}%
1571 {\global\@this@c@page=\c@page}%
1572 }%
1573 %

```

`\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.

```

1574
1575 \newcommand*{\flag@start}{%
1576 \ifledRcol%
1577 \edef\next{\write\linenum@outR{%
1578 \string\@ref[\the\insert@countR] []}}%
1579 \else%
1580 \edef\next{\write\linenum@out{%
1581 \string\@ref[\the\insert@count] []}}%
1582 \fi%
1583 \next%
1584 }%

```

```

1585
1586 \newcommand*{\flag@end}{%
1587   \ifledRcol%
1588     \write\linenum@outR{]}%
1589   \else%
1590     \write\linenum@out{]}%
1591   \fi%
1592 }%
1593
1594
1595 %

```

`\flag@start@RTL` `\flag@end@RTL` With Xe_{La}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.³⁰

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

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

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

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

```

1596 \newcommand{\flag@start@RTL}{%
1597   \edlabel{edtext:start:\csuse{thisedtext@the\@edtext@level}}%
1598   \IfStrEq{\xabslineref{edtext:start:\csuse{thisedtext@the\@edtext@level}}}{000}%
1599   }%
1600   {}%
1601   {%
1602     \ifnumequal%
1603       {\xabslineref{edtext:start:\csuse{thisedtext@the\@edtext@level}}}%
1604       {\xabslineref{edtext:end:\csuse{thisedtext@the\@edtext@level}}}%
1605       {\flag@end}%
1606       {\flag@start}%
1607     }%
1608   }%
1609
1610 \newcommand{\flag@end@RTL}{%
1611   \edlabel{edtext:end:\csuse{thisedtext@the\@edtext@level}}%

```

³⁰This problem is caused by the way Xe_{La}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.

```

1612 \IfStrEq{\xabslineref{edtext:start:\csuse{thisedtext@the\@edtext@level
}}}%
1613 {000}%
1614 }%
1615 {%
1616 \ifnumequal%
1617 {\xabslineref{edtext:start:\csuse{thisedtext@the\@edtext@level}}}%
1618 {\xabslineref{edtext:end:\csuse{thisedtext@the\@edtext@level}}}%
1619 {\flag@start}%
1620 {\flag@end}%
1621 }%
1622 }%
1623 %

```

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

\flag@end@later

```

1624 \newcommand*{\flag@start@later}{%
1625 \ifledRcol%
1626 \write\linenum@outR{\string\@ref@later[]}%
1627 \else%
1628 \write\linenum@out{\string\@ref@later[]}%
1629 \fi%
1630 }%
1631 \newcommand{\flag@end@later}{%
1632 \ifledRcol%
1633 \write\linenum@outR{[]}%
1634 \else%
1635 \write\linenum@out{[]}%
1636 \fi%
1637 }
1638 %

```

\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.

```

1639
1640
1641 \newcommand*{\startsub}{%
1642 \dimen0\lastskip%
1643 \ifdim\dimen0>0pt%
1644 \unskip%
1645 \fi%

```

```

1646 \ifledRcol%
1647 \write\linenum@outR{\string\sub@on}%
1648 \else%
1649 \write\linenum@out{\string\sub@on}%
1650 \fi%
1651 \ifdim\dimen0>0pt%
1652 \hskip\dimen0%
1653 \fi%
1654 }%
1655
1656 \def\endsub{%
1657 \dimen0\lastskip%
1658 \ifdim\dimen0>0pt%
1659 \unskip%
1660 \fi%
1661 \ifledRcol%
1662 \write\linenum@outR{\string\sub@off}%
1663 \else%
1664 \write\linenum@out{\string\sub@off}%
1665 \fi%
1666 \ifdim\dimen0>0pt%
1667 \hskip\dimen0%
1668 \fi%
1669 }%
1670
1671 %

```

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

```

1672 \newcommand*{\advanceline}[1]{\leavevmode%
1673 \ifledRcol \write\linenum@outR{\string\@adv[#1]}%
1674 \else \write\linenum@out{\string\@adv[#1]}%
1675 \fi%
1676 }
1677 %

```

\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.

```

1678
1679 \newcommand*{\setline}[1]{%
1680 \leavevmode%
1681 \ifnum#1<\z@
1682 \led@warn@BadSetline%
1683 \else%
1684 \ifledRcol%
1685 \write\linenum@outR{\string\@set[#1]}%
1686 \else%
1687 \write\linenum@out{\string\@set[#1]}%
1688 \fi%
1689 \fi%
1690 }%
1691
1692 %

```


\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.

```

1693
1694 \newcommand*{\setlinenum}[1]{%
1695   \ifnum#1<\z@%
1696     \led@warn@BadSetlinenum%
1697   \else%
1698     \ifledRcol%
1699       \write\linenum@outR{\string\l@d@set[#1]}%
1700     \else%
1701       \write\linenum@out{\string\l@d@set[#1]}%
1702     \fi%
1703   \fi%
1704 }%
1705
1706 %

```

\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.

\endlock

```

1707
1708 \newcommand*{\startlock}{%
1709   \ifledRcol \write\linenum@outR{\string\lock@on}%
1710   \else      \write\linenum@out{\string\lock@on}%
1711   \fi}
1712 \def\endlock{%
1713   \ifledRcol \write\linenum@outR{\string\lock@off}%
1714   \else      \write\linenum@out{\string\lock@off}%
1715   \fi}
1716 %

```

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

\ifl@dskipversenumber

```

1717 \l@dskipnumbertrue \newif\ifl@dskipnumber
1718 \l@dskipnumberfalse \newif\ifl@dskipversenumber%
1719 \skipnumbering \newcommand*{\skipnumbering}{%
1720   \leavevmode%
1721   \ifledRcol%
1722     \ifistanza%
1723       \write\linenum@outR{\string\n@num@stanza}%
1724     \else%
1725       \write\linenum@outR{\string\n@num}%
1726     \fi%
1727     \advanceline{-1}%
1728   \else%
1729     \ifistanza%
1730       \write\linenum@out{\string\n@num@stanza}%
1731     \else%
1732       \write\linenum@out{\string\n@num}%
1733     \fi%
1734     \advanceline{-1}%
1735   \fi%

```

```

1736 }%
1737
1738 %

```

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. 153). 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.

```
1739 \list@create{\end@lemmas}
1740 %
```

`\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.

```
1741 \newcommand{\dummy@edtext}[2]{#1}
1742 %
```

`\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`.

```
1743 \newcommand{\dummy@edtext@showlemma}[2]{\showlemma{#1}}%
1744 %
```

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` `\morenoexpands` We need to turn off macro expansion for certain sorts of macros we are likely to see 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.³¹ 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

³¹Since “control sequences equivalent to characters are not expandable”—*The TeXbook*, answer to Exercise 20.14.

ask for a particular style at a specified size. The notes may well be in a smaller font, so the command should be expanded later, when the note's environment is in effect.

A second sort to turn off includes a few of the accent macros. Most are not a problem: an accent that is expanded to an `\accent` command may be harder to read but it works just the same. The ones that cause problems are: those that use alignments— \TeX seems to get confused about the difference between alignment parameters and macro parameters; those that use temporary control sequences; and those that look carefully at what the current font is.

(The `\copyright` macro defined in `PLAIN \TeX` has this sort of problem as well, but is not used enough to bother with. That macro, and any other that causes trouble, will get by all right if you put a `\protect` in front of it in your file.)

We also need to eliminate all `reledmac` macros like `\edlabel` and `\setline` that write things to auxiliary files: that writing should be done only once. And we make `\edtext` itself, if it appears within its own argument, do nothing but copy its first argument.

Finally, we execute `\morenoexpands`. The version of `\morenoexpands` defined here does nothing; but you may define a version of your own when you need to add more expansion suppressions as needed with your macros. That makes it possible to make such additions without needing to copy or modify the standard `reledmac` code. If you define your own `\morenoexpands`, you must be very careful about spaces: if the macro adds any spaces to the text when it runs, extra space will appear in the main text when `\edtext` is used.

The `\new@series` command also adds `\let\footnote(X)\@gobble` to the end of the `\no@expands` macro for the series $\langle X \rangle$.

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

```

1745 \newcommand*{\no@expands}{%
1746   \let\select@lemmafnt=0%
1747   \let\startsub=\relax \let\endsub=\relax
1748   \let\startlock=\relax \let\endlock=\relax
1749   \let\edlabel=\@gobble
1750   \let\setline=\@gobble \let\advanceline=\@gobble
1751   \let\sameword\sameword@inedtext%
1752   \let\edtext=dummy@edtext
1753   \let\edindex\dummy@edindex%
1754   \l@dtabnoexpands
1755   \l@noexpands@edgl%
1756   \let\linenumannotation=\@gobble%
1757   \morenoexpands}
1758 \let\morenoexpands=\relax
1759
1760 %

```

\@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.

```
1761 \newcommand{\@tag}{}
1762 %
```

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

```
1763 \newcount\@edtext@level%
1764 \@edtext@level=0%
1765 %
```

\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.

```
1766 \newif\if@edtext@secondarg@%
1767 %
```

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

```
1768 \newcounter{edtext}
1769 \renewcommand{\theedtext}{edtxt@the\c@edtext}%
1770 %
```

\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

```
1771 \newcount\edtext@notes%
1772 \newcount\edtext@notes@start%
1773 %
```

\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.

```
1774 \newif\if@noneed@notes%
1775 %
```

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

```
1776 \newcommand{\edtext}[2]{\leavevmode%
1777 %
```

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

```
1778 \ifnumberedpar%
1779 %
```

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

```
1780 \@check@edtext@args{#1}{#2}%
1781 %
```

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

```

1782 \global\advance\@edtext@level by 1%
1783 %

```

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

```

1784 \global\edtext@notes@start=\edtext@notes%
1785 %

```

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

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

```

1786 \stepcounter{edtext}%
1787 \csxdef{thisedtext@the\@edtext@level}{\theedtext}%
1788 %

```

By default, we do not use `\lemma`

```

1789 \global\@lemmacommand@false%
1790 %

```

```

1791 \begingroup%
1792 %

```

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`.

```

1793 \ifledRcol%
1794 \ifcvoid{sw@list@edtextR@the\@edtext@level}%
1795 {\global\let\sw@inthisedtext\empty}%
1796 {\expandafter\glp\csname sw@list@edtextR@the\
@edtext@level\endcsname\to\sw@inthisedtext}%
1797 \else%
1798 \ifcvoid{sw@list@edtext@the\@edtext@level}%
1799 {\global\let\sw@inthisedtext\empty}%
1800 {\expandafter\glp\csname sw@list@edtext@the\@edtext@level
\endcsname\to\sw@inthisedtext}%
1801 \fi%
1802 %

```

`\@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.

```

1803 \global\renewcommand{\@tag}{%
1804 \no@expands #1%
1805 }%
1806 %

```

`\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`.

```

1807 \set@line%
1808 %

```

`\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`.

```

1809 \ifledRcol \global\insert@countR \z@%
1810 \else \global\insert@count \z@ \fi%
1811 %

```

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.

```

1812 \@edtext@secondarg@true%
1813 \ignorespaces #2\relax%
1814 \@edtext@secondarg@false%
1815 %

```

Check if we indeed insert footnotes or critical notes.

```

1816 \ifnum\edtext@notes=\edtext@notes@start%
1817 \unless\if@noneed@notes%
1818 \led@err@EdtextWithoutNote{#1}%
1819 \fi%
1820 \fi%
1821 %

```

With \LaTeX , you must track whether the language reads left to right (English) or right to left (Arabic). `reledmac` defines an `\if@RTL` boolean test is not already defined.

```

1822 \if@RTL%
1823 \IfStrEq{\xabslineref{edtext:start:\csuse{thisedtext@the\
@edtext@level}}}%
1824 {000}%
1825 {}%
1826 {%
1827 \ifnumequal%
1828 {\xabslineref{edtext:start:\csuse{thisedtext@the\
@edtext@level}}}%
1829 {\xabslineref{edtext:end:\csuse{thisedtext@the\@edtext@level
}}}%
1830 {}%
1831 {\write@lemma}%
1832 }%
1833 \flag@start@RTL%
1834 \IfStrEq{\xabslineref{edtext:start:\csuse{thisedtext@the\
@edtext@level}}}%
1835 {000}%
1836 {}%
1837 {%
1838 \ifnumequal%
1839 {\xabslineref{edtext:start:\csuse{thisedtext@the\
@edtext@level}}}%

```

```

1840         {\xabslineref{edtext:end:\csuse{thisedtext@the\@edtext@level
}}} %
1841         {\write@lemma}%
1842         {}%
1843     }%
1844 \else%
1845     \flag@start%
1846     \write@lemma%
1847 \fi%
1848 %

```

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.

```

1849 \endgroup%
1850 \ifdefined\hypertarget%
1851 {%
1852     \Hy@raisedlink@left{\hypertarget{\csuse{thisedtext@the\
@edtext@level}:start}{}}%
1853     \showlemma{#1}%
1854     \Hy@raisedlink{\hypertarget{\csuse{thisedtext@the\@edtext@level}:
end}{}}%
1855 }%
1856 {%
1857     \showlemma{#1}%
1858 }%
1859 %

```

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

```

1860 \ifdefined\hypertarget%
1861 {%
1862     \def\do##1{\Hy@raisedlink{\hypertarget{##1:end}{}}}%
1863     \ifcsdef{applabel@thisedtext@the\@edtext@level}{%
1864         \dolistcsloop{applabel@thisedtext@the\@edtext@level}%
1865         \global\cslet{applabel@thisedtext@the\@edtext@level}{\
empty}%
1866     }{}%
1867 }%
1868 {}%
1869 %

```

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.

```

1870 \ifx\end@lemmas\empty \else%
1871     \glp\end@lemmas\to\x@lemma%
1872     \x@lemma%
1873     \global\let\x@lemma=\relax%

```



```

1874 \fi%
1875 \if@RTL%
1876 \flag@end@RTL%
1877 \else%
1878 \flag@end%
1879 \fi%
1880 %

```

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`.

```

1881 \global\advance\@edtext@level by -1%
1882 \global\@lemmacommand@false%
1883 %

```

We also reset `\@beforeinsertofthisedtext`

```

1884 \global\let\@beforeinsertofthisedtext\relax%
1885 %

```

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

```

1886 \else%
1887 \showlemma{#1} (\textbf{\textsc{Edtext outside numbered paragraph}})\
led@err@edtextoutsidepstart%
1888 \fi%
1889 }%
1890
1891
1892 %

```

`\@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.

```

1893 \newcommand{\@check@edtext@args}[2]{%
1894 \begingroup%
1895 \noexpandarg%
1896 \IfSubStr{#1}{\pstart}{\led@err@PstartInEdtext{first}}{}%
1897 \IfSubStr{#1}{\pend}{\led@err@PendInEdtext{first}}{}%
1898 \IfSubStr{#2}{\pstart}{\led@err@PstartInEdtext{second}}{}%
1899 \IfSubStr{#2}{\pend}{\led@err@PendInEdtext{second}}{}%
1900 \endgroup%
1901 }%
1902 %
1903 %

```

`\@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`.

```

1904 \let\@beforeinsertofthisedtext\relax
1905 %

```

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

```

1906 \newif\ifnumberline
1907 \numberlinetrue
1908 %

```

\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.

```

1909 \newcommand*{\set@line}{%
1910   \ifledRcol
1911     \ifx\line@listR\empty
1912       \global\noteschanged@true
1913       \xdef\l@d@nums{000|000|000|000|000|000|000|\edfont@info}%
1914     \else
1915       \gl@p\line@listR\to\@tempb
1916       \xdef\l@d@nums{\@tempb|\edfont@info}%
1917       \global\let\@tempb=\undefined
1918     \fi
1919     \ifx\annot@listR\empty%
1920       \xdef\l@current@annotR{||}%
1921     \else%
1922       \gl@p\annot@listR\to\@tempb%
1923       \xdef\l@current@annotR{\@tempb}%
1924       \global\let\@tempb=\undefined%
1925     \fi%
1926   \else
1927     \ifx\line@list\empty
1928       \global\noteschanged@true
1929       \xdef\l@d@nums{000|000|000|000|000|000|000|\edfont@info}%
1930     \else
1931       \gl@p\line@list\to\@tempb
1932       \xdef\l@d@nums{\@tempb|\edfont@info}%
1933       \global\let\@tempb=\undefined
1934     \fi
1935     \ifx\annot@list\empty%
1936       \xdef\l@current@annot{||}%
1937     \else%
1938       \gl@p\annot@list\to\@tempb%
1939       \xdef\l@current@annot{\@tempb}%
1940       \global\let\@tempb=\undefined%
1941     \fi%
1942   \fi}
1943

```

```
1944 %
```

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

```
1945 \newcommand*{\edfont@info}{\f@encoding/\f@family/\f@series/\f@shape}
1946
1947 %
```

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. 134).

```
1948 \newcommand*{\lemma}[1]{%
1949   \global\@lemmacommand@true%
1950   \global\renewcommand{\@tag}{%
1951     \no@expands #1%
1952   }%
1953   \ignorespaces%
1954 }%
1955 %
```

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

```
\write@lemma
1956 \newcommand{\@lemma}{%
1957   \booltrue{lemmacommand@the\@edtext@level}%
1958 }%
1959 \newcommand{\write@lemma}{%
1960   \if@lemmacommand@%
1961     \ifledRcol%
1962       \write\linenum@outR{\string\@lemma}%
1963     \else%
1964       \write\linenum@out{\string\@lemma}%
1965     \fi%
1966   \fi%
1967 }%
1968 %
```

\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.

```
1969 \newif\if@lemmacommand@%
1970 %
```

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{13}` only changes the starting line number, and leaves the rest unaltered.

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

```

1971 \newcommand*{\linenum}[1]{%
1972   \xdef\@tempa{#1|}|}|}|}|noexpand\\l@d@nums}%
1973   \global\let\l@d@nums=\empty
1974   \expandafter\line@set\@tempa|\\ignorespaces}
1975 %

```

`\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.

```

1976 \def\line@set#1|#2\\#3|#4\\{%
1977   \gdef\@tempb{#1}%
1978   \ifx\@tempb\empty
1979     \l@d@add{#3}%
1980   \else
1981     \l@d@add{#1}%
1982   \fi
1983   \gdef\@tempb{#4}%
1984   \ifx\@tempb\empty\else
1985     \l@d@add{|}\line@set#2\\#4\\%
1986   \fi}
1987 %

```

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

```

1988 \newcommand{\l@d@add}[1]{\xdef\l@d@nums{\l@d@nums#1}}
1989
1990 %

```

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

`\lineannot@set`

```

1991 \newcommand*{\lineannot}[1]{%
1992   \lineannot@set#1|}%
1993 }%
1994 \def\lineannot@set#1|#2|{%
1995   \expandafter\parse@annot#1|#2|}%
1996   \IfStrEq{#1}{}%
1997     {\let\@tempa\annot@start}%
1998     {\def\@tempa{#1}}%
1999   \IfStrEq{#2}{}%
2000     {\let\@tempb\annot@start}%
2001     {\def\@tempb{#2}}%

```

```

2002 \xdef\l@current@annot{\@tempa|\@tempb}%
2003 }%
2004 %

```

VI.4 Lemma disambiguation

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

The general mechanism is the following:

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

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

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

```

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

```

2. Create lists of lists of `\sameword` by depth of `\edtext`. That is: create a list for `\edtexts` of level 1, a list for `\edtexts` of level 2, a list for `\edtexts` of level 3 etc. For each `\edtext` in these lists, we store all of the relative ranks of

`\saweword` which are called as lemma information. That is: 1) either called in the first argument of `\sameword`, or, 2) called in the `\lemma` macro of the second argument of `\sameword` AND marked by the optional argument of `\saweword` in first argument of `\edtext`.

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

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 2005 \newcommand{\get@sw@txt}[1]{%
2006   \begingroup%
2007   \swnoexpands%
2008   %.
```

Using case sensibility option.

```
2009   \ifsw@caseinsensitive%
2010     \def\@tmpa##1{\lowercase{##1}}%
2011   \else%
2012     \def\@tmpa##1{##1}%
2013   \fi%
2014 %
```

And now, define `\sw@txt`.

```
2015   \@tmpa{\protected@xdef\sw@txt{##1}}%
2016   \endgroup%
2017 }%
2018 %
```

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 2019 \newcommand{\swnoexpands}{%
2020   \let\sameword\l@secondmandarg% Allow to have nested \sameword
2021   \let\emph\@firstofone%
2022   \let\textit\@firstofone%
2023   \let\textbf\@firstofone%
2024   \let\textsc\@firstofone%
2025   \let\framebox\@firstofone%
2026   \let\edtext\dummy@edtext%
2027   \RenewExpandableDocumentCommand{\edindex}{om}{}%
2028   \ifdefined\index%
2029     \RenewExpandableDocumentCommand{\index}{om}{}%
2030   \fi%
2031   \let\selectlanguage\@gobble%
2032   \let\foreignlanguage\@secondoftwo%
2033   \ifdefined\xpg@loaded%
2034     \renewcommand\do[1]{%
2035       \expandafter\RenewExpandableDocumentCommand\expandafter{\csname
text##1\endcsname}{om}{###2}}%
2036     }%
2037     \expandafter\docsvlist\expandafter{\xpg@loaded}%
2038   \fi%
2039 }%
2040 %
```

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

```

2041 \newcommandx{\sameword}[2][1,usedefault]{%
2042     \ifxetex%
2043     \if@RTL%
2044         \led@err@samewordRTL%
2045     \fi%
2046     \fi%
2047     \leavevmode%
2048     \get@sw@txt{#2}%
2049 %

```

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

```

2050 \unless\ifledRcol%
2051     \csnumgdef{sw@\sw@txt}{\csuse{sw@\sw@txt}+\@one}%
2052 %

```

Then, write its value to the numbered file.

```

2053     \protected@write\linenum@out{}\string\@sw{\sw@txt}{\csuse{sw@\sw@txt}
2054     }{#1}}%
2055 %

```

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

```

2055 \else%
2056     \csnumgdef{sw@\sw@txt}{\csuse{sw@\sw@txt}+\@one}%
2057     \protected@write\linenum@outR{}\string\@sw{\sw@txt}{\csuse{sw@\sw@txt}
2058     }{#1}}%
2059 \fi%
2060 %

```

And print the word.

```

2060     #2%
2061 }%
2062 %

```

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

```

\if@addsw \newif\if@addsw%
2064 %

```

\@sw The command printed in the auxiliary files.

```

2065 \newcommand{\@sw}[3]{%
2066     \get@sw@txt{#1}%
2067     \unless\ifledRcol%
2068 %

```

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

```

2069     \csxdef{sw@\sw@txt @\the\absline@num @\the\section@num}{#2}%
2070 %

```

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


```

2071 \numdef{\prev@line}{\the\absline@num-1}%
2072 \ifcsundef{sw@sw@txt @\prev@line @\the\section@num}{%
2073   \csnumgdef{sw@sw@txt @\prev@line @\the\section@num}{#2-1}%
2074 }{}%
2075 %

```

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

```

2076 \numdef{\the@sw}{#2-\csuse{sw@sw@txt @\prev@line @\the\section@num}}%
2077 %

```

And do the same thing for the right side.

```

2078 \else%
2079   \csxdef{sw@sw@txt @\the\absline@numR @\the\section@numR @R}{#2}%
2080   \numdef{\prev@line}{\the\absline@numR-1}%
2081   \ifcsundef{sw@sw@txt @\prev@line @\the\section@numR @R}{%
2082     \csnumgdef{sw@sw@txt @\prev@line @\the\section@numR @R}{#2-1}%
2083   }{}%
2084   \numdef{\the@sw}{#2-\csuse{sw@sw@txt @\prev@line @\the\section@numR @R}}%
2085 \fi%
2086 %

```

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

```

2087 \@tempcnta=\@edtext@level
2088 \@whilenum{\@tempcnta>0}\do{%
2089   \ifcsdef{sw@list@edtext@tmp@\the\@tempcnta}%
2090   {%
2091     \@addswfalse%
2092     \notbool{lemmacommand@\the\@tempcnta}%
2093     {\@addswtrue}%
2094     {\IfStrEq{#3}{inlemma}%
2095      {\@addswtrue}%
2096      {%
2097        \def\do##1{%
2098          \ifnumequal{##1}{\the\@tempcnta}%
2099          {\@addswtrue\listbreak}%
2100          }%
2101        }%
2102        \docsvlist{#3}%
2103      }%
2104    }%
2105    \if@addsw%
2106      \letcs{\@tmp}{sw@list@edtext@tmp@\the\@tempcnta}%
2107      \ifledRcol%
2108        \xright@appenditem{\the@sw}{\the\absline@numR}\to\@tmp%
2109      \else%
2110        \xright@appenditem{\the@sw}{\the\absline@num}\to\@tmp%
2111      \fi%
2112      \cslet{sw@list@edtext@tmp@\the\@tempcnta}{\@tmp}%
2113    \fi%
2114  }%
2115 }%
2116 \advance\@tempcnta by -1%
2117 }%

```

```
2118 }%
2119 %
```

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

```
2120 \newcommandx{\sameword@inedtext}[2][1,usedefault]{%
2121   \get@sw@txt{#2}%
2122   \unless\ifledRcol{%
2123     %
```

Just a precaution.

```
2124   \ifx\sw@list@inedtext\empty%
2125     \def\the@sw{999}%
2126     \def\this@absline{-99}%
2127   \else%
2128     %
```

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

```
2129     \gl@p\sw@list@inedtext\to\@tmp%
2130     \edef\the@sw{\expandafter\@firstoftwo\@tmp}%
2131     \edef\this@absline{\expandafter\@secondoftwo\@tmp}%
2132   \fi%
2133   %
```

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

```
2134   \ifcsdef{sw@\sw@txt @\this@absline @\the\section@num}{%
2135     \numdef{\prev@line}{\this@absline-1}%
2136     \numdef{\sw@atthisline}{\csuse{sw@\sw@txt @\this@absline @\the\
section@num}-\csuse{sw@\sw@txt @\prev@line @\the\section@num}}%
2137   }%
2138   {\numdef{\sw@atthisline}{0}}%
2139   %
```

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

```
2140   \ifnumgreater{\sw@atthisline}{1}%
2141     {\showwordrank{#2}{\the@sw}}%
2142     {#2}%
2143   %
```

And the same for right side.

```
2144   \else%
2145     \ifx\sw@list@inedtext\empty%
2146       \def\the@sw{999}%
2147       \def\this@absline{-99}%
2148     \else%
2149       \gl@p\sw@list@inedtext\to\@tmp%
2150       \edef\the@sw{\expandafter\@firstoftwo\@tmp}%
2151       \edef\this@absline{\expandafter\@secondoftwo\@tmp}%
2152     \fi%
2153     \ifcsdef{sw@\sw@txt @\this@absline @\the\section@numR @R}{%
2154       \numdef{\prev@line}{\this@absline-1}%
2155     }
```

```

2155 \numdef{\sw@atthisline}{\csuse{sw@\sw@txt @\this@absline @\the\
section@numR @R}-\csuse{sw@\sw@txt @\prev@line @\the\section@numR @R}}%
2156 }%
2157 {\numdef{\sw@atthisline}{0}}%
2158 \ifnumgreater{\sw@atthisline}{1}%
2159 {\showwordrank{#2}{\the@sw}}%
2160 {#2}%
2161 \fi%
2162 }%
2163 %

```

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

```

2164 \newcommand{\showwordrank}[2]{%
2165 #1\textsuperscript{#2}%
2166 }%
2167 %

```

VII Paragraph decomposition and reassembly

In order to be able to count the lines of text and affix line numbers, we add an extra stage of processing for each paragraph. We send the paragraph into a box register, rather than straight onto the vertical list, and when the paragraph ends we slice the paragraph into its component lines; to each line we add any notes or line numbers, add a command to write to the line-list, and then at last send the line to the vertical list. This section contains all the code for this processing.

VII.1 Boxes, counters, `\pstart` and `\pend`

`\raw@text` Here are numbers and flags that are used internally in the course of the paragraph decomposition.

`\ifnumberedpar@` When we first form the paragraph, it goes into a box register, `\raw@text`, instead of onto the current vertical list. The `\ifnumberedpar@` flag will be `true` while a paragraph is being processed in that way. `\num@lines` will store the number of lines in the paragraph when it is complete. When we chop it up into lines, each line in turn goes into the `\one@line` register, and `\par@line` will be the number of that line within the paragraph.

```

2168 \newbox\raw@text
2169 \newif\ifnumberedpar@
2170 \newcount\num@lines
2171 \newbox\one@line
2172 \newcount\par@line
2173 %

```

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

```
2174 \newcount\pstarts@typeset@L%
2175 \newcount\pstarts@read@L%
2176 %
```

`\pstart` starts the paragraph by clearing the `\inserts@list` list and other relevant variables, and then arranges for the subsequent text to go into the `\raw@text` box. `\AtStartEveryPstart` `\pstart` needs to appear at the start of every paragraph that is to be numbered; the `\autopar` command below may be used to insert these commands automatically.

`\numberpstarttrue` `\autopar` command below may be used to insert these commands automatically. Beware: everything that occurs between `\pstart` and `\pend` is happening within a group; definitions must be global if you want them to survive past the end of the paragraph.

`\labelpstarttrue`
`\labelpstartfalse`

`\thepstart`

```
2177 \if@every@pstart@star@
2178 \newcommand{\AtStartEveryPstart}[1]{%
2179   \ifstrempy{#1}%
2180   {\gdef\@at@start@every@pstart{}}%
2181   {\gdef\@at@start@every@pstart{#1}}%
2182 }%
2183 \def\@at@start@every@pstart{}%
2184
2185 \newif\if@every@pstart@star@%
2186 \newcommand{\AtEveryPstart}[1]{%
2187   \ifstrempy{#1}%
2188   {\gdef\@at@every@pstart{}}%
2189   {\gdef\@at@every@pstart{\noindent#1}}%
2190   \global\@at@every@pstart@star@false%
2191 }%
2192 \WithSuffix\newcommand\AtEveryPstart*[1]{%
2193   \ifstrempy{#1}%
2194   {\gdef\@at@every@pstart{}}%
2195   {\gdef\@at@every@pstart{#1}}%
2196   \global\@at@every@pstart@star@true%
2197 }%
2198 \def\@at@every@pstart{}%
2199
2200 \newcounter{pstart}
2201 \renewcommand{\thepstart}{\bfseries\@arabic\c@pstart}. }
2202 \newif\ifnumberpstart
2203 \numberpstartfalse
2204 \newif\iflabelpstart
2205 \labelpstartfalse
2206 \newcommandx*{\pstart}[2][1,2,usedefault]{%
2207   \normal@pars%
2208   \ifboolexpr{%
2209     test {\ifstrempy{#1}}%
2210     and test {\ifstrempy{#2}}%
2211   }%
2212   {\@at@every@pstart}%
2213   {%
2214     \ifstrempy{#1}{\noindent#1}%
```

```

2215 \ifstrempy{#2}{#{#2}%
2216 }%
2217 \ifautopar%
2218 \autopar%
2219 \fi%
2220 \ifluatex%
2221 \edef\l@luatextextdir@L{\the\textdir}%
2222 \fi%
2223 \@nbreaktrue%
2224 \ifnumbering \else%
2225 \led@err@PstartNotNumbered%
2226 \beginnumbering%
2227 \fi%
2228 \ifnumberedpar@%
2229 \led@err@PstartInPstart%
2230 \pend%
2231 \fi%
2232 \list@clear{\inserts@list}%
2233 \global\let\next@insert=\empty%
2234 \begingroup\normal@pars%
2235 \global\advance \l@dnumpstartsL\@ne
2236 \global\advance \pstarts@typeset@L\@ne%
2237 \global\advance \pstarts@read@L\@ne%
2238 \global\setbox\raw@text=\vbox\bgroup%
2239 \if@nbreak%
2240 \if@afterindent\else%
2241 \unless\ifinstanza%
2242 \noindent%
2243 \fi%
2244 \global\@afterindenttrue%
2245 \fi%
2246 \fi%
2247 \ifboolexpr{%
2248 bool{autopar}%
2249 and bool{by@autopar}%
2250 }%
2251 {}%
2252 {%
2253 \ifnumberpstart%
2254 \ifinstanza\else%
2255 \ifsidepstartnum\else%
2256 \thepstart%
2257 \fi%
2258 \fi%
2259 \fi%
2260 }%
2261 \numberedpar@true%
2262 \iflabelpstart%
2263 \protected@edef\@currentlabel{\p@pstart\thepstart}%
2264 \fi%
2265 \l@dzeropenalties%
2266 \@at@start@every@pstart%
2267 \global\by@autoparfalse%
2268 \ignorespaces%because not automatically ignored if an optional argument

```

```

is used (classical TeX behavior for space after commands)
2269 }
2270 %

```

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

```

2271 \newcommandx*{\pend}[2][1,2,usedefault]{\ifnumbering \else%
2272   \led@err@PendNotNumbered%
2273   \fi%
2274   \global\l@dskipversenumberfalse%
2275   \ifnumberedpar@ \else%
2276   \led@err@PendNoPstart%
2277   \fi%
2278 %

```

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.

```

2279 \l@dzzeropenalties%
2280 \@at@end@every@pend%
2281 \endgraf\global\num@lines=\prevgraf\egroup%
2282 \global\par@line=0%
2283 \get@firsthalf@antilabe@wd%
2284 %

```

Output the lines.

```

2285 \loop\ifvbox\raw@text%
2286   \do@line%
2287   \repeat%
2288 %

```

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

```

2289 \flush@notes%
2290 \endgroup%
2291 \ignorespaces%
2292 %

```

Increase `pstart` counter.

```

2293 \ifnumberpstart%
2294   \global\pstartnumtrue%
2295   \fi%
2296   \addtocounter{pstart}{1}%
2297   \ifcontinuousnumberingwithcolumns%
2298     \addtocounter{pstartL}{1}%
2299     \addtocounter{pstartR}{1}%
2300   \fi%
2301 %

```

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

```

2302 \normal@pars%
2303 \ifboolexpr{%
2304   test {\ifstrempy{#1}}%
2305   and test {\ifstrempy{#2}}%
2306 }%
2307   {\at@every@pend}%
2308   {%
2309   \ifstrempy{#1}{\noindent#1}%
2310   \ifstrempy{#2}{#2}%
2311   }%
2312 %

```

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.

```

2313 \@nobreakfalse%
2314 \ifautopar%
2315   \autopar%
2316 \fi%
2317 %

```

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.

```

2318 \ifbypstart@%
2319   \begingroup%
2320     \unless\ifinstanza%
2321       \let\leavevmode\relax%
2322       \setlinenum{0}%
2323     \fi%
2324   \endgroup%
2325   \resetprevline@%
2326 \fi%
2327 }
2328 %

```

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.

```

\AtEveryPend29
\at@every@pend30
\ifat@every@pend@star@31 \newif\ifat@every@pend@star@%
2332 \newcommand{\AtEveryPend}[1]{%
2333   \ifstrempy{#1}%
2334     {\gdef\at@every@pend{}}%
2335     {\gdef\at@every@pend{\noindent#1}}%
2336   \global\at@every@pend@star@false%
2337 }%
2338 \WithSuffix\newcommand\AtEveryPend*[1]{%
2339   \ifstrempy{#1}%
2340     {\gdef\at@every@pend{}}%
2341     {\gdef\at@every@pend{#1}}%
2342   \global\at@every@pend@star@true%

```

```

2343 }%
2344 \xdef\at@every@pend{}%
2345
2346 %

```

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

```

2347 \newcommand{\AtEndEveryPend}[1]{%
2348   \ifstrempy{#1}%
2349   {\xdef\@at@end@every@pend{}}%
2350   {\gdef\@at@end@every@pend{#1}}%
2351 }%
2352 \def\@at@end@every@pend{%
2353 %

```

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

```

2354 \newcommand*\l@dzzeropenalties{%
2355   \brokenpenalty \z@ \clubpenalty \z@
2356   \displaywidowpenalty \z@ \interlinepenalty \z@ \predisplaypenalty \z@
2357   \postdisplaypenalty \z@ \widowpenalty \z@}
2358
2359 %

```

\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.


```

2360 \newif\ifautopar
2361 \newif\ifby@autopar%
2362 \newcommand*{\autopar}{
2363   \ifledRcol
2364   \ifnumberingR \else
2365   \led@err@AutoparNotNumbered
2366   \beginnumberingR
2367   \fi
2368   \else
2369   \ifnumbering \else
2370   \led@err@AutoparNotNumbered
2371   \beginnumbering
2372   \fi
2373   \fi
2374   \autopartrue
2375   \everypar{\setbox0=\lastbox
2376   \endgraf \vskip-\parskip
2377   \global\by@autopartrue%
2378   \pstart \noindent \kern\wd0%
2379   \ifnumberpstart%
2380   \ifinstanza\else%
2381   \thepstart%
2382   \fi%
2383   \fi%
2384   \let\par=\pend}%
2385   \ignorespaces}
2386 %

```

\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.

```

2387 \newcommand*{\normal@pars}{\everypar{}\let\par\endgraf}
2388
2389 %

```

\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.

```

2390 \newif\ifautopar@pause
2391 %

```

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.

```

2392 \newcommand*{\l@dunhbox@line}[1]{\unhbox #1}
2393 \newcommand*{\do@line}{%
2394   {\vbadness=10000

```

```

2395 \splittopskip=\z@
2396 \do@linehook
2397 \l@demptyd@ta
2398 \global\setbox\one@line=\vsplit\raw@text to\baselineskip}%
2399 \unvbox\one@line \global\setbox\one@line=\lastbox
2400 \getline@num
2401 \IfStrEq{\led@pb@setting}{before}{\led@check@pb\led@check@nopb}{\}
2402 \ifnum\@lock>\@ne
2403 \inserthangingsymboltrue
2404 \else
2405 \inserthangingsymbolfalse
2406 \fi
2407 \check@pb@in@verse
2408 \ifl@dhidenumber%
2409 \global\l@dhidenumberfalse%
2410 \f@x@l@cks%
2411 \else%
2412 \affixline@num%
2413 \fi%
2414 %

```

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

```

2415 \xifinlist{\the\pstarts@typeset@L}{\eled@sections@}%
2416 {\print@eledsection}%
2417 {\print@line}%
2418 \IfStrEq{\led@pb@setting}{after}{\led@check@pb\led@check@nopb}{\}
2419 }%
2420 %

```

VII.2.2 Process for “normal” line

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

```

2421 \def\print@line{
2422 %

```

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

```

2423 \affixpstart@num%
2424 %

```

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

```

2425 \hb@xt@ \linewidth{\%
2426 %

```

User hooks.

```

2427 \do@insidelinehook%
2428 \csuse{insidethis@the\absline@num @the\section@num}%
2429 \global\csundef{insidethis@the\absline@num @the\section@num}%
2430 %

```

Left line number

```

2431 \l@dld@ta%
2432 %

```

Prepare text to be inserted before notes.

```
2433 \if@firstlineofpage%
2434 \set@Xtxtbeforenotes%
2435 \set@txtbeforenotesX%
2436 \global\@firstlineofpagefalse%
2437 \fi%
2438 %
```

Insert footnotes made of manuscripts data and critical footnotes.

```
2439 \ifdefstring{\ms@data@position}{msdata-regular}{%
2440 \insert@msdata%
2441 \add@inserts%
2442 \add@Xgroupbyline%
2443 }{%
2444 \add@inserts%
2445 \add@Xgroupbyline%
2446 \insert@msdata%
2447 }%
2448 %
```

Insert marginal notes.

```
2449 \affixside@note%
2450 %
```

Print left notes.

```
2451 \l@dlsn@te
2452 %
```

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

```
2453 {\ledllfill\hb@xt@ \wd\one@line{%
2454 \new@line%
2455 %
```

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

```
2456 \continuousnumberingwithcolumns@sync@linenumber@singletext%
2457 %
```

If we use Lua \TeX then restore the direction.

```
2458 \ifluatex%
2459 \texdir\l@luatextextdir@L%
2460 \fi%
2461 %
```

Insert, if needed, the hanging symbol.

```
2462 \inserthangingsymbol%
2463 %
```

And so, print the line.

```
2464 \l@dunhbox@line{\one@line}}%
2465 %
```

Right line number

```

2466 \ledrlfill\l@drd@ta%
2467 %

```

Print right notes.

```

2468 \l@drsn@te%
2469 }}%
2470 %

```

And reinsert penalties (for page breaking)...

```

2471 \add@penalties%
2472 }
2473 %

```

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.

```

2474 \def\print@eledsection{%
2475   \disable@edindex%
2476   \if@firstlineofpage%
2477     \set@Xtxtbeforenotes%
2478     \set@txtbeforenotesX%
2479     \global\@firstlineofpagefalse%
2480   \fi%
2481   \ifdefstring{\ms@data@position}{msdata-regular}{%
2482     \insert@msdata%
2483     \add@inserts%
2484     \add@Xgroupbyline%
2485   }{%
2486     \add@inserts%
2487     \add@Xgroupbyline%
2488     \insert@msdata%
2489   }%
2490   \affixside@note%
2491   \numdef{\temp@}{\pstarts@typeset@L-1}%
2492   \xifinlist{\temp@}{\eled@sections@@}{\@nobreaktrue}{\@nobreakfalse}%
2493   \@eled@sectioningtrue%
2494   \csuse{eled@sectioning@the\pstarts@typeset@L}%
2495   \@eled@sectioningfalse%
2496   \global\csundef{eled@sectioning@the\pstarts@typeset@L}%
2497   \if@RTL%
2498     \hspace{-3\paperwidth}%
2499     {\hbox{\l@dunhbox@line{\one@line}} \new@line}%
2500   \else%
2501     \hspace{3\paperwidth}%
2502     {\new@line \hbox{\l@dunhbox@line{\one@line}}}%
2503   \fi%
2504   \vskip-\baselineskip%
2505   \continuousnumberingwithcolumns@sync@linenumber@singletext%
2506   \restore@edindex%
2507 }

```

```
2508 %
```

VII.2.4 Hooks

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

```
2509 \newcommand*{\do@linehook}{%
2510 \newcommand*{\do@insidelinehook}{%
2511 %
```

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

```
2512 \newcommand*{\dolinehook}[1]{\gdef\do@linehook{#1}}%
2513 \newcommand*{\doinsidelinehook}[1]{\gdef\do@insidelinehook{#1}}%
2514
2515 %
```

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

```
2516 \newcommand{\doinsidethislinehook}[1]{%
2517   \leavevmode%In case it begins with a \pstart, ensure the \@insidethisline
    is written after \@nl
2518   \ifledRcol%
2519     \write\linenum@outR{\string\@insidethisline[\unexpanded{#1}]}%
2520   \else%
2521     \write\linenum@out{\string\@insidethisline[\unexpanded{#1}]}%
2522   \fi%
2523 }%
2524 %
```

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

```
2525 \newcommand{\@insidethisline}[1]{%
2526   \ifledRcol%
2527     \csgappto{insidethisR@the\absline@numR @the\section@numR}{#1}%
2528   \else%
2529     \csgappto{insidethis@the\absline@num @the\section@num}{#1}%
2530   \fi%
2531 }%
2532 %
```

VII.2.5 Sidenotes and marginal line number initialization

`\l@demptyd@ta` Nulls the `\. . .d@ta`, which may later hold line numbers. Similarly for `\l@dcsnotetext`,
`\l@dld@ta` `\l@dcsnotetext@l`, `\l@dcsnotetext@r` for the texts of the sidenotes, left and right
`\l@drrd@ta` notes.
`\l@dcsnotetext`
`\l@dcsnotetext@l`
`\l@dcsnotetext@r`

```

2533 \newcommand*{\l@emptyd@ta}{%
2534   \gdef\l@dld@ta{}%
2535   \gdef\l@drd@ta{}%
2536   \gdef\l@dcsnotetext@l{}%
2537   \gdef\l@dcsnotetext@r{}%
2538   \gdef\l@dcsnotetext{}}
2539
2540 %

```

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

\l@drsn@te

```

2541 \newcommand{\l@dlsn@te}{%
2542   \ifboolexpr{%
2543     bool {l@dprintingcolumns}%
2544     and bool {ledRcol@}%
2545   }{% If we are on a right column
2546     \@tempdima=\@morespace@leftnote@rightcolumn%
2547   }{%
2548     \@tempdima=\z@%
2549   }%
2550   \hb@xt@ \z@{\hss\box\l@dldp@rbox\kern\ledlsnotesep\hskip\@tempdima}%
2551 }%
2552 \newcommand{\l@drsn@te}{%
2553   \ifboolexpr{%
2554     bool {l@dprintingcolumns}%
2555     and not bool {ledRcol@}%
2556   }{% If we are on a left column
2557     \@tempdima=\@morespace@rightnote@leftcolumn%
2558   }{%
2559     \@tempdima=\z@%
2560   }%
2561   \hb@xt@ \z@{\hskip\@tempdima\kern\ledrsnotesep\box\l@drp@rbox\hss}%
2562 }%
2563
2564 %

```

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

\ledrlfill

```

2565 \newcommand*{\ledllfill}{\unless\if@RTL\hfil\fi}%
2566 \newcommand*{\ledrlfill}{\if@RTL\hfil\fi}%
2567
2568 %

```

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.

```

2569 \newcommand*{\getline@num}{%
2570   \global\advance\absline@num \@ne%
2571   \do@actions

```

```

2572 \do@ballast
2573 \ifnumberline
2574   \ifsublines@
2575     \ifnum\sub@lock<\tw@
2576       \global\advance\subline@num \@ne
2577     \fi
2578   \else
2579     \ifnum\@lock<\tw@
2580       \global\advance\line@num \@ne
2581       \global\subline@num \z@
2582     \fi
2583   \fi
2584 \fi
2585 }
2586 %

```

\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. 172).

\ballast@count First we set up the required counters; they are initially set to zero, and will remain so unless you type `\setcounter{ballast}{\langle some figure \rangle}` in your document.

\c@ballast

```

2587 \newcount\ballast@count
2588 \newcounter{ballast}
2589 \setcounter{ballast}{0}
2590 %

```

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.

```

2591 \newcommand*{\do@ballast}{\global\ballast@count \z@
2592   \begingroup
2593     \advance\absline@num \@ne
2594     \ifnum\next@actionline=\absline@num
2595       \ifnum\next@action>-1001\relax
2596         \global\advance\ballast@count by -\c@ballast
2597       \fi
2598     \fi
2599   \endgroup}
2600 %

```

\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`.

```

2601 \newcommand*{\do@actions}{%
2602   \global\let\do@actions@next=\relax
2603   \ifnum\absline@num<\next@actionline\else
2604   %

```

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.

```

2605   \ifnum\next@action>-1001
2606     \global\page@num=\next@action
2607     \ifresumenumbering@start%
2608       \setbox0=\hbox{}%Required to get the correct page number, when the
\resumenumbering is just after a \newpage
2609       \ifnum\pausenumbering@page@num<\page@num%
2610         \global\resumenumbering@startfalse%
2611       \fi%
2612     \fi%
2613     \ifboolexpr{%
2614       bool{resumenumbering@start}%
2615       and test {\ifdimgreater{\pagedepth}{\z@}}%
2616     }%
2617     {%
2618       \global\@firstlineofpagetrue}%
2619     \ifcsdef{reset@line\the\absline@num @\the\section@num}%
2620     {%
2621       \global\line@num=\z@%
2622       \global\subline@num=\z@%
2623       \resetprevline@%
2624     }%
2625     {%
2626       \global\resumenumbering@startfalse%
2627       \add@msdata@firstlineofpage%
2628   %

```

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`.)

```

2629   \else
2630     \ifnum\next@action<-4999
2631       \@l@dttempcnta=-\next@action
2632       \advance\@l@dttempcnta by -5001
2633       \ifsublines@
2634         \global\subline@num=\@l@dttempcnta
2635       \else
2636         \global\line@num=\@l@dttempcnta
2637       \fi
2638   %

```

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

```

2639   \else
2640     \@l@dttempcnta=-\next@action
2641     \advance\@l@dttempcnta by -1000
2642     \do@actions@fixedcode
2643   \fi
2644 \fi

```



```

2645 %
    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.

2646 \ifx\actionlines@list\empty
2647   \gdef\next@actionline{1000000}%
2648 \else
2649   \gl@p\actionlines@list\to\next@actionline
2650   \gl@p\actions@list\to\next@action
2651   \global\let\do@actions@next=\do@actions
2652 \fi
2653 \fi
2654 %

    Make the recursive call, if necessary.

2655 \do@actions@next}
2656
2657 %

```

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

```

2658 \newcommand*{\do@actions@fixedcode}{%
2659   \ifcase\@l@dttempcnta
2660   \or% % 1001 = starting sublineation
2661     \global\sublines@true
2662   \or% % 1002 = ending sublineation
2663     \global\sublines@false
2664   \or% % 1003 = starting locking number
2665     \global\@lock=\@ne
2666   \or% % 1004 = ending locking number
2667     \ifnum\@lock=\tw@
2668       \global\@lock=\thr@@
2669     \else
2670       \global\@lock=\z@
2671     \fi
2672   \or% % 1005 = starting locking subnumber
2673     \global\sub@lock=\@ne
2674   \or% % 1006 = ending locking subnumber
2675     \ifnum\sub@lock=\tw@
2676       \global\sub@lock=\thr@@
2677     \else
2678       \global\sub@lock=\z@
2679     \fi
2680   \or% % 1007 = skipping numbering
2681     \l@dskipnumbertrue
2682   \or% % 1008 = skipping numbering in stanza
2683     \l@dskipversenumbertrue%
2684   \or% % 1009 = hiding number
2685     \l@dhiddenumbertrue
2686   \or% % 1010 = inserting msdata
2687     \add@msdata%
2688   \else

```

```

2689 \led@warn@BadAction
2690 \fi}
2691
2692
2693 %

```

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.

```

2694 \newcommand{\continuousnumberingwithcolumns@sync@linenumber@singletext}{%
2695   \ifcontinuousnumberingwithcolumns%
2696     \unless\ifafterendnumberingR%
2697       \new@lineR%
2698       \xappto\next@line@list@stuffR{%
2699         \unexpanded{\global\line@numR=}\the\line@num%
2700       }%
2701     \fi%
2702   \fi%
2703 }%
2704 %

```

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`

```

2705 \newcommand{\linenumannotation}[1]{%
2706   \leavevmode%In case it begins with a \pstart, ensure the \@annot is
written after \@nl
2707   \ifledRcol%
2708     \write\linenum@outR{\string\@annot[#1]}%
2709   \else%
2710     \write\linenum@out{\string\@annot[#1]}%
2711   \fi%
2712 }%
2713
2714 \newcommand{\resetlinenumannotation}[0]{%
2715   \leavevmode%In case it begins with a \pstart, ensure the \@resetannot is
written after \@nl
2716   \ifledRcol%
2717     \write\linenum@outR{\string\@resetannot}%
2718   \else%
2719     \write\linenum@out{\string\@resetannot}%
2720   \fi%
2721 }%
2722
2723 %

```

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

```

2724 \def\Xlinenumannotationposition@side{after}%
2725 %

```

\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`

```

2726 \def\Xwraplinenumannotation@ref{\textsuperscript}%
2727 \def\Xwraplinenumannotation@side{\textsuperscript}%
2728 \newtoggle{Xnoidenticallinenumannotation@ref}%
2729 \newtoggle{Xnoidenticallinenumannotation@side}%
2730 %

```

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

```

2731 \newif\iflinenumannotationotherside%
2732 %

```

\@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.

```

2733 \newcommand{\@annot}[1]{%
2734   \store@annot@to@absline{#1}%
2735   \def\current@annot{#1}%
2736 }%
2737 \newcommand{\store@annot@to@absline}[1]{%
2738   \ifledRcol%
2739     \ifcsdef{annotR@the\absline@numR @the\section@numR}{%
2740       \csgappto{annotR@the\absline@numR @the\section@numR}{\
2741         @linenumannotationsep#1}%
2742       }{%
2743         \csgdef{annotR@the\absline@numR @the\section@numR}{#1}%
2744       }%
2745     \else%
2746       \ifcsdef{annot@the\absline@num @the\section@num}{%
2747         \csgappto{annot@the\absline@num @the\section@num}{\
2748           @linenumannotationsep#1}%
2749         }{%
2750           \csgdef{annot@the\absline@num @the\section@num}{#1}%
2751         }%
2752       }%
2753     }%
2754   }%

```

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

```

2753 \let\current@annot=\empty%
2754 %

```

`\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.

```

2755 \newcommand{\reset@current@annot}{%
2756   \unless\ifnoresetlinenumannotation@
2757     \let\current@annot\empty%
2758   \fi
2759 }%
2760 \newcommand{\@resetannot}{%
2761   \let\current@annot\empty%
2762 }%
2763 %

```

`\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.

```

2764 \def\parse@annot#1|#2|{%
2765   \gdef\annot@start{#1}%
2766   \gdef\annot@end{#2}%
2767 }%
2768 %

```

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

```

\@linenumannotationsep
2769 \newcommand{\setlinenumannotationsep}[1]{\gdef\@linenumannotationsep{#1}}%
2770 \def\@linenumannotationsep{,}%
2771 %

```

X Line number printing

`\affixline@num` `\affixline@num` just puts a left line number into `\l@dld@ta` or a right line number into `\l@drd@ta` as required.

To determine whether we need to affix a line number to this line, we compute the following:

$$n = \text{int}((\text{linenum} - \text{firstlinenum}) / \text{linenumincrement})$$

$$m = \text{firstlinenum} + (n \times \text{linenumincrement})$$

(where *int* truncates a real number to an integer). *m* will be equal to *linenum* only if we are to paste a number on here. However, the formula breaks down for the first line to number (and any before that), so we check that case separately: if `\line@num ≤ \firstlinenum`, we compare the two directly instead of making these calculations.

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

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

```

2772 \newcommand*{\affixline@num}{%
2773 %

```

No number is attached if `\ifl@dskipnumber` is TRUE (and then it is set to its normal FALSE value). No number is attached if `\ifnumberline` is FALSE (the normal value is TRUE).

```

2774 \ifl@groupnotesL@else
2775   \ifnumberline
2776     \ifl@dskipnumber
2777       \global\l@dskipnumberfalse
2778     \else
2779       \ifsublines@
2780         \l@dtmpcntb=\subline@num
2781         \ifnum\subline@num>\c@firstsublinenum
2782           \l@dtmpcnta=\subline@num
2783           \advance\l@dtmpcnta by-\c@firstsublinenum
2784           \divide\l@dtmpcnta by\c@sublinenumincrement
2785           \multiply\l@dtmpcnta by\c@sublinenumincrement
2786           \advance\l@dtmpcnta by\c@firstsublinenum
2787         \else
2788           \l@dtmpcnta=\c@firstsublinenum

```

```
2789 \fi
2790 %
```

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.

```
2791 \checksublock
2792 %
```

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

```
2793 \else
2794 \l@dttempcntb=\line@num
2795 %
```

Check on the `\linenumberlist` If it is `\empty` use the standard algorithm.

```
2796 \ifx\linenumberlist\empty
2797 \ifnum\line@num>\c@firstlinenum
2798 \l@dttempcnta=\line@num
2799 \advance\l@dttempcnta by-\c@firstlinenum
2800 \divide\l@dttempcnta by\c@linenumincrement
2801 \multiply\l@dttempcnta by\c@linenumincrement
2802 \advance\l@dttempcnta by\c@firstlinenum
2803 \else
2804 \l@dttempcnta=\c@firstlinenum
2805 \fi
2806 \else
2807 %
```

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

```
2808 \l@dttempcnta=\line@num
2809 \edef\rem@inder{\linenumberlist,\number\line@num,}%
2810 \edef\sc@n@list{\def\noexpand\sc@n@list
2811 ###1,\number\l@dttempcnta,###2|{\def\noexpand\rem@inder
2812 {###2}}}%
2813 \sc@n@list\expandafter\sc@n@list\rem@inder|
2814 \ifx\rem@inder\empty%
2815 \advance\l@dttempcnta\@ne
2816 \fi
2817 %
```

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

```
2818 \checklock
2819 \fi
2820 %
```

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

```
2821 \ifnum\l@dttempcnta=\l@dttempcntb
2822 \ifl@dskipversenumber\else
2823 %
```

If we got here, we are going to print a line number; so now we need to calculate a number that will tell us which side of the page will get the line number. We start from `\line@margin`, which asks for one side always if it is less than 2; and then if the side does depend on the page number, we simply add the page number to this side code—because the values of `\line@margin` have been devised so that this produces a number that is even for left-margin numbers and odd for right-margin numbers.

For \LaTeX we have to consider two column documents as well. In this case Peter Wilson thought we need to put the numbers at the outside of the column — the left of the first column and the right of the second. Do the twocolumn stuff before going on with the original code.

`\l@dld@ta` A left line number is stored in `\l@dld@ta` and a right one in `\l@drd@ta`.

`\l@drd@ta`

```

2824         \if@twocolumn
2825             \if@firstcolumn
2826                 \gdef\l@dld@ta{\llap{\leftlinenum}}}%
2827                 \iflinenumannotationotherside%
2828                     \gdef\l@drd@ta{\rlap{\rightlinenumannotation}}}%
2829                 \fi%
2830             \else
2831                 \gdef\l@drd@ta{\rlap{\rightlinenum}}}%
2832                 \iflinenumannotationotherside%
2833                     \gdef\l@dld@ta{\llap{\leftlinenumannotation}}}%
2834                 \fi%
2835             \fi
2836         \else
2837             \ifboolexpr{bool {l@dprintingcolumns} and test {\
ifnumgreater{\line@margin@columns}{\m@ne}}}%
2838                 {\@l@tempcntb=\line@margin@columns}%
2839                 {\@l@tempcntb=\line@margin}%
2840                 \ifnum\@l@tempcntb>\@ne
2841                     \advance\@l@tempcntb \page@num
2842                 \fi
2843             \ifboolexpr{%
2844                 bool {l@dprintingcolumns}%
2845                 and (%
2846                     (test {\ifdefstring{\linenum@OnlyPages@ForColumns}{left
2847
2848                     and test {\ifnumodd{\page@num}}}%
2849                     )%
2850                     or%
2851                     (test {\ifdefstring{\linenum@OnlyPages@ForColumns}{right
2852
2853                     and not test {\ifnumodd{\page@num}}}%
2854                     )%
2855                     )%
2856                 }%
2857                 {}%
2858                 {\%
2859                     \ifodd\@l@tempcntb%
2860                         \gdef\l@drd@ta{\rlap{\rightlinenum}}}%
2861                         \iflinenumannotationotherside%
2862                             \gdef\l@dld@ta{\llap{\leftlinenumannotation}}}%
2863                         \fi%

```

```

2862         \else%
2863         \gdef\l@dld@ta{\llap{\leftlinenum}}}%
2864         \iflinenumannotationotherside%
2865         \gdef\l@drd@ta{\rlap{\rightlinenumannotation}}}%
2866         \fi%
2867     \fi%
2868 }%
2869 \fi
2870 \fi
2871 \fi
2872 %

```

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

```

2873     \f@x@l@cks
2874 \fi
2875 \fi
2876 \fi
2877 }
2878 %
2879 %

```

`\ch@cksub@l@ck` These macros handle line number locking for `\affixline@num`. `\ch@cksub@l@ck` checks subline locking. If it fails, then we disable the line-number display by setting the counters to arbitrary but unequal values.

```

2880 \newcommand*{\ch@cksub@l@ck}{%
2881 \ifcase\sub@lock
2882 \or
2883 \ifnum\sublock@disp=\@ne
2884 \l@ldtempcntb=\z@ \l@ldtempcnta=\@ne
2885 \fi
2886 \or
2887 \ifnum\sublock@disp=\tw@ \else
2888 \l@ldtempcntb=\z@ \l@ldtempcnta=\@ne
2889 \fi
2890 \or
2891 \ifnum\sublock@disp=\z@
2892 \l@ldtempcntb=\z@ \l@ldtempcnta=\@ne
2893 \fi
2894 \fi}
2895 %

```

Similarly for line numbers.

```

2896 \newcommand*{\ch@ck@l@ck}{%
2897 \ifcase\@lock
2898 \or
2899 \ifnum\lock@disp=\@ne
2900 \l@ldtempcntb=\z@ \l@ldtempcnta=\@ne
2901 \fi
2902 \or
2903 \ifnum\lock@disp=\tw@ \else
2904 \l@ldtempcntb=\z@ \l@ldtempcnta=\@ne

```



```

2905         \fi
2906     \or
2907         \ifnum\lock@disp=\z@
2908             \@l@tempcntb=\z@ \@l@tempcnta=\@ne
2909         \fi
2910     \fi}
2911 %

```

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

```

2912 \newcommand*{\f@x@l@cks}{%
2913     \ifcase\@lock
2914     \or
2915         \global\@lock=\tw@
2916     \or \or
2917         \global\@lock=\z@
2918     \fi
2919     \ifcase\sub@lock
2920     \or
2921         \global\sub@lock=\tw@
2922     \or \or
2923         \global\sub@lock=\z@
2924     \fi}
2925 %
2926 %

```

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.

```

\leftpstartnum27
\rightpstartnum28 \newif\ifsidepstartnum
\ifsidepstartnum29 \newcommand*{\affixpstart@num}{%
2930     \ifsidepstartnum
2931         \if@twocolumn
2932             \if@firstcolumn
2933                 \gdef\l@dld@ta{\llap{\leftpstartnum}}}%
2934             \else
2935                 \gdef\l@drd@ta{\rlap{\rightpstartnum}}}%
2936             \fi
2937         \else
2938             \@l@tempcntb=\line@margin%

```

```

2939         \ifnum\@l@tempcntb>\@ne
2940             \advance\@l@tempcntb \page@num
2941         \fi
2942         \ifodd\@l@tempcntb
2943             \gdef\l@drd@ta{\rlap{{\rightpstartnum}}}%
2944         \else
2945             \gdef\l@dld@ta{\llap{{\leftpstartnum}}}%
2946         \fi
2947     \fi
2948 \fi
2949
2950 }
2951 %
2952
2953 \newif\ifpstartnum
2954 \pstartnumtrue
2955 \newcommand*{\leftpstartnum}{
2956     \ifpstartnum\thepstart
2957     \kern\linenumsep\fi
2958     \global\pstartnumfalse
2959 }
2960 \newcommand*{\rightpstartnum}{
2961     \ifpstartnum
2962     \kern\linenumsep
2963     \thepstart
2964     \fi
2965     \global\pstartnumfalse
2966 }
2967 %

```

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.

```

2968 \list@create{\inserts@list}
2969 %

```

`\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`.

```
2970 \newcommand*{\add@inserts}{%
2971   \global\let\add@inserts@next=\relax
2972 }
```

If `\inserts@list` is empty, there are not any more notes or insertions for this paragraph, and we need not waste our time.

```
2973 \ifx\inserts@list\empty \else
2974 %
```

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.

```
2975 \ifx\next@insert\empty
2976   \ifx\insertlines@list\empty
2977     \global\noteschanged@true
2978     \gdef\next@insert{100000}%
2979   \else
2980     \glp\insertlines@list\to\next@insert
2981   \fi
2982 \fi
2983 %
```

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.

```
2984 \ifnum\next@insert=\absline@num
2985   \glp\inserts@list\to\@insert
2986   \@insert
2987   \global\let\@insert=\undefined
2988   \global\let\next@insert=\empty
2989   \global\let\add@inserts@next=\add@inserts
2990 \fi
2991 \fi
2992 %
```

Make the recursive call, if necessary.

```
2993 \add@inserts@next}
2994
2995 %
```

`\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`.

```
2996 \newcommand{\add@Xgroupbyline}{%
2997   \unless\ifnocritical{%
2998     \def\do##1{%Looping on the series
2999       \let\olddo\do%Save the old \do macro, that is this macro itself!
```

```

3000 \def\do####1{%Looping on the ##1@forinserting command
3001 \ifcsdef{##1@forinserting@####1}{%
3002 \X@beforeinsertion{##1}%
3003 \if@ledgroup%
3004 \global\setbox\@nameuse{mp##1footins}=\vbox%
3005 \else%
3006 \insert\csname ##1footins\endcsname%
3007 \fi%
3008 {%
3009 \Xsetparindent{##1}%
3010 \ifcsdef{Xhsize\csuse{series@display##1}@##1}%
3011 {\hsize \csuse{Xhsize\csuse{series@display##1}@##1}}%
3012 {}%
3013 \if@ledgroup%
3014 \unvbox\@nameuse{mp##1footins}%
3015 \fi%
3016 \X@atbegininsertion{##1}%
3017 \ifcsstring{series@display##1}{normal}%
3018 {%
3019 \Xledsetnormalparstuff{##1}%
3020 \rule\z@\splittopskip%
3021 }%
3022 {}%
3023 \csuse{##1@forinserting@####1}%
3024 \strut\par\allowbreak%
3025 }%
3026 \global\csundef{##1@forinserting@####1}%
3027 }%
3028 {}%
3029 }%
3030 \ifcsdef{##1@forinserting}{%
3031 \dolistcsloop{##1@forinserting}%
3032 }{}%
3033 \global\csundef{##1@forinserting}%
3034 \let\do\olddo%Restore old do
3035 }%
3036 \dolistloop{\@series}%
3037 \fi%
3038 }%
3039
3040
3041 %

```

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@l@tempcnta is used to calculate and accumulate the penalty; it is initially set to the value of

`\ballast@count`, which has been worked out in `\do@ballast` above (VIII p. 159). Finally, the penalty is checked to see that it does not go below -10000 .

```

3042 \newcommand*{\add@penalties}{\@l@tempcnta=\ballast@count
3043   \ifnum\num@lines>\@ne
3044     \global\advance\par@line \@ne
3045     \ifnum\par@line=\@ne
3046       \advance\@l@tempcnta \clubpenalty
3047     \fi
3048     \@l@tempcntb=\par@line \advance\@l@tempcntb \@ne
3049     \ifnum\@l@tempcntb=\num@lines
3050       \advance\@l@tempcnta \widowpenalty
3051     \fi
3052     \ifnum\par@line<\num@lines
3053       \advance\@l@tempcnta \interlinepenalty
3054     \fi
3055   \fi
3056   \ifnum\@l@tempcnta=\z@
3057     \relax
3058   \else
3059     \ifnum\@l@tempcnta>-10000
3060       \penalty\@l@tempcnta
3061     \else
3062       \penalty -10000
3063     \fi
3064   \fi}
3065
3066 %

```

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

```

3067 \newcommand*{\flush@notes}{%
3068   \iftoggle{notfirstrun@}\jobname.\extensionchars\the\section@num}{%
3069     \xloop%
3070     \ifx\inserts@list\empty \else%
3071       \gl@p\inserts@list\to\@insert%
3072       \@insert%
3073       \global\let\@insert=\undefined%
3074     \repeat%
3075   }{}%
3076 }%
3077
3078

```

3079 %

\@xloop \@xloop is a variant of the PLAIN T_EX \loop macro, useful when it's hard to construct a positive test using the T_EX \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 T_EX \loop is used, too, so we could just call it \loop; but it seems preferable not to change the definitions of any of the standard macros.)

This variant of \loop was introduced by Alois Kabsch in *TUGboat* 8 (1987), pp. 184–5.

```
3080 \def\@xloop#1\repeat{%
3081   \def\body{#1\expandafter\body\fi}%
3082   \body}
3083
3084 %
```

XII.4 Text before notes

\set@Xtxtbeforenotes The \set@Xtxtbeforenotes macro resets the Xtxtbeforenotes@⟨series⟩@typeset boolean to false. Just before the first note of the ⟨series⟩ in a page, the \Xtxtbeforenotes will be inserted.

```
3085 \newcommand{\set@Xtxtbeforenotes}{%
3086   \unless\ifnocritical@%
3087   \def\do##1{%
3088     \nottoggle{Xtxtbeforenotesonlyonce@##1}{%
3089       \global\togglefalse{Xtxtbeforenotes@##1@typeset}%
3090     }{}%
3091   }%
3092   \dolistloop{\@series}%
3093 \fi%
3094 }%
3095 %
```

\set@txtbeforenotesX The \set@txtbeforenotesX does the same for the \txtbeforenotesX.

```
3096 \newcommand{\set@txtbeforenotesX}{%
3097   \unless\ifnofamiliar@%
3098   \def\do##1{%
3099     \nottoggle{txtbeforenotesonlyonceX@##1}{%
3100       \global\togglefalse{txtbeforenotesX@##1@typeset}%
3101     }{}%
3102   }%
3103   \dolistloop{\@series}%
3104 \fi%
3105 }%
3106 %
```

\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.

```

3107 \newcommand{\insert@Xtxtbeforenotes}[1]{%
3108 \nottoggle{Xtxtbeforenotes@#1@typeset}{%
3109 \global\toggletrue{Xtxtbeforenotes@#1@typeset}%
3110 \ifcvoid{Xtxtbeforenotes@#1}{}%
3111 \ifcsstring{series@display#1}{paragraph}%
3112 {\noindent\csuse{Xtxtbeforenotes@#1}}%
3113 {\expandafter\insert\csname#1footins\endcsname%
3114 \bgroup%
3115 \noindent%
3116 \ifcsdef{\csuse{series@display#1}@begin@insert}{%
3117 \csuse{\csuse{series@display#1}@begin@insert}{#1}%
3118 }{}%
3119 \strut\csuse{Xnotefontsize@#1}%
3120 \csuse{Xtxtbeforenotes@#1}%
3121 \egroup%
3122 }%
3123 }%
3124 }%
3125 {}%
3126 }%
3127
3128
3129 \newcommand{\insert@txtbeforenotesX}[1]{%
3130 \nottoggle{txtbeforenotesX@#1@typeset}{%
3131 \global\toggletrue{txtbeforenotesX@#1@typeset}%
3132 \ifcvoid{txtbeforenotesX@#1}{}%
3133 \ifcsstring{series@displayX#1}{paragraph}%
3134 {\noindent\csuse{txtbeforenotesX@#1}}%
3135 {\expandafter\insert\csname footins#1\endcsname%
3136 \bgroup%
3137 \noindent%
3138 \ifcsdef{\csuse{series@displayX#1}@begin@insert}{%
3139 \csuse{\csuse{series@displayX#1}@begin@insert}{#1}%
3140 }{}%
3141 \strut\csuse{notefontsizeX@#1}\csuse{txtbeforenotesX@#1}%
3142 \egroup%
3143 }%
3144 }%
3145 }%
3146 {}%
3147 }%
3148
3149
3150 %

```

XIII Critical footnotes

The footnote macros are adapted from those in PLAIN T_EX, 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.

```

3151 \def\select@lemmafont#1|#2|#3|#4|#5|#6|#7|{\select@lemmafont#7|}
3152 \def\select@lemmafont#1|#2|#3|#4|{%
3153   {\fontencoding{#1}\fontfamily{#2}\fontseries{#3}\fontshape{#4}%
3154    \selectfont}
3155   %
3156   %

```

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.

```

3157 \newcommand*{\footnoteoptions@}[3]{%
3158   \def\do##1{%
3159     \ifstrequal{#1}{L}{% On the left side
3160       \xright@appenditem{\noexpand\setkeys[mac]{#3footnoteoption}{\
3161         unexpanded{##1}}}{\to\inserts@list}%
3162       \global\advance\insert@count \@ne% Increment the left insert
3163       counter.
3164     }%
3165     \xright@appenditem{\noexpand\setkeys[mac]{#3footnoteoption}{\
3166       unexpanded{##1}}}{\to\inserts@listR}%
3167     \global\advance\insert@countR \@ne% Increment the right insert
3168     counter insert.
3169   }%
3170   %
3171   \notblank{#2}{\docsvlist{#2}}{}% Parsing all options
3172 }
3173 %

```

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.

```

3171 \newcommandx*{\footnotelang@lua}[1][1=L,usedefault]{%
3172   \ifstrequal{#1}{L}{%
3173     \xright@appenditem{{\csxdef{footnote@luatextextdir}{\the\textdir}}}{\to\
3174     inserts@list}% Know the dir of lemma
3175     \global\advance\insert@count \@ne%
3176     \xright@appenditem{{\csxdef{footnote@luatexpardir}{\the\pardir}}}{\to\
3177     inserts@list}% Know the dir of lemma

```



```

3176 \global\advance\insert@count \@ne%
3177 }%
3178 {%
3179 \xright@appenditem{\csxdef{footnote@luatexttextdir}{\the\textdir}}\to\
inserts@listR%Know the dir of lemma
3180 \global\advance\insert@countR \@ne%
3181 \xright@appenditem{\csxdef{footnote@luatexpardir}{\the\pardir}}\to\
inserts@listR%Know the dir of lemma
3182 \global\advance\insert@countR \@ne%
3183 }%
3184 }
3185 %

```

`\footnotelang@poly` `\footnotelang@poly` is called to remember the information about the language of a lemma when polyglossia is used.

```

3186 \newcommand*{\footnotelang@poly}[1][1=L,usedefault]{%
3187 \ifstrequal{#1}{L}{%
3188 \if@RTL%
3189 \xright@appenditem{\csxdef{footnote@dir}{@RTLtrue}}\to\
inserts@list%Know the language used in the lemma
3190 \global\advance\insert@count \@ne%
3191 \else
3192 \xright@appenditem{\csxdef{footnote@dir}{@RTLfalse}}\to\
inserts@list%Know the language of lemma
3193 \global\advance\insert@count \@ne%
3194 \fi%
3195 \xright@appenditem{\csxdef{footnote@lang}{\expandonce\language\language}}\
to\inserts@list%Know the language of lemma
3196 \global\advance\insert@count \@ne%
3197 }%
3198 {%
3199 \if@RTL
3200 \xright@appenditem{\csxdef{footnote@dir}{@RTLtrue}}\to\
inserts@listR%Know the language of lemma
3201 \global\advance\insert@countR \@ne%
3202 \else
3203 \xright@appenditem{\csxdef{footnote@dir}{@RTLfalse}}\to\
inserts@listR%Know the language of lemma
3204 \global\advance\insert@countR \@ne%
3205 \fi
3206 \xright@appenditem{\csxdef{footnote@lang}{\expandonce\language\language}}\
to\inserts@listR%Know the language of lemma
3207 \global\advance\insert@countR \@ne%
3208 }%
3209 }
3210 %

```

XIII.4 General survey of the way we manage notes

The processing of each note is done by four principal macros: the `\vfootnote` macro takes the text of the footnote and does the `\insert`; it calls on the `\footfmt` macro to select the right fonts, print the line number and lemma, and do any other formatting needed for that individual note. Within the output routine, the two other macros,

`\footstart` and `\footgroup`, are called; the first prints extra vertical space and a footnote rule, if desired; the second does any reformatting of the whole set of the footnotes in this series for this page—such as paragraphing or division into columns—and then sends them to the page.

These four macros, and the other macros and parameters shown here, are distinguished by the “series letter” that indicates which set of the footnotes we are dealing with—A, B, C, D, or E. The series letter always precedes the string `foot` in macro and parameter names. Hence, for the A series, the four macros are called `\vAfootnote`, `\Afootfmt`, `\Afootstart`, and `\Afootgroup`.

These macros are changed depending on the footnote arrangement: “normal”, “paragraphed”, “two columns” or “three columns”.

XIII.5 General setup

`\footsplitskips` Some setup code that is common for a variety of the footnotes. The setup is for:

- `\interlinepenalty`.
- `\splittopskip` (skip before last part of notes that flow from one page to another).
- `\splitmaxdepth`.
- `\floatingpenalty`, that is penalty values being added when a long note flows from one page to another. Here, we let it to 0 when we process parallel pages in `eledpar`, in order to allow notes to flow from left to right pages and *vice-versa*. Otherwise, we let it to `\@MM`, which is the standard \TeX `\floatingpenalty`.

```

3211 \newcommand*{\footsplitskips}{%
3212   \interlinepenalty=\interfootnotelinepenalty
3213   \unless\ifl@dprintingpages%
3214     \floatingpenalty=\@MM%
3215   \fi%
3216   \splittopskip=\ht\strutbox \splitmaxdepth=\dp\strutbox
3217   \leftskip=\z@skip \rightskip=\z@skip}
3218
3219 %
```

`\normalfootnoterule` `\normalfootnoterule` is a standard footnote-rule macro, for use by a `footstart` macro: just the same as the `PLAIN` \TeX footnote rule.

```

3220 \let\normalfootnoterule=\footnoterule
3221 %
```

XIII.6 Footnotes arrangement

XIII.6.1 User level macro

`\Xarrangement` `\Xarrangement[$\langle s \rangle$]{ $\langle arrangement \rangle$ }` The command calls, for each series, a specific command which set many counters and commands in order to define specific arrangement.

```

3222 \newcommand{\Xarrangement}[2][1,usedefault]{%
3223   \def\do##1{%
3224     \csname Xarrangement@#2\endcsname{##1}%
3225   }%
3226   \ifstrempy{#1}%
3227     {%
3228       \dolistloop{\@series}%
3229     }%
3230     {
3231       \docsvlist{#1}%
3232     }%
3233   }%
3234   %

```

XIII.6.2 Normal footnote

`\Xarrangement@normal` We can now define all the parameters for the series of footnotes; initially they use the “normal” footnote formatting.

What we want to do here is to insert something like the following for each footnote series. (This is an example, not part of the actual `reledmac` code.)

```

\skip\Afootins=12pt plus5pt minus5pt
\count\Afootins=1000
\dimen\Afootins=0.8\vsiz
\let\vAfootnote=\normalvfootnote \let\Afootfmt=\normalfootfmt
\let\Afootstart=\normalfootstart \let\Afootgroup=\normalfootgroup
\let\Afootnoterule=\normalfootnoterule

```

(Read *The TeXbook* in order to understand what are the counter, skip and dimen associated to an insertion.)

Instead of repeating ourselves, we define a `\Xarrangement@normal` macro that makes all these assignments for us, for any given series letter. This command is called when people use `\Xarrangement[series]{normal}`

Now we set up the `\Xarrangement@normal` macro itself. It takes one argument: the footnote series letter.

```

3235 \newcommand*{\Xarrangement@normal}[1]{%
3236   \csgdef{series@display#1}{normal}
3237   \expandafter\let\csname #1footstart\endcsname=\normalfootstart
3238   \expandafter\let\csname v#1footnote\endcsname=\normalvfootnote
3239   \expandafter\let\csname #1footfmt\endcsname=\normalfootfmt
3240   \expandafter\let\csname #1footgroup\endcsname=\normalfootgroup
3241   \expandafter\let\csname #1footnoterule\endcsname=%
3242                                     \normalfootnoterule
3243   \count\csname #1footins\endcsname=1000
3244   \dimen\csname #1footins\endcsname=\csuse{Xmaxhnotes@#1}
3245   \skip\csname #1footins\endcsname=\csuse{Xbeforenotes@#1}%
3246   \advance\skip\csname #1footins\endcsname by\csuse{Xafterrule@#1}%
3247   %

```

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.

```

3248 \csxdef{default@#1footins}{1000}%Use this to confine the notes to one
side only
3249 %

```

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).

```

3250 \ifnoledgroup@else%
3251 \expandafter\let\csname mpv#1footnote\endcsname=\mpnormalvfootnote
3252 \expandafter\let\csname mp#1footgroup\endcsname=\mpnormalfootgroup
3253 \count\csname mp#1footins\endcsname=1000
3254 \dimen\csname mp#1footins\endcsname=\csuse{Xmaxhnotes@#1}
3255 \skip\csname mp#1footins\endcsname=\csuse{Xbeforenotes@#1}%
3256 \advance\skip\csname mp#1footins\endcsname by\csuse{Xafterrule@#1}%
3257 \fi
3258 }
3259
3260 %

```

\normalvfootnote We now begin a series of commands that do “normal” footnote formatting: a format much like that implemented in PLAIN T_EX, 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.

```

3261 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\normalvfootnote}[2]{%
3262 \iftoggle{Xgroupbyline@#1}{%In the case we use \Xgroupbyline, the
insertion is done later, in \add@Xgroupbyline.
3263 \prepare@Xgroupbyline{#1}{#2}{\normalvfootnote@inserted}%
3264 }{%In the case we don't use \Xgroupbyline, the insertion is made directly
\X@beforeinsertion{#1}%
3265 \insert\csname #1footins\endcsname{%
3266 \X@atbegininsertion{#1}%
3267 \normalvfootnote@inserted{#1}{#2}%
3268 }%
3269 }%
3270 }%
3271 }%
3272 %

```

\normalvfootnote@inserted The **\normalvfootnote@inserted** macro is expanded to the content to be added to a **\insert** for normal critical footnote.

```

3273 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\normalvfootnote@inserted}[2]{%
3274 \nottoggle{Xgroupbyline@#1}{\noindent}{\csuse{Xhooknote@#1}%
3275 \csuse{Xnotefontsize@#1}%
3276 \iftoggle{Xgroupbyline@#1}{\strut}{}%
3277 \footsplitskips
3278 \ifl@dpairing\ifl@dpaging\else%
3279 \setXnoteswidthliketwocolumns@{#1}%
3280 \fi\fi%
3281 \setXnotespositionliketwocolumns@{#1}%
3282 \spaceskip=\z@skip \xspaceskip=\z@skip%
3283 \csname #1footfmt\endcsname #2{#1}%

```

```

3284 }%
3285 %

```

```

\X@beforeinsertion86 \newcommand{\X@beforeinsertion}[1]{%
3287   \if@ledgroup\else%
3288   \insert@Xtxtbeforenotes{#1}%
3289   \fi%
3290   \csuse{Xbeforeinserting@#1}%
3291 }%
3292 %

```

```

\beforeinsertion@X93 \newcommand{\beforeinsertion@X}[1]{%
3294   \if@ledgroup\else%
3295   \insert@txtbeforenotesX{#1}%
3296   \fi%
3297   \csuse{beforeinsertingX@#1}%
3298 }%
3299 %

```

```

\X@atbegininsertion90 \newcommand{\X@atbegininsertion}[1]{%
3301   \hsize=\expandafter\dimexpr\csuse{Xwidth@#1}\relax%
3302 }%
3303 %

```

And somewhat different versions of `\normalvfootnote` and `\normalvfootnote@inserted` for minipages.

```

\mpnormalvfootnote84 \notbool{parapparatus@}\newcommand*{\newcommand}{\mpnormalvfootnote}[2]{%
3305   \iftoggle{Xgroupbyline@#1}{%
3306     \prepare@Xgroupbyline{#1}{#2}{\mpnormalvfootnote@inserted}%
3307   }%
3308   {%
3309     \global\setbox\@nameuse{mp#1footins}%
3310     \vbox{%
3311       \unvbox\@nameuse{mp#1footins}%
3312       \mpnormalvfootnote@inserted{#1}{#2}%
3313     }%
3314   }%
3315 }%
3316 %
3317 %

```

```

normalvfootnote@inserted18 \newcommand{\mpnormalvfootnote@inserted}[2]{%
3319   \noindent\csuse{Xbhooknote@#1}%
3320   \csuse{Xnotefontsize@#1}%
3321   \hsize\columnwidth%
3322   \@parboxrestore%
3323   \color@begingroup%
3324   \csname #1footfmt\endcsname #2{#1}\color@endgroup%
3325 }%
3326 %

```

`\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.

```

3327 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\normalfootfmt}[4]{%
3328   \xstorelineinfo{#1}{#4}%
3329   \nottoggle{Xgroupbyline@#4}{\Xledsetnormalparstuff{#4}}{}%
3330   \Xsethangindent{#4}%
3331   \nottoggle{Xgroupbyline@#4}{\rule{z@}{\splittopskip}}{}%
3332   {\printlinefootnote{#1}{#4}}%
3333   \print@lemma{#1}{#2}{#4}%
3334   \csuse{Xwrapcontent@#4}{#3}%
3335   \nottoggle{Xgroupbyline@#4}{\strut\par}{}%
3336 }%
3337 %

```

`\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`.

```

3338 \newcommand*{\normalfootstart}[1]{%
3339 %

```

The first series of notes printed in a page can have a specific skip before it. In order to insert this specific skip without overlap the bottom margin of the page, Maïeul Rouquette have defined an algorithm explained in XIX p. 235. Here is part of this algorithm, when the block of notes are ready to be printed.

```

3340 \ifdimequal{0pt}{\Xprenotes@}{}%
3341 {%
3342   \iftoggle{Xprenotes@}{%
3343     \togglefalse{Xprenotes@}%
3344     \skip\csname #1footins\endcsname=%
3345     \glueexpr\csuse{Xprenotes@}+\csuse{Xafterrule@#1}\relax%
3346   }%
3347   {}%
3348 }%
3349 \vskip\skip\csname #1footins\endcsname%

```

```
3350 %
```

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.

```
3351 \leftskip0pt \rightskip0pt
3352 \ifl@dpairing\else%
3353   \hsize=\old@hsize%
3354 \fi%
3355 \setXnoteswidthliketwocolumns@{#1}%
3356 \setXnotespositionliketwocolumns@{#1}%
3357 %
```

And now, print the footnote's rule to finish the footnote's introduction.

```
3358 \print@Xfootnoterule{#1}%
3359 }%
3360 %
```

\normalfootgroup `\normalfootgroup` is a standard footnote-grouping macro: it sends the contents of the footnote-insert box to the output page without alteration.

```
3361 \newcommand*{\normalfootgroup}[1]{%
3362   \csuse{Xbhookgroup@#1}%
3363   \unvbox\curname #1footins\endcurname%
3364   \hsize=\old@hsize%
3365 }%
3366 %
3367 %
```

\mpnormalfootgroup A somewhat different version for minipages. Note that, in this case, we do not make distinctions between the `\Xfootgroup` and `\Xfootstarts` macros.

```
3368 \unless\ifnoledgroup@
3369 \newcommand*{\mpnormalfootgroup}[1]{%
3370   \vskip\skip\@nameuse{mp#1footins}
3371   \ifl@dpairing\ifparledgroup%
3372     \leavevmode\marks\parledgroup@{begin}%
3373     \marks\parledgroup@series{#1}%
3374     \marks\parledgroup@type{Xfootnote}%
3375   \fi\fi\normalcolor%
3376   \ifparledgroup%
3377     \ifl@dpairing%
3378     \else%
3379       \setXnoteswidthliketwocolumns@{#1}%
3380       \setXnotespositionliketwocolumns@{#1}%
3381       \print@Xfootnoterule{#1}%
3382     \fi%
3383   \else%
3384     \setXnoteswidthliketwocolumns@{#1}%
3385     \setXnotespositionliketwocolumns@{#1}%
3386     \print@Xfootnoterule{#1}%
3387   \fi%
3388   \setlength{\parindent}{0pt}
3389   \csuse{Xbhookgroup@#1}%

```

```

3390 \unvbox\csname mp#1footins\endcsname}}
3391 \fi
3392 %

```

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.

```

3393 \newcommand*{\Xarrangement@paragraph}[1]{%
3394   \csgdef{series@display#1}{paragraph}
3395   \expandafter\let\csname #1footstart\endcsname=\parafootstart
3396   \expandafter\let\csname v#1footnote\endcsname=\paravfootnote
3397   \expandafter\let\csname #1footfmt\endcsname=\parafootfmt
3398   \expandafter\let\csname #1footgroup\endcsname=\parafootgroup
3399   \count\csname #1footins\endcsname=1000
3400   \csxdef{default@#1footins}{1000}%Use this to confine the notes to one
side only
3401   \dimen\csname #1footins\endcsname=\csuse{Xmaxhnotes@#1}
3402   \skip\csname #1footins\endcsname=\csuse{Xbeforenotes@#1}%
3403   \advance\skip\csname #1footins\endcsname by\csuse{Xafterrule@#1}%
3404   \para@footsetup{#1}
3405   %

```

And the extra setup for minipages.

```

3406   \ifnoledgroup@else
3407     \expandafter\let\csname mpv#1footnote\endcsname=\mpparavfootnote
3408     \expandafter\let\csname mp#1footgroup\endcsname=\mpparafootgroup
3409     \count\csname mp#1footins\endcsname=1000
3410     \dimen\csname mp#1footins\endcsname=\csuse{Xmaxhnotes@#1}
3411     \skip\csname mp#1footins\endcsname=\csuse{Xbeforenotes@#1}%
3412     \advance\skip\csname mp#1footins\endcsname by\csuse{Xafterrule@#1}%
3413     \fi
3414   }
3415   %

```

`\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.

```

3416 \providecommand{\footfudgefiddle}{64}
3417 %

```


`\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`.

```

3418 \newcommand*{\para@footsetup}[1]{\csuse{Xhookgroup@#1}\csuse{
Xnotefontsize@#1}
3419 \setXnoteswidthliketwocolumns@{#1}%
3420 \ifcempty{Xwidth@#1}%
3421 {}%
3422 {\columnwidth=\expandafter\dimexpr\csuse{Xwidth@#1}\relax}%
3423 \dimen0=\baselineskip
3424 \multiply\dimen0 by 1024
3425 \divide \dimen0 by \columnwidth \multiply\dimen0 by \footfudgefiddle\
relax
3426 \csxdef{#1footfudgefactor}{%
3427 \expandafter\strip@pt\dimen0 }}
3428
3429 %

```

`\strip@pt` strip the characters pt from a dimen value.

`\parafootstart` `\parafootstart` is the same as `\normalfootstart`, but we give it again to ensure that `\rightskip` and `\leftskip` are zeroed (this needs to be done before `\para@footgroup` in the output routine). The size of paragraphed notes is calculated using a fudge factor which in turn is based on `\hsize`. So the paragraph of notes needs to be that wide.

The argument of the macro is again the note series letter.

```

3430 \newcommand*{\parafootstart}[1]{%
3431 \rightskip=0pt \leftskip=0pt%
3432 \Xsetparindent{#1}%
3433 \ifdimequal{0pt}{\Xprenotes@}{}%
3434 {%
3435 \iftoggle{Xprenotes@}{%
3436 \togglefalse{Xprenotes@}%
3437 \skip\csname #1footins\endcsname=%
3438 \glueexpr\csuse{Xprenotes@}+\csuse{Xafterrule@#1}\relax%
3439 }%
3440 }%
3441 }%
3442 \vskip\skip\csname #1footins\endcsname%
3443 \setXnoteswidthliketwocolumns@{#1}%
3444 \setXnotespositionliketwocolumns@{#1}%
3445 \print@Xfootnoterule{#1}%
3446 \let\old@bidi@RTL@everypar\bidi@RTL@everypar%
3447 \let\bidi@RTL@everypar\empty%
3448 \noindent\leavevmode%
3449 \let\bidi@RTL@everypar\old@bidi@RTL@everypar%
3450 }%
3451 %

```

`\paravfootnote` `\paravfootnote` is a version of the `\vfootnote` command that is used for paragraphed notes. It gets appended to the `\inserts@list` list by an outer-level footnote command like `\Afootnote`. The first argument is the note series letter; the second is the full text of the printed note itself, including line numbers, lemmata, and footnote text.

The initial model for this insertion is, of course, the `\insert\footins` definition in *The TeXbook*, p. 398. There, the footnotes are first collected up in `hboxes`, and these `hboxes` are later unpacked and stuck together into a paragraph.

However, Michael Downes has pointed out that because text in `hboxes` gets typeset in restricted horizontal mode, there are some undesirable side-effects if you later want to break such text across lines. In restricted horizontal mode, where \TeX does not expect to have to break lines, it does not insert certain items like `\discretionary`s. If you later unbox these `hboxes` and stick them together, as the *TeXbook* macros do to make these footnotes, you lose the ability to hyphenate after an explicit hyphen. This can lead to overfull `\hboxes` when you would not expect to find them, and to the uninitiated it might be very hard to see why the problem had arisen.³²

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.³³ 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.³⁴ 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.³⁵ 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. 182 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.

3452 `\newcommand*{\paravfootnote}[2]{%`

³²Michael Downes, "Line Breaking in `\unhboxed` Text", *TUGboat* **11** (1990), pp. 605–612.

³³See *The TeXbook*, p. 455 (editions after January 1990).

³⁴Wayne 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.

³⁵"Line Breaking", p. 610.

```

3453 \csuse{Xbeforeinserting@#1}%
3454 \insert\csname #1footins\endcsname
3455 \bgroup
3456 \csuse{Xnotefontsize@#1}
3457 \footsplitskips
3458 \setbox0=\vbox{\hsize=\maxdimen%
3459 \let\bid@RTL@everypar\@empty%
3460 \insert@Xtxtbeforenotes{#1}%
3461 \noindent\csuse{Xhooknote@#1}%
3462 \csname #1footfmt\endcsname #2{#1}}%
3463 \setbox0=\hbox{\Xunvxh{0}{#1}}%
3464 \dp0=0pt
3465 \ht0=\csname #1footfudgefactor\endcsname\wd0
3466 %

```

Here we produce the contents of the footnote from box 0, and add a penalty of 0 between boxes in this insert.

```

3467 \if@RTL\noindent \leavevmode\fi\box0%
3468 \penalty0
3469 \egroup}
3470
3471 %

```

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.

```

3472 \newcommand*{\mpparavfootnote}[2]{%
3473 \global\setbox\@nameuse{mp#1footins}\vbox{%
3474 \unvbox\@nameuse{mp#1footins}%
3475 \csuse{Xnotefontsize@#1}
3476 \footsplitskips
3477 \setbox0=\vbox{\hsize=\maxdimen%
3478 \let\bid@RTL@everypar\@empty%
3479 \insert@Xtxtbeforenotes{#1}%
3480 \noindent\color@begingroup%
3481 \csuse{Xhooknote@#1}%
3482 \csname #1footfmt\endcsname #2{#1}\color@endgroup}%
3483 \setbox0=\hbox{\Xunvxh{0}{#1}}%
3484 \dp0=\z@
3485 \ht0=\csname #1footfudgefactor\endcsname\wd0
3486 \box0
3487 \penalty0
3488 }}
3489

```

3490 %

\Xunvxh Here is (modified) Michael’s definition of `\unvxh`, used above. Michael’s macro also takes care to remove some unwanted penalties and glue that T_EX automatically attaches to the end of paragraphs. When T_EX 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`.

```
3491 \newcommand*{\Xunvxh}[2]{%
3492   \setbox0=\vbox{\unvbox#1%
3493     \global\setbox1=\lastbox}%
3494   \unhbox1
3495   \unskip           % remove \rightskip,
3496   \unskip           % remove \parfillskip,
3497   \unpenalty        % remove \penalty of 10000,
3498   \hskip\csuse{Xafternote@#2}\relax}% add the glue to go between the notes
3499 %
3500 %
```

\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).

```
3501 \newcommand*{\parafootfmt}[4]{%
3502   \Xstorelineinfo{#1}{#4}%
3503   \Xinsertparafootsep{#4}%
3504   \ledsetnormalparstuff@common%
3505   \printlinefootnote{#1}{#4}%
3506   \print@lemma{#1}{#2}{#4}%
3507   \csuse{Xwrapcontent@#4}{#3}%
3508   \penalty-10 }
3509 %
```

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.

```
3510 \newcommand*{\parafootgroup}[1]{%
3511   \hsize=\expandafter\dimexpr\csuse{Xwidth@#1}\relax%
3512   \unvbox\csname #1footins\endcsname
3513   \ifcsstring{Xragged@#1}{L}{\RaggedLeft}{}%
3514   \ifcsstring{Xragged@#1}{R}{\RaggedRight}{}%
3515   \makehboxofhboxes
3516   \setbox0=\hbox{\unhbox0 \removehboxes}%

```

```

3517 \csuse{Xbhookgroup@#1}%
3518 \csuse{Xnotefontsize@#1}%
3519 \unhbox0\par%
3520 \global\hsize=\old@hsize%
3521 }%
3522
3523 %

```

`\mpparafootgroup` The minipage version.

```

3524 \newcommand*{\mpparafootgroup}[1]{%
3525   \setXnoteswidthliketwocolumns@{#1}%
3526   \vskip\skip\@nameuse{mp#1footins}
3527   \ifl@dpairing\ifparledgroup%
3528     \leavevmode\marks\parledgroup@{begin}%
3529     \marks\parledgroup@series{#1}%
3530     \marks\parledgroup@type{Xfootnote}%
3531     \fi\fi\normalcolor
3532   \ifparledgroup%
3533     \ifl@dpairing%
3534       \else%
3535         \setXnoteswidthliketwocolumns@{#1}%
3536         \setXnotespositionliketwocolumns@{#1}%
3537         \print@Xfootnoterule{#1}%%
3538       \fi%
3539     \else%
3540       \setXnoteswidthliketwocolumns@{#1}%
3541       \setXnotespositionliketwocolumns@{#1}%
3542       \print@Xfootnoterule{#1}%
3543     \fi%
3544     \unvbox\csname mp#1footins\endcsname
3545     \ifcsstring{Xragged@#1}{L}{\RaggedLeft}{}%
3546     \ifcsstring{Xragged@#1}{R}{\RaggedRight}{}%
3547     \makehboxofhboxes
3548     \setbox0=\hbox{\unhbox0 \removehboxes}%
3549     \csuse{Xbhookgroup@#1}%
3550     \csuse{Xnotefontsize@#1}%
3551     \Xsetparindent{#1}%
3552     \unhbox0\par}}
3553
3554 %

```

And finally, the two macros which are required to transform the long horizontal box stored in the insert' box to a printable text.

```

\makehboxofhboxes\newcommand*{\makehboxofhboxes}{\setbox0=\hbox{}}%
\removehboxes\loop
3557   \unpenalty
3558   \setbox2=\lastbox
3559   \ifhbox2
3560     \setbox0=\hbox{\box2\unhbox0}%
3561   \repeat}
3562

```

```

3563 \newcommand*{\removehboxes}{\setbox0=\lastbox
3564 \ifhbox0{\removehboxes}\unhbox0 \fi}
3565
3566 %

```

Insertion of the footnotes separator The command `\Xinsertparafootsep{<series>}` must be called at the beginning of `\parafootftm`.

```

\prevpage@num 3567 \newcommand{\Xinsertparafootsep}[1]{%
\Xinsertparafootsep 3568 \ifledRcol{%
3569 \ifnumequal{\csuse{#1prevpage@numR}}{\page@numR}%
3570 {\ifcsdef{prevline#1}% Be sur \prevline#1 exists.
3571 {\ifcsequal{prevline#1}{lineinfo@}%
3572 {\ifcsequal{Xsymlinenum@#1}{\csuse{Xparafootsep@#1}}{}}%
3573 {\csuse{Xparafootsep@#1}}}%
3574 }%
3575 {\csuse{Xparafootsep@#1}}%
3576 }%
3577 {}%
3578 \global\csname #1prevpage@numR\endcsname=\page@numR%
3579 \else%
3580 \ifnumequal{\csuse{#1prevpage@num}}{\page@num}%
3581 {\ifcsdef{prevline#1}% Be sur \prevline#1 exists.
3582 {\ifcsequal{prevline#1}{lineinfo@}%
3583 {\ifcsequal{Xsymlinenum@#1}{\csuse{Xparafootsep@#1}}{}}%
3584 {\csuse{Xparafootsep@#1}}}%
3585 }%
3586 {\csuse{Xparafootsep@#1}}%
3587 }%
3588 {}%
3589 \global\csname #1prevpage@num\endcsname=\page@num%
3590 \fi%
3591 }
3592 %

```

XIII.6.4 Columnar footnotes

Common tools

`\rigidbalance` We will now define macros for three-column notes and two-column notes. Both sets of macros will use `\rigidbalance`, which splits a box (#1) into a number (#2) of columns, each with a space (#3) between the top baseline and the top of the `\vbox`. The `\rigidbalance` macro is taken from *The TeXbook*, p. 397, with a slight change to the syntax of the arguments so that they do not depend on white space. Note also the extra unboxing in `\splitoff`, which allows the new `\vbox` to have its natural height as it goes into the alignment.

The \LaTeX `\line` macro has no relationship to the TeX `\line`. The \LaTeX equivalent is `\@@line`.

We do not call directly `\rigidbalance`, but we call `\Xrigidbalance` for critical notes and `\rigidbalanceX` for familiar notes. Both of them call `\rigidbalance`.

```

3593 \newcount\@k \newdimen\@h
3594 \newcommand*\Xrigidbalance}[3]{%
3595   \hsize=\expandafter\dimexpr\csuse{Xwidth@\@currentseries}\relax%
3596   \rigidbalance{#1}{#2}{#3}%
3597 }%
3598
3599 \newcommand*\XrigidbalanceX}[3]{%
3600   \hsize=\expandafter\dimexpr\csuse{widthX@\@currentseries}\relax%
3601   \rigidbalance{#1}{#2}{#3}%
3602 }%
3603
3604 \newcommand*\Xrigidbalance}[3]{%
3605   \setbox0=\box#1 \@k=#2 \@h=#3%
3606   \@@line{\splittopskip=\@h \vbadness=\@M \hfilneg
3607     \valign{##\vfil\cr\dosplits}}}}
3608
3609 \newcommand*\Xdosplits{\ifnum\@k>0 \noalign{\hfil}\splitoff
3610   \global\advance\@k-1\cr\dosplits\fi}
3611
3612 \newcommand*\Xsplitoff{\dimen0=\ht0
3613   \divide\dimen0 by\@k \advance\dimen0 by\@h
3614   \setbox2 \vsplit0 to \dimen0
3615   \unvbox2 }
3616
3617 %

```

Three columns

```

\Xarrangement@threecol 18 \newcommand*\Xarrangement@threecol}[1]{%
3619   \csgdef{series@display#1}{threecol}
3620   \expandafter\let\csname v#1footnote\endcsname=\threecolvfootnote
3621   \expandafter\let\csname #1footfmt\endcsname=\threecolfootfmt
3622   \expandafter\let\csname #1footgroup\endcsname=\threecolfootgroup
3623   \dimen\csname #1footins\endcsname=\csuse{Xmaxhnotes@#1}%
3624   \skip\csname #1footins\endcsname=\csuse{Xbeforenotes@#1}%
3625   \advance\skip\csname #1footins\endcsname by\csuse{Xafterrule@#1}%
3626   \threecolfootsetup{#1}
3627 %

```

The additional setup for minipages.

```

3628   \ifnoledgroup@else
3629     \expandafter\let\csname mpv#1footnote\endcsname=\mpnormalvfootnote
3630     \expandafter\let\csname mp#1footgroup\endcsname=\mpthreecolfootgroup
3631     \skip\csname mp#1footins\endcsname=\csuse{Xbeforenotes@#1}%
3632     \advance\skip\csname mp#1footins\endcsname by\csuse{Xafterrule@#1}%
3633     \mpthreecolfootsetup{#1}
3634   \fi
3635 }
3636
3637 %

```

The `\footstart` and `\footnoterule` macros for these notes assume the normal values (XIII.6.2 p. 182 above).

`\threecolfootsetup` The `\threecolfootsetup` macro calculates and sets some numbers for three-column footnotes.

We set the `\count` of the foot insert to 333. Each footnote can be thought of as contributing only one third of its height to the page, since the footnote insertion has been made as a long narrow column, which then gets trisected by the `\rigidbalance` routine (inside `\threecolfootgroup`). These new, shorter columns are saved in a box, and then that box is *put back* into the footnote insert, replacing the original collection of the footnotes. This new box is, therefore, only about a third of the height of the original one.

The `\dimen` value for this note series has to change in the inverse way: it needs to be three times the actual limit on the amount of space these notes are allowed to fill on the page, because when \TeX is accumulating material for the page and checking that limit, it does not apply the `\count` scaling.

```

3638 \newcommand*{\threecolfootsetup}[1]{%
3639   \count\csname #1footins\endcsname 333
3640   \csxdef{default@#1footins}{333}%Use this to confine the notes to one
side only
3641   \multiply\dimen\csname #1footins\endcsname \thr@@}
3642   %

```

`\mpthreecolfootsetup` The setup for minipages.

```

3643 \newcommand*{\mpthreecolfootsetup}[1]{%
3644   \count\csname mp#1footins\endcsname 333
3645   \multiply\dimen\csname mp#1footins\endcsname \thr@@}
3646   %
3647   %

```

`\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 #2 the full text of the note (including numbers, lemma and text).

```

3648 \notbool{parapparatus@}{\newcommand*{\threecolvfootnote}[2]{%
3649   \iftoggle{Xgroupbyline@#1}{%
3650     \prepare@Xgroupbyline{#1}{#2}{\threecolvfootnote@inserted}%
3651   }%
3652   {%
3653     \let\bidir@RTL@everypar\relax%
3654     \X@beforeinsertion{#1}%
3655     \insert\csname #1footins\endcsname{%
3656       \threecolvfootnote@inserted{#1}{#2}%
3657     }%
3658   }%
3659 }%
3660 %

```



```

\recolvfootnote@inserted 3661 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\
3662   threecolvfootnote@inserted}[2]{%
3663   \hspace{\dimexpr\csuse{Xwidth@#1}\relax%
3664   \noindent\csuse{Xbhooknote@#1}%
3665   \csuse{Xnotefontsize@#1}%
3666   \footplitskips%
3667   \csname #1footfmt\endcsname #2{#1}%
3668   }%
%
```

\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.

```

3669 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\threecolfootfmt}[4]{%
3670   \Xstorelineinfo{#1}{#4}%
3671   \threecol@begin@insert{#4}%
3672   \hspace{\parindent}%
3673   \printlinefootnote{#1}{#4}%
3674   \print@lemma{#1}{#2}{#4}%
3675   \csuse{Xwrapcontent@#4}{#3}%
3676   \nottoggle{Xgroupbyline@#4}%
3677   {\strut\par\allowbreak}%
3678   }%
3679 }%
3680 %
```

\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.

```

3681 \newcommand{\threecol@begin@insert}[1]{%
3682   \normal@pars%
3683   \nottoggle{Xgroupbyline@#1}%
3684   {\hspace \csuse{Xsizethreecol@#1}}%
3685   }%
3686   \Xsetparindent{#1}%
3687   \tolerance=5000%
3688   \Xsethangindent{#1}%
3689   \@tempdima=\parindent%
3690   \csuse{Xcolalign@#1}%
3691   \parindent=\@tempdima%
3692   \strut%
3693 }%
3694 %
```

\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.

```

3695 \newcommand*{\threecolfootgroup}[1]{%
3696   \begingroup%
3697   \csuse{Xhookgroup@#1}%
3698   \csuse{Xnotefontsize@#1}%
3699   \par%
3700   \splittopskip=\ht\strutbox%
3701   \expandafter%
3702   \Xrigidbalance\csname #1footins\endcsname \thr@@ \splittopskip%
3703   \endgroup%
3704 }%
3705 %

```

`\mpthreecolfootgroup` The setup for minipages.

```

3706 \newcommand*{\mpthreecolfootgroup}[1]{%
3707   \vskip\skip\@nameuse{mp#1footins}
3708   \ifl@dpairing\ifparledgroup%
3709     \leavevmode\marks\parledgroup@{begin}%
3710     \marks\parledgroup@series{#1}%
3711     \marks\parledgroup@type{Xfootnote}%
3712     \fi\fi\normalcolor
3713   \ifparledgroup%
3714     \ifl@dpairing%
3715     \else%
3716       \setXnoteswidthliketwocolumns@{#1}%
3717       \setXnotespositionliketwocolumns@{#1}%
3718       \print@Xfootnoterule{#1}%
3719     \fi%
3720   \else%
3721     \setXnoteswidthliketwocolumns@{#1}%
3722     \setXnotespositionliketwocolumns@{#1}%
3723     \print@Xfootnoterule{#1}%
3724   \fi%
3725   \csuse{Xhookgroup@#1}\par%
3726   \splittopskip=\ht\strutbox
3727   \expandafter
3728   \Xrigidbalance\csname mp#1footins\endcsname \thr@@ \splittopskip}}
3729
3730 %

```

Two columns

```

\Xarrangement@twocol 3731 \newcommand*{\Xarrangement@twocol}[1]{%
3732   \csgdef{series@display#1}{twocol}
3733   \expandafter\let\csname v#1footnote\endcsname=\twocolvfootnote
3734   \expandafter\let\csname #1footfmt\endcsname=\twocolfootfmt
3735   \expandafter\let\csname #1footgroup\endcsname=\twocolfootgroup
3736   \dimen\csname #1footins\endcsname=\csuse{Xmaxhnotes@#1}%
3737   \skip\csname #1footins\endcsname=\csuse{Xbeforenotes@#1}%
3738   \advance\skip\csname #1footins\endcsname by\csuse{Xafterrule@#1}%

```

```

3739 \twocolfootsetup{#1}
3740 %

```

The additional setup for minipages.

```

3741 \ifnoledgroup@else
3742 \expandafter\let\csname mpv#1footnote\endcsname=\mpnormalvfootnote
3743 \expandafter\let\csname mp#1footgroup\endcsname=\mptwocolfootgroup
3744 \skip\csname mp#1footins\endcsname=\csuse{Xbeforenotes@#1}%
3745 \advance\skip\csname mp#1footins\endcsname by\csuse{Xafterrule@#1}%
3746 \mptwocolfootsetup{#1}
3747 \fi
3748 }
3749
3750 %

```

`\twocolfootsetup` Here is a series of macros which are very similar to their three-column counterparts. In this case, each note is assumed to contribute only a half a line of text. And the notes are set in columns giving a gap between them of one tenth of the `\hsize`.

```

\twocolvfootnote
\twocolvfootnote@inserted
\twocolfootfmt
\twocolfootgroup
3751 \newcommand*{\twocolfootsetup}[1]{%
3752 \count\csname #1footins\endcsname 500
3753 \csxdef{default@#1footins}{500}%
3754 \multiply\dimen\csname #1footins\endcsname \tw@}
3755 %

```

```

3756 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\twocolvfootnote}[2]{%
3757 \iftoggle{Xgroupbyline@#1}{%
3758 \prepare@Xgroupbyline{#1}{#2}{\twocolvfootnote@inserted}%
3759 }{%
3760 \let\bidir@RTL@everypar\relax%
3761 \X@beforeinsertion{#1}%
3762 \insert\csname #1footins\endcsname{%
3763 \twocolvfootnote@inserted{#1}{#2}%
3764 }%
3765 }%
3766 }%
3767 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\twocolvfootnote@inserted}[2]{%
3768 \hsize=\expandafter\dimexpr\csuse{Xwidth@#1}\relax%
3769 \noindent\csuse{Xhooknote@#1}%
3770 \csuse{Xnotefontsize@#1}%
3771 \footsplitskips%
3772 \csname #1footfmt\endcsname #2{#1}%
3773 }%
3774 %

```

```

3775 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\twocolfootfmt}[4]{% 4th
arg is optional, for backward compatibility
3776 \Xstorelineinfo{#1}{#4}%
3777 \twocol@begin@insert{#4}%
3778 \hspace{\parindent}%
3779 \printlinefootnote{#1}{#4}%
3780 \print@lemma{#1}{#2}{#4}%
3781 \csuse{Xwrapcontent@#4}{#3}%

```

```

3782 \nottoggle{Xgroupbyline@#4}%
3783 {\strut\par\allowbreak}%
3784 }%
3785 }%
3786 %

3787 \newcommand{\twocol@begin@insert}[1]{%
3788 \normal@pars%
3789 \nottoggle{Xgroupbyline@#1}%
3790 {\hsize \csuse{Xhsizetwocol@#1}}%
3791 }%
3792 \Xsetparindent{#1}%
3793 \tolerance=5000%
3794 \Xsethangindent{#1}%
3795 \@tempdima=\parindent%
3796 \csuse{Xcolalign@#1}%
3797 \parindent=\@tempdima%
3798 \strut%
3799 }%
3800
3801 \newcommand*\{twocolfootgroup}[1]{%
3802 \begin@group%
3803 \csuse{Xhookgroup@#1}%
3804 \csuse{Xnotefontsize@#1}%
3805 \par%
3806 \splittopskip=\ht\strutbox%
3807 \expandafter%
3808 \Xrigidbalance\csname #1footins\endcsname \tw@ \splittopskip%
3809 \endgroup%
3810 }%
3811
3812 %

```

`\mptwocolfootsetup` The versions for minipages.

`\mptwocolfootgroup`

```

3813 \newcommand*\{mptwocolfootsetup}[1]{%
3814 \count\csname mp#1footins\endcsname 500
3815 \multiply\dimen\csname mp#1footins\endcsname \tw@}
3816 %

3817 \newcommand*\{mptwocolfootgroup}[1]{%
3818 \vskip\skip\@nameuse{mp#1footins}
3819 \ifl@dpairing\ifparledgroup%
3820 \leavevmode\marks\parledgroup@{begin}%
3821 \marks\parledgroup@series{#1}%
3822 \marks\parledgroup@type{Xfootnote}%
3823 \fi\fi\normalcolor
3824 \ifparledgroup%
3825 \ifl@dpairing%
3826 \else%
3827 \setXnoteswidthliketwocolumns@{#1}%
3828 \setXnotespositionliketwocolumns@{#1}%
3829 \print@Xfootnoterule{#1}%
3830 \fi%

```

```

3831 \else%
3832   \setXnoteswidthliketwocolumns@{#1}%
3833   \setXnotespositionliketwocolumns@{#1}%
3834   \print@Xfootnoterule{#1}%
3835 \fi%
3836 \csuse{Xbhookgroup@#1}\par%
3837 \splittopskip=\ht\strutbox
3838 \expandafter
3839 \Xrigidbalance\csname mp#1footins\endcsname \tw@ \splittopskip}}
3840
3841 %

```

XIII.7 Footnote paragraph indent

\Xsetparindent These two commands set the paragraph indentation of the footnotes, depending on the settings of the user.

\setparindentX

```

3842 \newcommand{\Xsetparindent}[1]{%
3843   \nottoggle{Xparindent@#1}{%
3844     \parindent=\z@%
3845   }%
3846   {%
3847     \ifdef{\parindent@beforestanza}{\parindent=\parindent@beforestanza}{}%
3848   }%
3849 }%
3850 \newcommand{\setparindentX}[1]{%
3851   \nottoggle{parindentX@#1}{%
3852     \parindent=\z@%
3853   }%
3854   {%
3855     \ifdef{\parindent@beforestanza}{\parindent=\parindent@beforestanza}{}%
3856   }%
3857 }%
3858 %

```

XIII.8 Footnote hanging indent

\Xsethangindent \Xsethangindent sets the hangindent for a critical footnote, while \sethangindentX does it for a familiar footnote.

\sethangindentX

```

3859 \newcommand{\Xsethangindent}[1]{%
3860   \if@RTL%
3861     \hangindent=-\csuse{Xhangindent@#1}%
3862     \everypar{\hangindent=-\csuse{Xhangindent@#1}}%
3863   \else%
3864     \hangindent=\csuse{Xhangindent@#1}%
3865     \everypar{\hangindent=\csuse{Xhangindent@#1}}%
3866   \fi%
3867 }%
3868 \newcommand{\sethangindentX}[1]{%
3869   \if@RTL%
3870     \hangindent=-\csuse{hangindentX@#1}%
3871     \everypar{\hangindent=-\csuse{hangindentX@#1}}%

```

```

3872 \else%
3873 \hangindent=\csuse{hangindentX@#1}%
3874 \everypar{\hangindent=\csuse{hangindentX@#1}}%
3875 \fi%
3876 }%
3877 %

```

XIII.9 Critical notes presentation

Here, we define some commons macro which are used in order to print a critical notes, that is a note with 1) line number 2) lemma 3) lemma separator 4) text associated to the lemma.

XIII.9.1 Font tools

`\endashchar` The fonts that are used for printing notes might not have the character mapping we expect: for example, the Computer Modern font that contains old-style numerals does not contain an en-dash or square brackets, and its period and comma are in odd locations. To allow use of the standard footnote macros with such fonts, we use the following macros for certain characters.

The `\endashchar` macro is simply an en-dash from the normal font and is immune to changes in the surrounding font. The same goes for the full stop. These two are used in `\printlines`. The right bracket macro is the same again; it crops up in `\normalfootfmt` and the other footnote macros for controlling the format of the footnotes.

Note that these commands are not directly called by `reledmac`, but are enclosed as default value of specific hooks. Consequently, people should not redefine them, but use instead the `\Xlinerangeseparator`, `\Xendlinrangeseparator`, `\Xsublinesep`, `\Xendsublinesep` and `\Xlemmaseparator` macros.

With `polyglossia`, each critical note has a `\footnote@lang` which shows the language of the lemma, and which can be used to switch the bracket from right to left.

```

3878 \def\endashchar{\textnormal{--}}
3879
3880 \newcommand*{\fullstop}{\textnormal{.}}
3881 \def\Xsublinesep@side{\fullstop}
3882
3883 \newcommand*{\rbracket}{\textnormal{%
3884   \csuse{text\csuse{footnote@lang}}{%
3885     \ifluatex%
3886       \ifdefstring{\footnote@luatextextdir}{TRT}{\thinspace[]}{\thinspace
3887     ]}%
3888     \else%
3889       \thinspace]}%
3890     \fi}%
3891   }%
3892 }
3893 %

```

XIII.9.2 Pstart number in footnote

`\printpstart` The `\printpstart` macro prints the pstart number for a note.

```

3894 \newcommand{\printpstart}[0]{%
3895   \ifboolexpr{bool{!@dpairing} or bool{!@dprintingpages} or bool{
!@dprintingcolumns}}{%
3896     \ifledRcol%
3897       \thepstartR%
3898     \else%
3899       \thepstartL%
3900     \fi%
3901   }{%
3902     \thepstart%
3903   }%
3904 }
3905 %

```

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).

```

3906 %
3907 \newcommand{\print@lemma}[3]{%
3908   \bgroup%
3909   \nottoggle{Xlemmadisablefontselection@#3}%
3910   {\select@lemmafont#1|}%
3911   {}%
3912   \bgroup%
3913   \csuse{Xlemmafont@#3}%Deprecated
3914   \csuse{Xwraplemma@#3}{#2}%
3915   \egroup%
3916   \egroup%
3917   \iftoggle{nosep@}{%
3918     \hskip\csuse{Xinplaceoflemmaseparator@#3}%
3919     \relax%
3920   }%
3921   {\ifcempty{Xlemmaseparator@#3}%
3922     {%
3923       \hskip\csuse{Xinplaceoflemmaseparator@#3}%
3924       \relax%
3925     }%
3926     {%
3927       \nobreak%
3928       \hskip\csuse{Xbeforelemmaseparator@#3}%
3929       \csuse{Xlemmaseparator@#3}%
3930       \hskip\csuse{Xafterlemmaseparator@#3}%
3931       \relax%
3932     }%
3933   }%
3934 }%
3935 %

```

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.

```

3936 \newcommand{\Xstorelineinfo}[2]{%
3937   \l@dp@rsefootspec#1}%
3938   \iftoggle{Xnumberonlyfirstintwolines@#2}{%
3939     \xdef\lineinfo@{\l@dparsedstartline - \l@dparsedstartsub - \
3940     \l@dparsedendline - \l@dparsedendsub}%
3941     }%
3942     {%
3943     \xdef\lineinfo@{\l@dparsedstartline - \l@dparsedstartsub}%
3944     }%
3945   }%
3946   %

```

\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`.

```

3946 \newcommand{\printlinefootnote}[2]{%
3947   \xdef\@currentseries{#2}%
3948   \iftoggle{nonum@}{%Try if the line number must printed for this specific
3949   not (by default, yes)
3950   \hspace{\csuse{Xinplaceofnumber@#2}}%
3951   }%
3952   {%
3953     \iftoggle{Xnonumber@#2}%Try if the line number must printed (by
3954     default, yes)
3955     {%
3956     \hspace{\csuse{Xinplaceofnumber@#2}}%
3957     }%
3958     {%\iftoggle{Xnumberonlyfirstinline@#2}% If for this series the
3959     line number must be printed only in the first time.
3960     {%
3961     \ifcsdef{prevline#2}%
3962     {%Be sure the \prevline exists.
3963     \ifcsequal{prevline#2}{\lineinfo@}%Try it
3964     {%
3965     \ifcsequal{Xsymlinenumber@#2}%
3966     {%
3967     \hspace{\csuse{Xinplaceofnumber@#2}}%
3968     }%
3969     {\printsymlinefootnotearea{#2}}%
3970     }%
3971     {\printlinefootnotearea{#1}{#2}%
3972     }%
3973   }%

```



```

3974         {%
3975         \printlinefootnotearea{#1}{#2}%
3976         }%
3977     }%
3978     {%
3979     \printlinefootnotearea{#1}{#2}%
3980     }%
3981     \csxdef{prevline#2}{\lineinfo@}%
3982     }%
3983 }%
3984 }%
3985 }%
3986 }
3987 %

```

\printsymlinefootnotearea This macro prints the space before the line symbol, changes the font, when prints the line symbol and the space after it.

```

3988 \newcommand{\printsymlinefootnotearea}[1]{%
3989   \hspace{\csuse{Xbeforesymlinenum@#1}}%
3990   \csuse{Xnotenumfont@#1}%
3991   \ifdimequal{\csuse{Xboxsymlinenum@#1}}{\z@}%
3992     {\csuse{Xsymlinenum@#1}}%
3993     {\hbox to \csuse{Xboxsymlinenum@#1}%
3994       {\csuse{Xsymlinenum@#1}\hfill}%
3995     }%
3996   \hspace{\csuse{Xaftersymlinenum@#1}}%
3997 }%
3998 %

```

\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.)

```

3999 \newcommand{\printlinefootnotearea}[2]{%
4000   \printXbeforenumber{#2}%
4001   \csuse{Xnotenumfont@#2}%
4002   \boxfootnotenumbers{#1}{#2}%
4003   \printXafternumber{#2}%
4004 }%
4005 %

```

\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.

```

4006 \newcommand{\boxfootnotenumbers}[2]{%
4007   \ifdimequal{\csuse{Xboxlinenum@#2}}{0pt}{%
4008     \printlinefootnotenumbers{#1}{#2}%
4009   }%
4010   {%
4011     \hbox to \csuse{Xboxlinenum@#2}%
4012     {%

```

```

4013         \IfSubStr{RC}{\csuse{Xboxlinenumalign@#2}}{\hfill}{}%
4014         \printlinefootnotenumbers{#1}{#2}%
4015         \IfSubStr{LC}{\csuse{Xboxlinenumalign@#2}}{\hfill}{}%
4016         }%
4017     }%
4018 }%
4019 %

```

\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.

```

4020 \newcommand{\printlinefootnotenumbers}[2]{%
4021     \get@pstartcounter%
4022     \ifboolexpr{%
4023         (togl{Xpstart@#2} and bool{numberpstart})%
4024         or togl{Xpstarteverytime@#2}%
4025         and not (%
4026             togl{Xpstartonlyfirst@#2}%
4027             and test {\ifnumequal{\csuse{Xprevpstart@#2}}{\current@pstart}}%
4028         )%
4029     }%
4030     {%
4031         \printpstart%
4032         \csuse{Xpstartseparator@#2}%
4033     }%
4034     \global\cslet{Xprevpstart@#2}{\current@pstart}%
4035     \get@stanzacounter%
4036     \ifboolexpr{%
4037         togl{Xstanza@#2}%
4038         and bool {numberstanza}%
4039         and not (%
4040             togl{Xstanzaonlyfirst@#2}%
4041             and test {\ifnumequal{\csuse{Xprevstanza@#2}}{\current@stanza}}%
4042         )%
4043     }%
4044     \printstanza%
4045     \csuse{Xstanzaseparator@#2}%
4046     }%
4047     \global\cslet{Xprevstanza@#2}{\current@stanza}%
4048     \iftoggle{Xonlypstart@#2}{-%
4049         \csuse{Xtxtbeforenumber@#2}%
4050         \printlines#1|\ifledRcol@{\@Rlineflag\fi}|}%
4051     }%
4052 }%

```

\get@pstartcounter A single macro to get the current pstart counter.

```

4053 \newcommand{\get@pstartcounter}[0]{%
4054     \ifboolexpr{bool{l@dprintingpages} or bool{l@dprintingcolumns}}%
4055     {%
4056         \ifledRcol%
4057             \global\numdef\current@pstart{\the\c@pstartR}%
4058         \else%

```

```

4059 \global\newcommand{\current@pstart}{\the\c@pstartL}%
4060 \fi%
4061 }%
4062 {%
4063 \global\newcommand{\current@pstart}{\the\c@pstart}%
4064 }%
4065 }%
4066 %

```

`\get@stanzacounter` The same to get the current stanza counter.

```

4067 \newcommand{\get@stanzacounter}[0]{%
4068 \ifbool{bool{!@dprintingpages} or bool{!@dprintingcolumns}}{%
4069 {%
4070 \ifl@Rcol%
4071 \global\newcommand{\current@stanza}{\the\c@stanzaR}%
4072 \else%
4073 \global\newcommand{\current@stanza}{\the\c@stanzaL}%
4074 \fi%
4075 }%
4076 {%
4077 \global\newcommand{\current@stanza}{\the\c@stanza}%
4078 }%
4079 }%
4080 %

```

`\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.)

```

4081 \newcommand{\printXbeforenumber}[1]{%
4082 \hspace{\csuse{Xbeforenumber@#1}}%
4083 }%
4084 %

```

`\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

```

4085 \newcommand{\printXafternumber}[1]{%
4086 \iftoggle{Xnonbreakableafternumber@#1}{\nobreak}{}%
4087 \hspace{\csuse{Xafternumber@#1}}%
4088 }%
4089 %

```

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@pnum90 \newif\ifl@d@pnum
\ifl@d@ssub91 \newif\ifl@d@ssub
\ifl@d@elin92 \newif\ifl@d@elin
\ifl@d@esl93 \newif\ifl@d@esl
\ifl@d@dash94 \newif\ifl@d@dash
\ifl@d@Xtwolines95 \newif\ifl@d@Xtwolines%
\ifl@d@Xmorethantwolines96 \newif\ifl@d@Xmorethantwolines%
\ifl@d@Xlinenumber97 \let\@annot@start@print\relax%
\@annot@start@print98 \let\@annot@end@print\relax%
\@annot@end@print99 \newif\ifl@d@Xlinenumber%
100 %

```

```

\l@dp@rsefootspec \l@dp@rsefootspec{<spec>}{<lemma>}{<text>} parses a footnote specification. <lemma>
\l@dp@rsefootspec and <text> are the lemma and text respectively. <spec> is the line and page num-
\l@dp@rsefootspec ber and lemma font specifier in \l@d@nums style format. The real work is done by
\l@dp@rsefootspec \l@dp@rsefootspec which defines macros holding the numeric values. In many cases,
\l@dp@rsefootspec this last command is called directly. Just a reminder of the arguments:
\l@dp@rsefootspec \printlines #1 | #2 | #3 | #4 | #5 | #6 | #7
\l@dp@rsefootspec \printlines start-page | line | subline | end-page | line | subline | fontflag
\l@dp@rsefootspec \l@dp@rsefootspec
4101 \newcommand*{\l@dp@rsefootspec}[3]{\l@dp@rsefootspec#1|}
4102 \def\l@dp@rsefootspec#1|#2|#3|#4|#5|#6|#7|{%
4103   \gdef\l@dp@rsefootspecstartpage{#1}%
4104   \gdef\l@dp@rsefootspecstartline{#2}%
4105   \gdef\l@dp@rsefootspecstartsub{#3}%
4106   \gdef\l@dp@rsefootspecendpage{#4}%
4107   \gdef\l@dp@rsefootspecendline{#5}%
4108   \gdef\l@dp@rsefootspecendsub{#6}%
4109 }
4110 %

```

Initialise the several number value macros.

```

4111 \def\l@dparsedstartpage{0}%
4112 \def\l@dparsedstartline{0}%
4113 \def\l@dparsedstartsub{0}%
4114 \def\l@dparsedendpage{0}%
4115 \def\l@dparsedendline{0}%
4116 \def\l@dparsedendsub{0}%
4117
4118 %

```

\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`.

```

4119 \newcommand*\setprintlines[6]{%
4120   \let\@annot@start@print\relax%
4121   \let\@annot@end@print\relax%
4122   \l@d@pnumfalse%
4123   \l@d@dashfalse%
4124   \l@d@elinfalse%
4125 %

```

Should we print the line number? It depends on hooks and annotations.

```

4126   \l@d@Xlinenumbertrue%
4127   \ifboolexpr{togl{Xnolinenumber@\@currentseries}}%
4128     or (%
4129       togl{Xnolinenumberifannotation@\@currentseries}%
4130       and not test{\ifdefvoid{\annot@start}}%
4131       and not test{\ifdefvoid{\annot@end}}%
4132     )%
4133   }{%
4134     \l@d@Xlinenumberfalse%
4135   }{%
4136 %

```

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.

```

4137   \ifbypage@
4138     \ifnum#4=#1 \else
4139       \l@d@pnumtrue
4140       \l@d@dashtrue
4141     \fi
4142   \fi
4143 %

```

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.

```

4144   \ifboolexpr{%
4145     bool{\l@d@pnum}%

```

```

4146   or (%)
4147     not test{\ifnumequal{#2}{#5}}%
4148     and bool{l@d@Xlinenumber}%
4149   )%
4150 }{%
4151   \l@d@elintrue%
4152   \l@d@dashtrue%
4153   \unless\ifx\relax\annot@end%
4154     \def\@annot@end@print{%
4155       \l@wrapcs@ifnotemptybox{Xwraplinenumannotation@\@currentseries}{\
annot@end}%
4156     }%
4157   \fi%
4158 }{%}%
4159 %

```

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.

```

4160   \ifl@d@elin%
4161     \def\@annot@start@print{%
4162       \l@wrapcs@ifnotemptybox{Xwraplinenumannotation@\@currentseries}{\
annot@start}%
4163     }%
4164   \else%
4165     \ifx\annot@start\annot@end%
4166       \unless\ifx\@annot@start\relax%
4167         \def\@annot@start@print{%
4168           \l@wrapcs@ifnotemptybox{Xwraplinenumannotation@\@currentseries}{\
annot@start}%
4169         }%
4170       \fi%
4171     \else%
4172       \ifx\@annot@end@print\relax%
4173         \def\@annot@start@print{%
4174           \l@wrapcs@ifnotemptybox{Xwraplinenumannotation@\@currentseries}{%
\ifx\annot@start\empty\else%
4175             \annot@start%
4176             \ifdefined\linrangesep%
4177               \linrangesep%
4178             \else%
4179               \csuse{Xlinrangeseparator@\@currentseries}%
4180             \fi%
4181           \fi%
4182           \annot@end%
4183         }%
4184       }%
4185     \else%
4186       \let\@annot@start@print\@annot@end@print%
4187       \let\@annot@end@print\relax%
4188     \fi%
4189   \fi%
4190 \fi%
4191 %
4192 %

```

We print the starting sub-line if it is nonzero.

```

4193 \l@d@ssubfalse
4194 \ifnum#3=0 \else
4195     \l@d@ssubtrue
4196 \fi
4197 %

```

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.

```

4198 \l@d@eslfalse
4199 \ifnum#6=0 \else
4200     \ifnum#6=#3
4201         \ifl@d@elin \l@d@esltrue \else \l@d@eslfalse \fi
4202     \else
4203         \l@d@esltrue
4204         \l@d@dashtrue
4205     \fi
4206 \fi%
4207 %

```

However, if the `\Xtwolines` is set for the current series, we do not print the last line number.

```

4208 \ifl@d@dash%
4209     \ifboolexpr{togl{fulllines@} or test{\ifcempty{Xtwolines@}\
@currentseries}}}%
4210 {}%
4211 {%
4212     \setistwofollowinglines{#1}{#2}{#4}{#5}%
4213     \ifboolexpr{%
4214         (%
4215             togl {Xtwolinesbutnotmore@\@currentseries}%
4216             and not%
4217             (%
4218                 bool {istwofollowinglines@}%
4219             )%
4220         )%
4221     or%
4222     (%
4223         (not test{\ifnumequal{#1}{#4}})%
4224         and togl{Xtwolinesonlyinsamepage@\@currentseries}%
4225     )%
4226 }%
4227 {}%
4228 {%
4229     \l@d@dashfalse%
4230     \l@d@Xtwolinestrue%
4231     \l@d@elinfalse%
4232     \l@d@eslfalse%
4233     \ifcempty{Xmorethantwolines@\@currentseries}%
4234     {%
4235         {\ifistwofollowinglines@\else%
4236             \l@d@Xmorethantwolinestrue%
4237         \fi%

```

```

4238         }%
4239     }%
4240 }%
4241 \fi%
4242 %

```

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

```

4243 \iftoggle{Xnoidenticallinenumannotation@}\@currentseries}{%
4244     \ifx\annot@start\annot@end%
4245     \let\@annot@end@print\relax%
4246     \ifboolexpr{%Do not print the dash if we do not print the line number
4247         test {\ifdefempty{\linenumrep}}%
4248         or not bool {l@d@Xlinenumber}}%
4249     }{%
4250         \l@d@dashfalse%
4251     }{}%
4252 \fi%
4253 }{}%
4254 %

```

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.

```

4255 \iftoggle{Xlinenumannotationonlyfirst@}\@currentseries}{%
4256     \ifboolexpr{%
4257         (%
4258             togl{Xlinenumannotationonlyfirstintwo@}\@currentseries}%
4259             and test{\ifcsequal{annot@start}{prevannot@start@}\@currentseries}}%
4260             and test{\ifcsequal{annot@end}{prevannot@end@}\@currentseries}}%
4261         )%
4262         or%
4263         (%
4264             not togl{Xlinenumannotationonlyfirstintwo@}\@currentseries}%
4265             test{\ifcsequal{annot@start}{prevannot@start@}\@currentseries}}%
4266         )%
4267     }{%
4268         \def\@annot@start@print{%
4269             \l@wrapcs@ifnotemptybox{Xwraplinenumannotation@}\@currentseries}{\
csuse{Xsymlinenumannotation@}\@currentseries}}%
4270     }%
4271     \let\@annot@end@print\relax%
4272     \ifboolexpr{%Do not print the dash if we do not print the line number
4273         test {\ifdefempty{\linenumrep}}%
4274         or not bool {l@d@Xlinenumber}}%
4275     }{%
4276         \l@d@dashfalse%
4277     }{}%
4278 }{}%
4279 \global\cslet{prevannot@start@}\@currentseries}{\annot@start}%
4280 \global\cslet{prevannot@end@}\@currentseries}{\annot@end}%
4281 }%
4282 {}%
4283 %

```


End of `\setprintlines`.

```
4284 }%
4285 %
```

`\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.

```
4286 \newif\ifistwofollowinglines%
4287 \newcommand{\setistwofollowinglines}[4]{%
4288   \ifcsdef{lastlinenumberon#1}%
4289     {\numdef{\tmp}{\csuse{lastlinenumberon#1}}}%
4290     {\numdef{\tmp}{0}}%
4291   \istwofollowinglines@false%
4292   \ifnumequal{#4-#2}{1}%
4293     {\istwofollowinglines@true}%
4294     {\ifbypage%
4295       \ifnumequal{#3-#1}{1}%
4296       {%
4297         \ifnumequal{#2}{\tmp}%
4298           {\ifnumequal{#4}{1}{\istwofollowinglines@true}{}}%
4299           {}%
4300       }%
4301     }%
4302   \fi%
4303 }%
4304 }%
4305 %
```

`\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

`\printlines@count`

```

4306 \newcount\printlines@count%
4307 \def\printlines#1|#2|#3|#4|#5|#6|#7|#8|{%
4308   \global\advance\printlines@count by 1%
4309   \begingroup%
4310 %

```

Decide which part of line number components we will print.

```

4311 \setprintlines{#1}#{#2}#{#3}#{#4}#{#5}#{#6}%
4312 %

```

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.

```

4313 \ifdimequal{\csuse{Xboxstartlinenum@\@currentseries}}{0pt}%
4314   {\bgroup}%
4315   {\leavevmode\hbox to \csuse{Xboxstartlinenum@\@currentseries}\bgroup\
hfill}%
4316 \ifcsstring{Xlinenumannotationposition@\@currentseries}{before}%
4317   {\@annot@start@print}%
4318   {}%
4319 %

```

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.

```

4320 \ifl@d@pnum%
4321   \iftoggle{Xnopagenumberifcurrent@\@currentseries}{%
4322     \edlabel{printlines@the\printlines@count}%
4323   }{%
4324     \ifboolexpr{%
4325       not test{\ifnumequal{\xpageref{printlines@the\printlines@count}
}}{#1}}%
4326       or not togl{Xnopagenumberifcurrent@\@currentseries}%
4327     }%
4328     {%
4329       \csuse{Xbeforepagenumber@\@currentseries}%
4330       \wrap@edcrossref{\@this@crossref@start}{%
4331         #1%
4332       }%
4333       \ifl@d@Xlinenum%
4334         \csuse{Xpagelinesep@\@currentseries}%
4335       \fi%
4336     }{%
4337   }%
4338 \ifl@d@Xlinenum%
4339   \wrap@edcrossref{\@this@crossref@start}{%
4340     \linenumrep{#2}%
4341     \iftoggle{Xlineflag@\@currentseries}{#8}{%
4342   }%
4343   \iftoggle{Xnotboxingsubline@\@currentseries}{\egroup}{}%Close startline
box

```

```

4344 \ifl@d@ssub%
4345 \csuse{Xsublinesep@}\@currentseries}%
4346 \wrap@edcrossref{\@this@crossref@start}{\sublinenumrep{#3}}%
4347 \fi%
4348 \fi%
4349 \ifcsstring{Xlinenumannotationposition@}\@currentseries}{after}%
4350 {\@annot@start@print}%
4351 {}%
4352 \iftoggle{Xnotboxingsubline@}\@currentseries}{\egroup}%Close startline
box
4353 %

```

Then print the dash + end line number, or the range symbol.

```

4354 \ifdimequal{\csuse{Xboxendlinenum@}\@currentseries}{0pt}%
4355 {\bgroup}%
4356 {\hbox to \csuse{Xboxendlinenum@}\@currentseries}\bgroup}%
4357 \ifl@d@Xtwolines%
4358 \ifl@d@Xlinenum%
4359 \ifl@d@Xmorethantwolines%
4360 \wrap@edcrossref{\@this@crossref@end}{\csuse{Xmorethantwolines@}\@currentseries}}%
4361 \else%
4362 \wrap@edcrossref{\@this@crossref@end}{\csuse{Xtwolines@}\@currentseries}}%
4363 \fi%
4364 \fi%
4365 \else%
4366 \ifl@d@dash%
4367 \ifdefined\linrangesep%
4368 \linrangesep%
4369 \else%
4370 \csuse{Xlinrangeseparator@}\@currentseries}%
4371 \fi%
4372 \fi%
4373 \ifcsstring{Xlinenumannotationposition@}\@currentseries}{before}%
4374 {\@annot@end@print}%
4375 {}%
4376 \ifl@d@pnum%
4377 \ifbool{expr}%
4378 not test{\ifnumequal{\xpageref{printlines@}\the\printlines@count}}{#4}}%
4379 or not tog1{Xnopagenumberifcurrent@}\@currentseries}%
4380 }{%
4381 \csuse{Xbeforepagenumber@}\@currentseries}%
4382 \wrap@edcrossref{\@this@crossref@end}{%
4383 #4%
4384 }%
4385 \csuse{Xpagelinesep@}\@currentseries}%
4386 }{}%
4387 \fi%
4388 \ifl@d@Xlinenum%
4389 \ifl@d@elin%
4390 \wrap@edcrossref{\@this@crossref@end}{%
4391 \linenumrep{#5}}%

```

```

4392 \iftoggle{Xlineflag@\@currentseries}{#8}{}%
4393 }%
4394 \fi%
4395 \ifl@d@esl%
4396 \ifl@d@elin%
4397 \csuse{Xsublinesep@\@currentseries}%
4398 \fi%
4399 \wrap@edcrossref{\@this@crossref@end}{\sublinenumrep{#6}}%
4400 \fi%
4401 \fi%
4402 \ifcsstring{Xlinenumannotationposition@\@currentseries}{after}%
4403 {\@annot@end@print}%
4404 {}%
4405 \fi%
4406 \ifdimequal{\csuse{Xboxendlinenum@\@currentseries}}{0pt}%
4407 {}%
4408 {\hfill}%Prevent underfull hbox
4409 \egroup%
4410 \endgroup%
4411 }%
4412 %

```

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

```

4413 \newcommand{\prepare@Xgroupbyline}[3]{%
4414 \iftoggle{Xgroupbylineseparetwolines@#1}{%
4415 \l@dparsesfootspec#2%
4416 \ifcsdef{#1@forinserting@\l@dparsedendpage-\l@dparsedendline-\l@dparsedendsub}%
4417 {%
4418 \csgappto{#1@forinserting@\l@dparsedendpage-\l@dparsedendline-\l@dparsedendsub}{%
4419 \hskip\csuse{Xafternote@#1}\relax%
4420 }%
4421 }%
4422 {}%
4423 \add@hooktoggle@specific@to@cs{#1@forinserting@\l@dparsedendpage-\l@dparsedendline-\l@dparsedendsub}%
4424 \add@hookarg@specific@to@cs{#1@forinserting@\l@dparsedendpage-\l@dparsedendline-\l@dparsedendsub}%
4425 \csxappto%
4426 {#1@forinserting@\l@dparsedendpage-\l@dparsedendline-\l@dparsedendsub}%
4427 {%
4428 \keep@this@crossref@forinserting%
4429 \unexpanded{%
4430 \ifcempty{Xsymlinenum@#1}%
4431 {\csuse{Xparafootsep@#1}}%

```

```

4432     {}%
4433     #3{#1}{#2}%
4434   }%
4435 }%
4436 \listcsxadd{#1@forinserting}{\l@dparsedendpage-\l@dparsedendline-\l@dparsedendsub}%
4437 }{%
4438 \ifcsdef{#1@forinserting@all}{%
4439   \csgappto%
4440   {#1@forinserting@all}%
4441   {\hskip\csuse{Xafternote@#1}\relax}%
4442 }{%}
4443 \add@hooktoggle@specific@to@cs{#1@forinserting@all}%
4444 \add@hookarg@specific@to@cs{#1@forinserting@all}%
4445 \csxappto%
4446 {#1@forinserting@all}%
4447 {%
4448   \keep@this@crossref@forinserting%
4449   \unexpanded{%
4450     \ifcsemt{Xsymlinenum@#1}%
4451     {\csuse{Xparafootsep@#1}}%
4452   }%
4453   #3{#1}{#2}%
4454 }%
4455 }%
4456 }%
4457 \listcsgadd{#1@forinserting}{all}%
4458 }%
4459 %

```

`\s@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.

```

4460 \newcommand{\keep@this@crossref@forinserting}{%
4461   \unexpanded{\gdef\@this@crossref@start}{\@this@crossref@start}%
4462   \unexpanded{\gdef\@this@crossref@end}{\@this@crossref@end}%
4463 }%
4464 %

```

XIV Familiar footnotes

XIV.1 Adjacent footnotes

The original EDMAC provided users with five series of critical footnotes (\Afootnote \Bfootnote \Cfootnote \Dfootnote \Efootnote), and L^AT_EX provides a single numbered footnote. The reledmac package uses the EDMAC mechanism to provide six series of numbered footnotes.

First, though, the footmisc package has an option whereby two or more consecutive \footnotes have their marks separated by commas. This seemed to Peter Wilson such a useful ability that it was provided automatically by eledmac.

Maïeul Rouquette has maintained this feature in reledmac, despite he thought that is not directly in relationship with the aim of reledmac.

`\multiplefootnotemarker` These macros may have been defined by the memoir class, are provided by the footmisc package and perhaps by other footnote packages. That is why we use `\providecommand` and not `\newcommand`.

```

4465 \providecommand*\multiplefootnotemarker}{3sp}
4466 \providecommand*\multfootsep}{\textsuperscript{\normalfont,}}
4467
4468 %

```

`\m@mmf@prepare` A pair of self-cancelling kerns. This may have been defined in the memoir class.

```

4469 \providecommand*\m@mmf@prepare}{%
4470 \kern-\multiplefootnotemarker
4471 \kern\multiplefootnotemarker\relax}
4472 %

```

`\m@mmf@check` This may have been defined in the memoir class. If it recognises the last kern as `\multiplefootnotemarker` it typesets `\multfootsep`.

```

4473 \providecommand*\m@mmf@check}{%
4474 \ifdim\lastkern=\multiplefootnotemarker\relax
4475 \edef\@x@sf{\the\spacefactor}%
4476 \unkern
4477 \multfootsep
4478 \spacefactor\@x@sf\relax
4479 \fi}
4480
4481 %

```

We have to modify `\@footnotetext` and `\@footnotemark`. However, if memoir is used the modifications have already been made.

```

4482 \ifclassloaded{memoir}{}{%
4483 %

```

`\@footnotetext` Add `\m@mmf@prepare` at the end of `\@footnotetext`.

```

4484 \apptocmd{\@footnotetext}{\m@mmf@prepare}{}{}
4485 %

```

`\@footnotemark` Modify `\@footnotemark` to cater for adjacent `\footnotes`.

```

4486
4487 \patchcmd{\@footnotemark}
4488 {\nobreak}
4489 {\m@mmf@check
4490 \nobreak
4491 }
4492 {}{}
4493 \patchcmd{\@footnotemark}
4494 {\@makefnmark}
4495 {\@makefnmark
4496 \m@mmf@prepare
4497 }
4498 {}{}
4499 %

```

Finished the modifications for the non-memoir case.

```
4500 }
4501
4502 %
```

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
`\@footnotetext` with its `\@footnotetext`, using different forms for when in numbered or regular text.

```
4503 \pretocmd{\@footnotetext}{%
4504   \ifnumberedpar@
4505     \edtext{}{%
4506       \advance \edtext@notes by 1%
4507       \l@dbfnote{#1}%
4508     }%
4509   \else
4510     {}{}
4511 \apptocmd{\@footnotetext}{\fi}{}{}%
4512 %
```

`\l@dbfnote` `\l@dbfnote` adds the footnote to the insert list, and `\v1@dbfnote` calls the original
`\v1@dbfnote` `\@footnotetext`. We also patch `\footnote` in order to get the correct footnote
`\v1@dbfnote` numbers when typesetting parallel texts. This is moved into a `\get@fnmark` command.

```

\footnote
\get@fnmark
\get@thisfootnote
4513 \patchcmd%
4514 {\footnote}%
4515 {\stepcounter\@mpfn}%
4516 {%
4517   \ifboolexpr{bool{l@dpairing} or bool{l@dprintingpages} or bool{
4518     l@dprintingcolumns}}{%
4519     \global\advance\footnote@reading by \one%
4520     \get@thisfootnote%
4521     \get@fnmark{\thisc@footnote}%
4522     \ifcsdef{footnotereading\the\footnote@reading=typeset}%
4523       {\setcounter{\@mpfn}{\csuse{footnotereading\the\footnote@reading=
typeset}}}%
4524       {\setcounter{\@mpfn}{\footnote@reading}}}%
4525     }{%
4526       \stepcounter\@mpfn%
4527     }%
4528   }%
4529   {}
4530   {}
4531
4532 \newcommand{\get@thisfootnote}{%
4533   \ifboolexpr{bool{l@dpairing} or bool{l@dprintingpages} or bool{
l@dprintingcolumns}}{%
4534     \protected@xdef\thisc@footnote{\the\footnote@reading}%
4535     }{%
```

```

4536 \protected@xdef\thisc@footnote{\the\c@footnote}%
4537 }%
4538 }%
4539
4540 \newcommand{\l@dbfnote}[1]{%
4541 \get@thisfootnote%
4542 \gdef\@tag{#1\relax}%
4543 \ifledRcol%
4544 \xright@appenditem{%
4545 \ifdefined\Hy@footnote@currentHref%
4546 \noexpand\def\noexpand\Hy@footnote@currentHref{\
Hy@footnote@currentHref}%
4547 \fi%
4548 \noexpand\vl@dbfnote{{\expandonce\@tag}}{\thisc@footnote}%
4549 }%
4550 \to\inserts@listR
4551 \global\advance\insert@countR \@ne%
4552 \else%
4553 \xright@appenditem{%
4554 \ifdefined\Hy@footnote@currentHref%
4555 \noexpand\def\noexpand\Hy@footnote@currentHref{\
Hy@footnote@currentHref}%
4556 \fi%
4557 \noexpand\vl@dbfnote{{\expandonce\@tag}}{\thisc@footnote}%
4558 }%
4559 \to\inserts@list
4560 \global\advance\insert@count \@ne%
4561 \fi
4562 \ignorespaces%
4563 }%
4564
4565 \newcommand{\get@fnmark}[1]{%
4566 \ifboolexpr{bool{l@dpairing} or bool{l@dprintingpages} or bool{
l@dprintingcolumns}}%
4567 {%
4568 \stepcounter{footnote@typeset}%
4569 \setcounter{footnote}{\c@footnote@typeset}%
4570 \immediate\write\@mainaux{%
4571 \csgdef{footnotereading#1=typeset}{\the\c@footnote@typeset}%
4572 }%
4573 \def\@thefnmark{\thefootnote}%
4574 }%
4575 {%
4576 \setcounter{footnote}{#1}%
4577 \def\@thefnmark{\thefootnote}%
4578 }%
4579 }%
4580
4581 \newcommand{\vl@dbfnote}[2]{%
4582 \get@fnmark{#2}%
4583 \@footnotetext{#1}%
4584 }%
4585 %

```


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
4586 \newcommand*\prebodyfootmark{%
4587   \leavevmode
4588   \ifhmode
4589     \edef\x@sf{\the\spacefactor}%
4590     \m@mmf@check
4591     \nobreak
4592     \fi}
4593 \newcommand\postbodyfootmark{%
4594   \m@mmf@prepare
4595   \ifhmode\spacefactor\x@sf\fi\relax}
4596
4597 %
```

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.

```
4598 \newcommand\arrangementX[2][1,usedefault]{%
4599   \def\do##1{%
4600     \csname arrangementX@#2\endcsname{##1}%
4601   }%
4602   \ifstrempy{#1}%
4603     {%
4604       \dolistloop{\@series}%
4605     }%
4606     {
4607       \docsvlist{#1}%
4608     }%
4609   }%
4610 %
```

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.

```
4611 \newcommand*\normal@footnotemarkX[1]{%
4612   \prebodyfootmark
4613   \wrapped@bodyfootmarkX{#1}%
4614   \postbodyfootmark}
4615
4616 %
```

`\normalbodyfootmarkX` The `\normalbodyfootmarkX{series}` really typesets the in-text marker. The style is the normal superscript.

```

4617 \newcommand*{\normalbodyfootmarkX}[1]{%
4618   \hbox{\textsuperscript{\normalfont\@nameuse{@thefnmark#1}}}%
4619   %

```

\normalvfootnoteX `\normalvfootnoteX{<series>}{<text>}` does the `\insert` for the `<series>` and calls the series' `\footfmt...` to format the `<text>`.

```

4620 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\normalvfootnoteX}[2]{%
4621   \beforeinsertion@X{#1}%
4622   \insert\@nameuse{footins#1}\bgroup
4623     \fontseries{\seriesdefault}%
4624     \fontshape{\shapedefault}%
4625     \hsize=\expandafter\dimexpr\csuse{widthX@#1}\relax%
4626     \noindent\csuse{bhooknoteX@#1}%
4627     \csuse{notefontsizeX@#1}%
4628     \footplitskips
4629     \ifl@dpairing\ifl@dpaging\else%
4630       \setnoteswidthliketwocolumnsX@{#1}%
4631       \fi\fi%
4632     \setnotesXpositionliketwocolumns@{#1}%
4633     \spaceskip=\z@skip \xspaceskip=\z@skip
4634     \csuse{\csuse{footnote@dir}}\@nameuse{footfmt#1}{#1}{#2}\egroup}
4635   %
4636 %

```

\mpnormalvfootnoteX The minipage version.

```

4637 \newcommand*{\mpnormalvfootnoteX}[3]{%
4638   \get@thisfootnoteX{#1}%
4639   \get@fnmarkX{#1}{\thisc@footnote}{#3}%
4640   \ifstrempy{#3}{%
4641     \edef\this@footnoteX@reading{\the\c@name footnote#1@reading\endcsname}%
4642   }{%
4643     \edef\this@footnoteX@reading{###3}%
4644   }%
4645   \global\setbox\@nameuse{mpfootins#1}\vbox{%
4646     \unvbox\@nameuse{mpfootins#1}
4647     \noindent\csuse{bhooknoteX@#1}%
4648     \csuse{notefontsizeX@#1}%
4649     \hsize\columnwidth
4650     \@parboxrestore
4651     \color@begingroup
4652     \@nameuse{footfmt#1}{#1}{#2}\color@endgroup}}
4653   %
4654 %

```

\normalfootfmtX `\normalfootfmtX{<series>}{<text>}` typesets the footnote text, prepended by the marker.

```

4655 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\normalfootfmtX}[2]{%
4656   \ifluatex%
4657     \textdir\footnote@luatextextdir%
4658     \pardir\footnote@luatexpardir%
4659   \fi%
4660   \protected@edef\@currentlabel{%

```

```

4661 \@nameuse{@thefnmark#1}%
4662 }%
4663 \ledsetnormalparstuffX{#1}%
4664 \sethangindentX{#1}%
4665 \rule\z@\splittopskip%
4666 {{\csuse{notenumfontX@#1}\wrapped@footfootmarkX{#1}}}%
4667 \csuse{wrapcontentX@#1}{#2}%
4668 \strut\par}}
4669
4670 %

```

\normalfootfootmarkX `\normalfootfootmarkX{<series>}` is called by `\normalfootfmtX` to typeset the footnote marker in the footer before the footnote text.

```

4671 \newcommand*{\normalfootfootmarkX}[1]{%
4672 \textsuperscript{\@nameuse{@thefnmark#1}}}%
4673
4674 %

```

\normalfootstartX `\normalfootstartX{<series>}` is the `<series>` footnote starting macro used in the output routine.

```

4675 \newcommand*{\normalfootstartX}[1]{%
4676 \ifdimequal{0pt}{\prenotesX@}{}%
4677 {%
4678 \iftoggle{prenotesX@}{%
4679 \togglefalse{prenotesX@}%
4680 \skip\csname footins#1\endcsname=%
4681 \glueexpr\csuse{prenotesX@}+\csuse{afterruleX@#1}\relax%
4682 }%
4683 }%
4684 }%
4685 \vskip\skip\csname footins#1\endcsname%
4686 \leftskip=\z@
4687 \rightskip=\z@
4688 \ifl@dpairing\else%
4689 \hsize=\old@hsize%
4690 \fi%
4691 \setnoteswidthliketwocolumnsX@{#1}%
4692 \setnotesXpositionliketwocolumns@{#1}%
4693 \print@footnoteXrule{#1}%
4694 }%
4695
4696 %

```

\normalfootnoteruleX The rule drawn before the footnote series group.

```

4697 \let\normalfootnoteruleX=\footnoterule
4698
4699 %

```

\normalfootgroupX `\normalfootgroupX{<series>}` sends the contents of the `<series>` insert box to the output page without alteration.

```

4700 \newcommand*{\normalfootgroupX}[1]{%
4701   \csuse{bhookgroupX@#1}%
4702   \unvbox\@nameuse{footins#1}%
4703   \hsize=\old@hsize%
4704   }%
4705
4706 %

```

`\mpnormalfootgroupX` The minipage version.

```

4707 \newcommand*{\mpnormalfootgroupX}[1]{%
4708   \vskip\skip\@nameuse{mpfootins#1}
4709   \ifl@dpairing\ifparledgroup%
4710     \leavevmode\marks\parledgroup@{begin}%
4711     \marks\parledgroup@series{#1}%
4712     \marks\parledgroup@type{footnoteX}%
4713   \fi\fi\normalcolor
4714   \ifparledgroup%
4715     \ifl@dpairing%
4716     \else%
4717       \setnoteswidthliketwocolumnsX@{#1}%
4718       \setnotesXpositionliketwocolumns@{#1}%
4719       \print@footnoteXrule{#1}%
4720     \fi%
4721   \else%
4722     \setnoteswidthliketwocolumnsX@{#1}%
4723     \setnotesXpositionliketwocolumns@{#1}%
4724     \print@footnoteXrule{#1}%
4725   \fi%
4726   \csuse{bhookgroupX@#1}%
4727   \unvbox\@nameuse{mpfootins#1}}
4728
4729 %

```

`\normalbfnoteX` #1 = footnote series, #2 = footnote content, #3 manual footnote number

```

4730
4731 \newcommand{\normalbfnoteX}[3]{%
4732   \get@thisfootnoteX{#1}%
4733   \ifledRcol%
4734     \ifluatex
4735       \footnotelang@lua[R]%
4736     \fi
4737     \@ifundefined{xpg@main@language}%if polyglossia
4738     {}%
4739     {\footnotelang@poly[R]}%
4740   \xright@appenditem{%
4741     \noexpand\led@set@index@fornote{#1}%
4742     \noexpand\prepare@edindex@fornote{\l@d@nums}%
4743     \unexpanded{\gdef\@currentnotetype{familiar}}%
4744     \unexpanded{\def\this@footnoteX@reading}{\the\c@name footnote#1
@reading\endcsname}%
4745     \noexpand\vbfnoteX{#1}{#2}{\this@footnote}{#3}%
4746     \noexpand\led@reinit@index@fornote%

```

```

4747 \unexpanded{\advance\@edindex@fornote@\m@ne}%
4748 }%
4749 \to\inserts@listR
4750 \global\advance\insert@countR \@ne%
4751 \else%
4752 \ifluatex
4753 \footnotelang@lua%
4754 \fi
4755 \ifundefined{xpg@main@language}%if polyglossia
4756 {}%
4757 {\footnotelang@poly}%
4758 \xright@appenditem{%
4759 \noexpand\led@set@index@fornote{#1}%
4760 \noexpand\prepare@edindex@fornote{\l@d@nums}%
4761 \unexpanded{\gdef\@currentnotetype{familiar}}%
4762 \unexpanded{\def\this@footnoteX@reading}{\the\csname footnote#1
@reading\endcsname}%
4763 \noexpand\vbfnoteX{#1}{#2}{\thisc@footnote}{#3}%
4764 \noexpand\led@reinit@index@fornote%
4765 \unexpanded{\advance\@edindex@fornote@\m@ne}%
4766 }%
4767 \to\inserts@list
4768 \global\advance\insert@count \@ne%
4769 \fi
4770 \ignorespaces}
4771
4772 %

```

\get@thisfootnoteX The macro `\get@thisfootnote` command just saves the footnote number in the `\thisfootnote` macro, depending on the use of pairing environments.

```

4773 \newcommand{\get@thisfootnoteX}[1]{%
4774 \ifboolexpr{bool{l@dpairing} or bool{l@dprintingpages} or bool{
l@dprintingcolumns}}{%
4775 \protected@xdef\thisc@footnote{\the\csname footnote#1@reading\
endcsname}%
4776 }{%
4777 \protected@xdef\thisc@footnote{\the\csname c@footnote#1\endcsname}%
4778 }%
4779 }%
4780 %

```

\vbfnoteX This command calls the correct footnote-inserting commands. #1 footnote series, #2 footnote content, #3 footnote counter, #4 manual footnote mark

```

4781 \newcommand{\vbfnoteX}[4]{%
4782 \get@fnmarkX{#1}{#3}{#4}\relax%
4783 \@nameuse{regvfootnote#1}{#1}{#2}%
4784 }%
4785 %

```

\get@fnmarkX This command gets the correct footnote number when typesetting parallel texts. #1 footnote series, #2 footnote counter, #3 manual footnote number

```

4786 \newcommand{\get@fnmarkX}[3]{%
4787   \ifstrempy{#3}{%
4788     \ifboolexpr{bool{1@dpairing} or bool{1@dprintingpages} or bool{
4789       1@dprintingcolumns}}{%
4790       {%
4791         \stepcounter{footnote#1@typeset}%
4792         \setcounter{footnote#1}{\value{footnote#1@typeset}}%
4793         \@namedef{@thefnmark#1}{\csuse{thefootnote#1}}%
4794         \immediate\write\@mainaux{%
4795           \csgdef{footnote#1reading#2=typeset}{\the\csname c@footnote#1
4796             @typeset\endcsname}%
4797         }%
4798       }%
4799     }%
4800     \setcounter{footnote#1}{#2}%
4801     \@namedef{@thefnmark#1}{\csuse{thefootnote#1}}%
4802   }%
4803   \csdef{@thefnmark#1}{#3}%
4804 }%
4805 }%
4806 %
4807 %

```

\vnumfootnoteX #1 footnote series, #2 footnote content, #3 manual footnote mark

```

4808 \newcommand{\vnumfootnoteX}[3]{%
4809   \ifnumberedpar@
4810     \edtext{}{%
4811       \advance\edtext@notes by 1%
4812       \normalbfnoteX{#1}{#2}{#3}%
4813     }%
4814   \else
4815     \def\this@footnoteX@reading{\the\csname footnote#1@reading\endcsname}%
4816     \get@thisfootnoteX{#1}%
4817     \get@fnmarkX{#1}{\expandonce\thisc@footnote}{#3}%
4818     \@nameuse{regvfootnote#1}{#1}{#2}%
4819   \fi}
4820 %
4821 %

```

arrangementX@normal \arrangementX@normal{<series>} initialises the settings for the <series> footnotes. This should always be called for each series.

```

4822 \newcommand*\arrangementX@normal[1]{%
4823   \csgdef{series@displayX#1}{normal}
4824   \expandafter\let\csname footstart#1\endcsname=\normalfootstartX
4825   \@namedef{@footnotemark#1}{\normal@footnotemarkX{#1}}
4826   \@namedef{bodyfootmark#1}{\normalbodyfootmarkX{#1}}
4827   \expandafter\let\csname regvfootnote#1\endcsname=\normalvfootnoteX
4828   \expandafter\let\csname vfootnote#1\endcsname=\vnumfootnoteX
4829   \expandafter\let\csname footfmt#1\endcsname=\normalfootfmtX
4830   \@namedef{footfootmark#1}{\normalfootfootmarkX{#1}}

```

```

4831 \expandafter\let\csname footgroup#1\endcsname=\normalfootgroupX
4832 \expandafter\let\csname footnoterule#1\endcsname=\normalfootnoteruleX
4833 \count\csname footins#1\endcsname=1000
4834 \csxdef{default@footins#1}{1000}%Use to have note only for one side
4835 \dimen\csname footins#1\endcsname=\csuse{maxhnotesX@#1}
4836 \skip\csname footins#1\endcsname=\csuse{beforenotesX@#1}%
4837 \advance\skip\csname footins#1\endcsname by\csuse{afterruleX@#1}%
4838 %

```

Additions for minipages.

```

4839 \ifnoledgroup@else%
4840 \expandafter\let\csname mpvfootnote#1\endcsname=\mpnormalvfootnoteX
4841 \expandafter\let\csname mpfootgroup#1\endcsname=\mpnormalfootgroupX
4842 \count\csname mpfootins#1\endcsname=1000
4843 \dimen\csname mpfootins#1\endcsname=\csuse{maxhnotesX@#1}
4844 \skip\csname mpfootins#1\endcsname=\csuse{beforenotesX@#1}%
4845 \advance\skip\csname mpfootins#1\endcsname by\csuse{afterruleX@#1}%
4846 \fi
4847 }
4848
4849 %

```

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#50 \newcommand*\arrangementX@twocol}[1]{%
4851 \csgdef{series@displayX#1}{twocol}
4852 \expandafter\let\csname regvfootnote#1\endcsname=\twocolvfootnoteX
4853 \expandafter\let\csname footfmt#1\endcsname=\twocolfootfmtX
4854 \expandafter\let\csname footgroup#1\endcsname=\twocolfootgroupX
4855 \dimen\csname footins#1\endcsname=\csuse{maxhnotesX@#1}%
4856 \skip\csname footins#1\endcsname=\csuse{beforenotesX@#1}%
4857 \advance\skip\csname footins#1\endcsname by \csuse{afterruleX@#1}\relax%
4858 \twocolfootsetupX{#1}
4859 \ifnoledgroup@else%
4860 \expandafter\let\csname mpvfootnote#1\endcsname=\mpnormalvfootnoteX
4861 \expandafter\let\csname mpfootgroup#1\endcsname=\mptwocolfootgroupX
4862 \skip\csname mpfootins#1\endcsname=\csuse{beforenotesX@#1}%
4863 \advance\skip\csname mpfootins#1\endcsname by\csuse{afterruleX@#1}
4864 \mptwocolfootsetupX{#1}
4865 \fi%
4866 }
4867
4868 %

```

```

\twocolfootsetupX \twocolfootsetupX{<series>}
\mptwocolfootsetupX

```

```

4869 \newcommand*\twocolfootsetupX}[1]{%
4870 \count\csname footins#1\endcsname 500
4871 \csxdef{default@footins#1}{500}%Use this to confine the notes to one
side only

```

```

4872 \multiply\dimen\csname footins#1\endcsname by \tw@}
4873 \newcommand*{\mptwocolfootsetupX}[1]{%
4874 \count\csname mpfootins#1\endcsname 500
4875 \multiply\dimen\csname mpfootins#1\endcsname by \tw@}
4876
4877 %

```

`\twocolvfootnoteX` `\twocolvfootnoteX{<series>}`

```

4878 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\twocolvfootnoteX}[2]{%
4879 \beforeinsertion@X{#1}%
4880 \let\bid@RTL@everypar\relax%
4881 \insert\csname footins#1\endcsname\bgroup%
4882 \hsize=\expandafter\dimexpr\csuse{widthX@#1}\relax%
4883 \noindent\csuse{bhooknoteX@#1}%
4884 \csuse{notefontsizeX@#1}%
4885 \footplitskips%
4886 \spaceskip=\z@skip \xspaceskip=\z@skip%
4887 \@nameuse{footfmt#1}{#1}{#2}\egroup}
4888
4889 %

```

`\twocolfootfmtX` `\twocolfootfmtX{<series>}`

```

4890 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\twocolfootfmtX}[2]{%
4891 \protected@edef\@currentlabel{%
4892 \@nameuse{@thefnmark#1}%
4893 }%
4894 \normal@pars%
4895 \sethangindentX{#1}%
4896 \hsize \csuse{hsizetwocolX@#1}%
4897 \setparindentX{#1}%
4898 \tolerance=5000\relax%
4899 \par%
4900 \@tempdima=\parindent%
4901 \csuse{colalignX@#1}%
4902 \parindent=\@tempdima%
4903 {\hspace{\parindent}%
4904 \csuse{notenunfontX@#1}\wrapped@footfootmarkX{#1}\strut%
4905 \csuse{wrapcontentX@#1}{#2}%
4906 \strut\par}%
4907 \allowbreak%
4908 }%
4909
4910 %

```

`\twocolfootgroupX` `\twocolfootgroupX{<series>}`
`\mptwocolfootgroupX`

```

4911 \newcommand*{\twocolfootgroupX}[1]{\csuse{bhookgroupX@#1}\csuse{
notefontsizeX@#1}
4912 \splittopskip=\ht\strutbox
4913 \expandafter
4914 \rigidbalanceX\csname footins#1\endcsname \tw@ \splittopskip}}
4915

```



```

4916 \newcommand*{\mptwocolfootgroupX}[1]{%
4917   \vskip\skip\@nameuse{mpfootins#1}
4918   \ifl@dpairing\ifparledgroup%
4919     \leavevmode\marks\parledgroup@{begin}%
4920     \marks\parledgroup@series{#1}%
4921     \marks\parledgroup@type{footnoteX}%
4922   \fi\fi\normalcolor
4923   \ifparledgroup%
4924     \ifl@dpairing%
4925     \else%
4926       \setnoteswidthliketwocolumnsX@{#1}%
4927       \setnotesXpositionliketwocolumns@{#1}%
4928       \print@footnoteXrule{#1}%
4929     \fi%
4930   \else%
4931     \setnoteswidthliketwocolumnsX@{#1}%
4932     \setnotesXpositionliketwocolumns@{#1}%
4933     \print@footnoteXrule{#1}%
4934   \fi%
4935   \csuse{bhookgroupX@#1}%
4936   \splittopskip=\ht\strutbox
4937   \expandafter
4938   \rigidbalanceX\csname mpfootins#1\endcsname \tw@ \splittopskip}}
4939
4940 %

```

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 4941 \newcommand*{\arrangementX@threecol}[1]{%
4942   \csgdef{series@displayX#1}{threecol}
4943   \expandafter\let\csname regvfootnote#1\endcsname=\threecolvfootnoteX
4944   \expandafter\let\csname footfmt#1\endcsname=\threecolfootfmtX
4945   \expandafter\let\csname footgroup#1\endcsname=\threecolfootgroupX
4946   \dimen\csname footins#1\endcsname=\csuse{maxhnotesX@#1}%
4947   \skip\csname footins#1\endcsname=\csuse{beforenotesX@#1}%
4948   \advance\skip\csname footins#1\endcsname by \csuse{afterruleX@#1}\relax%
4949   \threecolfootsetupX{#1}
4950   \ifnoledgroup@else%
4951     \expandafter\let\csname mpvfootnote#1\endcsname=\mpnormalvfootnoteX
4952     \expandafter\let\csname mpfootgroup#1\endcsname=\mpthreecolfootgroupX
4953     \skip\csname mpfootins#1\endcsname=\csuse{beforenotesX@#1}%
4954     \advance\skip\csname mpfootins#1\endcsname by \csuse{afterruleX@#1}
4955     \mpthreecolfootsetupX{#1}
4956   \fi%
4957 }
4958
4959 %

```

```

\threecolfootsetupX \threecolfootsetupX{<series>}
\mpthreecolfootsetupX

```

```

4960 \newcommand*{\threecolfootsetupX}[1]{%
4961   \count\csname footins#1\endcsname 333
4962   \csxdef{default@footins#1}{333}%Use this to confine the notes to one
side only
4963   \multiply\dimen\csname footins#1\endcsname by \thr@@
4964 \newcommand*{\mpthreecolfootsetupX}[1]{%
4965   \count\csname mpfootins#1\endcsname 333
4966   \multiply\dimen\csname mpfootins#1\endcsname by \thr@@
4967
4968 %

```

`\threecolvfootnoteX` `\threecolvfootnoteX{<series>}{<text>}`

```

4969 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\threecolvfootnoteX}[2]{%
%
4970   \let\bidir@RTL@everypar\relax%
4971   \beforeinsertion@X{#1}%
4972   \insert\csname footins#1\endcsname\bgroup%
4973     \hsize=\expandafter\dimexpr\csuse{widthX@#1}\relax%
4974     \noindent\csuse{bhooknoteX@#1}%
4975     \csuse{notefontsizeX@#1}%
4976     \footplitskips%
4977     \@nameuse{footfmt#1}{#1}{#2}\egroup
4978
4979 %

```

`\threecolfootfmtX` `\threecolfootfmtX{<series>}`

```

4980 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\threecolfootfmtX}[2]{%
4981   \protected@edef\@currentlabel{%
4982     \@nameuse{@thefnmark#1}%
4983   }%
4984   \sethangindentX{#1}%
4985   \normal@pars%
4986   \hsize \csuse{hsizethreecolX@#1}%
4987   \setparindentX{#1}%
4988   \tolerance=5000\relax%
4989   \@tempdima=\parindent%
4990   \csuse{colalignX@#1}%
4991   \parindent=\@tempdima%
4992   {\hspace{\parindent}%
4993     \csuse{notenumfontX@#1}\wrapped@footfootmarkX{#1}\strut%
4994     \csuse{wrapcontentX@#1}{#2}%
4995     \strut\par}\allowbreak}
4996
4997 %

```

`\threecolfootgroupX` `\threecolfootgroupX{<series>}`
`\mpthreecolfootgroupX`

```

4998 \newcommand*{\threecolfootgroupX}[1]{\csuse{bhookgroupX@#1}\csuse{
notefontsizeX@#1}
4999   \splittopskip=\ht\strutbox
5000   \expandafter
5001   \rigidbalanceX\csname footins#1\endcsname \thr@@ \splittopskip}}

```

```

5002
5003 \newcommand*{\mpthreecolfootgroupX}[1]{%
5004   \vskip\skip\@nameuse{mpfootins#1}
5005   \ifl@dpairing\ifparledgroup
5006     \leavevmode\marks\parledgroup@{begin}%
5007     \marks\parledgroup@series{#1}%
5008     \marks\parledgroup@type{footnoteX}%
5009   \fi\fi\normalcolor
5010   \ifparledgroup%
5011     \ifl@dpairing%
5012     \else%
5013       \setnoteswidthliketwocolumnsX@{#1}%
5014       \setnotesXpositionliketwocolumns@{#1}%
5015       \print@footnoteXrule{#1}%
5016     \fi%
5017   \else%
5018     \setnoteswidthliketwocolumnsX@{#1}%
5019     \setnotesXpositionliketwocolumns@{#1}%
5020     \print@footnoteXrule{#1}%
5021   \fi%
5022   \csuse{bhookgroupX@#1}%
5023   \splittopskip=\ht\strutbox
5024   \expandafter
5025   \rigidbalanceX\csname mpfootins#1\endcsname \thr@@ \splittopskip}}
5026
5027 %

```

XIV.4.5 Paraphed footnotes

The following macros set footnotes as one paragraph.

`\arrangementX@paragraph` `\footparagraphX{<series>}`

```

5028 \newcommand*{\arrangementX@paragraph}[1]{%
5029   \csgdef{series@displayX#1}{paragraph}%
5030   \expandafter\let\csname footstart#1\endcsname=\parafootstartX
5031   \expandafter\let\csname regvfootnote#1\endcsname=\para@vfootnoteX
5032   \expandafter\let\csname footfmt#1\endcsname=\parafootfmtX
5033   \expandafter\let\csname footgroup#1\endcsname=\para@footgroupX
5034   \expandafter\let\csname footnoterule#1\endcsname=\normalfootnoteruleX
5035   \count\csname footins#1\endcsname=1000
5036   \csxdef{default@footins#1}{1000}%Use this to confine the notes to one
side only
5037   \dimen\csname footins#1\endcsname=\csuse{maxhnotesX@#1}
5038   \skip\csname footins#1\endcsname=\csuse{beforenotesX@#1}%
5039   \advance\skip\csname footins#1\endcsname by\csuse{afterruleX@#1}%
5040   \para@footsetupX{#1}
5041   \ifnoledgroup\else
5042     \expandafter\let\csname mpvfootnote#1\endcsname=\mppara@vfootnoteX
5043     \expandafter\let\csname mpfootgroup#1\endcsname=\mppara@footgroupX
5044     \count\csname mpfootins#1\endcsname=1000
5045     \dimen\csname mpfootins#1\endcsname=\csuse{maxhnotesX@#1}
5046     \skip\csname mpfootins#1\endcsname=\csuse{beforenotesX@#1}%
5047     \advance\skip\csname mpfootins#1\endcsname by\csuse{afterruleX@#1}%

```

```

5048 \fi
5049 }
5050
5051 %

```

`\para@footsetupX` `\para@footsetupX{<series>}`

```

5052 \newcommand*{\para@footsetupX}[1]{\csuse{bhookgroupX@#1}\csuse{
notefontsizeX@#1}
5053 \setnoteswidthliketwocolumnsX@{#1}%
5054 \ifcempty{widthX@#1}%
5055 {}%
5056 {\columnwidth=\expandafter\dimexpr\csuse{widthX@#1}\relax}%
5057 \dimen0=\baselineskip
5058 \multiply\dimen0 by 1024
5059 \divide\dimen0 by \columnwidth \multiply\dimen0 by \footfudgefiddle\relax
5060 %
5061 \expandafter
5062 \xdef\csname footfudgefactor#1\endcsname{%
\expandafter\strip@pt\dimen0 }}
5063 %
5064 %

```

`\parafootstartX` `\parafootstartX{<series>}`

```

5065 \newcommand*{\parafootstartX}[1]{%
5066 \ifdimequal{0pt}{\prenotesX@}{}%
5067 {%
5068 \iftoggle{prenotesX@}{%
5069 \togglefalse{prenotesX@}%
5070 \skip\csname footins#1\endcsname=%
5071 \glueexpr\csuse{prenotesX@}+\csuse{afterruleX@#1}\relax%
5072 }%
5073 }%
5074 }%
5075 \leftskip=\z@
5076 \rightskip=\z@
5077 \setparindentX{#1}%
5078 \vskip\skip\@nameuse{footins#1}%
5079 \setnoteswidthliketwocolumnsX@{#1}%
5080 \setnotesXpositionliketwocolumns@{#1}%
5081 \print@footnoteXrule{#1}%
5082 }
5083 %
5084 %

```

`\para@vfootnoteX` `\para@vfootnoteX{<series>}{<text>}`
`\mppara@vfootnoteX`

```

5085 \newcommand*{\para@vfootnoteX}[2]{%
5086 \csuse{beforeinsertingX@#1}%
5087 \insert\csname footins#1\endcsname%
5088 \bgroup
5089 \csuse{notefontsizeX@#1}
5090 \footplitskips

```

```

5091 \setbox0=\vbox{\hsize=\maxdimen%
5092 \let\bidir@RTL@everypar\@empty%
5093 \insert@txtbeforenotesX{#1}%
5094 \noindent\csuse{bhooknoteX@#1}%
5095 \@nameuse{footfmt#1}{#1}{#2}}%
5096 \setbox0=\hbox{\unvXH{0}{#1}}%
5097 \dp0=\z@
5098 \ht0=\csname footfudgefactor#1\endcsname\wd0
5099 \box0
5100 \penalty0
5101 \egroup}
5102 \newcommand*\mppara@vfootnoteX}[3]{%
5103 \get@thisfootnoteX{#1}%
5104 \get@fnmarkX{#1}{\thisc@footnote}{#3}%
5105 \edef\thisc@footnoteX@reading{\the\csname footnote#1@reading\endcsname}%
5106 \global\setbox\@nameuse{mpfootins#1}\vbox{%
5107 \unvbox\@nameuse{mpfootins#1}
5108 \csuse{notefontsizeX@#1}
5109 \footsplitskips
5110 \setbox0=\vbox{\hsize=\maxdimen%
5111 \let\bidir@RTL@everypar\@empty%
5112 \noindent\color@begingroup%
5113 \csuse{bhooknoteX@#1}%
5114 \@nameuse{footfmt#1}{#1}{#2}\color@endgroup}%
5115 \setbox0=\hbox{\unvXH{0}{#1}}%
5116 \dp0=\z@
5117 \ht0=\csname footfudgefactor#1\endcsname\wd0
5118 \box0
5119 \penalty0}}
5120
5121 %

```

\unvXH₂₂ \newcommand*\unvXH}[2]{% 2th is optional for retro-compatibility

```

5123 \setbox0=\vbox{\unvbox#1%
5124 \global\setbox1=\lastbox}%
5125 \unhbox1
5126 \unskip % remove \rightskip,
5127 \unskip % remove \parfillskip,
5128 \unpenalty % remove \penalty of 10000,
5129 \hskip\csuse{afternoteX@#2}%
5130 \relax}% but add the glue to go between the notes
5131
5132 %

```

\parafootfmtX \parafootfmtX{<series>}

```

5133 \newcommand*\parafootfmtX}[2]{%
5134 \protected@edef\@currentlabel{%
5135 \@nameuse{@thefnmark#1}%
5136 }%
5137 \insertparafootsepX{#1}%
5138 \ledsetnormalparstuff@common%
5139 {\csuse{notenumfontX@#1}%

```

```

5140 \csuse{notenumfontX@#1}%
5141 \wrapped@footfootmarkX{#1}%
5142 \strut%
5143 \csuse{wrapcontentX@#1}{#2}%
5144 \penalty-10}}
5145
5146 %

```

```

\para@footgroupX \para@footgroupX{<series>}
\mppara@footgroupX
5147 \newcommand*{\para@footgroupX}[1]{%
5148 \hspace=\expandafter\dimexpr\csuse{widthX@#1}\relax%
5149 \unvbox\csname footins#1\endcsname
5150 \ifcsstring{raggedX@#1}{L}{\RaggedLeft}{}%
5151 \ifcsstring{raggedX@#1}{R}{\RaggedRight}{}%
5152 \makehboxofhboxes
5153 \setbox0=\hbox{\unhbox0 \removehboxes}%
5154 \csuse{bhookgroupX@#1}
5155 \csuse{notefontsizeX@#1}
5156 \unhbox0\par}
5157
5158 \newcommand*{\mppara@footgroupX}[1]{%
5159 \setnoteswidthliketwocolumnsX@{#1}%
5160 \vskip\skip\@nameuse{mpfootins#1}
5161 \ifl@dpairing\ifparledgroup
5162 \leavevmode%
5163 \leavevmode\marks\parledgroup@{begin}%
5164 \marks\parledgroup@series{#1}%
5165 \marks\parledgroup@type{footnoteX}%
5166 \fi\fi\normalcolor
5167 \ifparledgroup%
5168 \ifl@dpairing%
5169 \else%
5170 \setnoteswidthliketwocolumnsX@{#1}%
5171 \setnotesXpositionliketwocolumns@{#1}%
5172 \print@footnoteXrule{#1}%
5173 \fi%
5174 \else%
5175 \setnoteswidthliketwocolumnsX@{#1}%
5176 \setnotesXpositionliketwocolumns@{#1}%
5177 \print@footnoteXrule{#1}%
5178 \fi%
5179 \unvbox\csname mpfootins#1\endcsname
5180 \ifcsstring{raggedX@#1}{L}{\RaggedLeft}{}%
5181 \ifcsstring{raggedX@#1}{R}{\RaggedRight}{}%
5182 \makehboxofhboxes
5183 \setbox0=\hbox{\unhbox0 \removehboxes}%
5184 \csuse{bhookgroupX@#1}%
5185 \csuse{notefontsizeX@#1}%
5186 \setparindentX{#1}%
5187 \unhbox0\par}}
5188
5189 %

```

Insertion of the footnotes separator The command `\insertparafootsepX{<series>}` must be called at the beginning of `\parafootftmX`.

```
\insertparafootsepX% \newcommand{\insertparafootsepX}[1]{%
5191   \ifledRcol{%
5192     \ifnumequal{\csuse{prevpage#1@numR}}{\page@numR}%
5193     {\csuse{parafootsepX@#1}}%
5194     {}%
5195     \global\csname prevpage#1@numR\endcsname=\page@numR%
5196   }%
5197   \ifboolexpr{bool{!@dprintingpages} or bool{!@dprintingcolumns} or bool
{numbering}}{%
5198     \ifnumequal{\csuse{prevpage#1@num}}{\page@num}%
5199     {\csuse{parafootsepX@#1}}%
5200     {}%
5201     \global\csname prevpage#1@num\endcsname=\page@num%
5202   }%We can't use directly the page counter, see https://tex.
stackexchange.com/a/239457/7712
5203   \global\expandafter\advance\csname footnote#1@parafootsepX\endcsname
by 1%
5204   \edef\@tmpa{parafootsepX@#1@the\csname footnote#1@parafootsepX\
endcsname}%
5205   \expandafter\edlabel\expandafter{\@tmpa}%
5206   \edef\@tmpb{\expandafter\pageref\expandafter{\@tmpa}}%
5207   \IfStrEq{\csuse{prevpage#1@num@outnumbering}}{\@tmpb}%
5208   {\csuse{parafootsepX@#1}}%
5209   {}%
5210   \global\cslet{prevpage#1@num@outnumbering}{\@tmpb}%
5211 }%
5212 \fi%
5213 }
5214 %
```

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.

```
5215 \newcommand{\wrapped@footfootmarkX}[1]{%
5216   \ifdefined\hypertarget%
5217   \hyperlink%
5218     {@bodyfootmark#1@this@footnoteX@reading}%
5219     {@nameuse{footfootmark#1}}%
5220   \Hy@raisedlink%
5221   \hypertarget%
5222     {@footnotemark#1@this@footnoteX@reading}%
5223     {}%
5224   }%
5225   \else%
5226     \@nameuse{footfootmark#1}%
5227   \fi%
5228 }%
5229 %
```

`\wrapped@bodyfootmarkX` `\wrapped@bodyfootmarkX` prints the footnote mark of the text body, wrapped in `hyperref` package’s commands, if needed.

```

5230 \newcommand{\wrapped@bodyfootmarkX}[1]{%
5231   \ifdefined\hypertarget%
5232     \hyperlink%
5233       {@footnotemark#1@\expandafter\the\csname footnote#1@reading\
endcsname}%
5234       {\@nameuse{bodyfootmark#1}}%
5235     \Hy@raisedlink{%
5236       \hypertarget%
5237         {@bodyfootmark#1@\expandafter\the\csname footnote#1@reading\
endcsname}%
5238         {}}%
5239     }%
5240   \else%
5241     \@nameuse{bodyfootmark#1}%
5242   \fi%
5243 }%
5244 %

```

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 `LuaTeX`.

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@common45 \newcommand*{\ledsetnormalparstuff@common}{%
\Xledsetnormalparstuff46   \ifluatex%
\ledsetnormalparstuffX47   \ifdefstring{\footnote@luatextextdir}{TLT}{}%
5248     {\textdir\footnote@luatextextdir}%
5249     \pardir\footnote@luatexpardir%
5250   \fi%
5251   \csuse{\csuse{footnote@dir}}%
5252   \normal@pars%
5253   \parfillskip \z@ \@plus 1fil}%
5254
5255 \newcommand*{\Xledsetnormalparstuff}[1]{%
5256   \ledsetnormalparstuff@common%
5257   \Xsetparindent{#1}%
5258   \iftoggle{Xparindent@#1}{\hspace{\parindent}}{}%
5259 }%
5260
5261 \newcommand*{\ledsetnormalparstuffX}[1]{%
5262   \ledsetnormalparstuff@common%

```



```

5263 \setparindentX{#1}%
5264 \iftoggle{parindentX@#1}{\hspace{\parindent}}{}%
5265 }%
5266 %

```

XVI Footnotes' width for two columns

We define here some commands which make sense only with `reledpar`, but must be called when defining notes parameters. These commands change the width of block notes to allow them to have the same size than two parallel columns.

`\old@hsize` These two commands are called at the beginning of critical or familiar notes groups. They set, if the option is enabled, the `\hsize`. They are also called at the on the setup for paragraphed notes.

```

5267
5268 \newdimen\old@hsize%
5269 \AtBeginDocument{\old@hsize=\hsize}%
5270
5271 \newcommand{\setXnoteswidthliketwocolumns@}[1]{%
5272   \global\let\hsize@fornote=\hsize%
5273   \global\old@hsize=\hsize%
5274   \let\old@columnwidth=\columnwidth%
5275   \iftoggle{Xnoteswidthliketwocolumns@#1}%
5276     {%
5277       \setwidthliketwocolumns%
5278       \global\let\hsize@fornote=\hsize%
5279     }%
5280   {}%
5281   \let\hsize=\hsize@fornote%
5282   \let\columnwidth=\old@columnwidth%
5283 }%
5284
5285 \newcommand{\setnoteswidthliketwocolumnsX@}[1]{%
5286   \global\let\hsize@fornote=\hsize%
5287   \global\old@hsize=\hsize%
5288   \let\old@columnwidth=\columnwidth%
5289   \iftoggle{noteswidthliketwocolumnsX@#1}%
5290     {%
5291       \setwidthliketwocolumns%
5292       \global\let\hsize@fornote=\hsize%
5293     }%
5294   {}%
5295   \let\hsize=\hsize@fornote%
5296   \let\columnwidth=\old@columnwidth%
5297 }%
5298
5299 %

```

`\positionliketwocolumns@` These two commands set the position of the critical / familiar footnotes, depending on the hooks `Xnoteswidthliketwocolumns` and `noteswidthliketwocolumnsX`. They call commands which are defined only in `reledpar`, because this feature has no sens without `reledpar`.

```

5300 \newcommand{\setXnotespositionliketwocolumns@}[1]{%
5301   \iftoggle{Xnoteswidthliketwocolumns@#1}{%
5302     \csuse{setnotespositionliketwocolumns@\columns@position}%
5303   }{}%
5304 }%
5305
5306 \newcommand{\setnotesXpositionliketwocolumns@}[1]{%
5307   \iftoggle{noteswidthliketwocolumnsX@#1}{%
5308     \csuse{setnotespositionliketwocolumns@\columns@position}%
5309   }{}%
5310 }%
5311
5312 %

```

XVII Footnotes' order

`\fnpos` The `\fnpos` and `\mpfnpos` simply place their arguments in `\@fnpos` and `\@mpfnpos`, which will be used later in the output routine.

```

5313 \def\@fnpos{familiar-critical}
5314 \def\@mpfnpos{critical-familiar}
5315 \newcommand{\fnpos}[1]{\xdef\@fnpos{#1}}
5316 \newcommand{\mpfnpos}[1]{\xdef\@mpfnpos{#1}}
5317 %

```

XVIII Footnotes' rule

Because the footnotes' rules can be shifted to the right when footnotes are set like two columns, we do not print them directly, but we put them in a `\vbox`.

```

\print@Xfootnoterule18 \newcommand{\print@Xfootnoterule}[1]{%
\print@footnoteXrule19   \vskip-\csuse{Xafterrule@#1}%Because count in \dimen\csuse{#1footins}
5320   \nointerlineskip%
5321   \moveleft-\leftskip\vbox{\csuse{#1footnoterule}}%
5322   \nointerlineskip%
5323   \vskip\csuse{Xafterrule@#1}%
5324 }%
5325
5326 \newcommand{\print@footnoteXrule}[1]{%
5327   \vskip-\csuse{afterruleX@#1}%Because count in \dimen\csuse{footins#1}
5328   \nointerlineskip%
5329   \moveleft-\leftskip\vbox{\csuse{footnoterule#1}}%
5330   \nointerlineskip%
5331   \vskip\csuse{afterruleX@#1}%
5332 }%
5333
5334 %

```

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.

```

5335 \newtoggle{Xprenotes@}%
5336 \toggletrue{Xprenotes@}%
5337 \newcommand{\Xprenotes@}{0pt}%
5338 \newcommand*{\Xprenotes}[1]{\renewcommand{\Xprenotes@}{#1}}%
5339 \newcommand{\preXnotes}[1]{\led@warning@preXnotes@deprecated\Xprenotes{#1}}
    %For compatibility
5340 %

```

The same, but for familiar footnotes.

```

\Xprenotes@1 \newtoggle{prenotesX@}
\Xprenotes@2 \toggletrue{prenotesX@}
5343 \newcommand{\prenotesX@}{0pt}
5344 \newcommand*{\prenotesX}[1]{\renewcommand{\prenotesX@}{#1}}
5345 %

```

XIX.3 Internal commands

```

firstXseries@46 \gdef\firstXseries@{}
prepare@Xprenotes47 \newcommand{\prepare@Xprenotes}[1]{%
5348 \ifdimequal{Opt}{\Xprenotes@}%
5349 {}%
5350 {%
5351 \IfStrEq{\firstXseries@}{}{%
5352 \global\skip\csuse{#1footins}=\Xprenotes@%
5353 \global\advance\skip\csname #1footins\endcsname by\csuse{Xafterrule@
#1}%
5354 \gdef\firstXseries@{#1}%
5355 }%
5356 {%
5357 \ifseriesbefore{#1}{\firstXseries@}%
5358 {%
5359 \global\skip\csuse{#1footins}=\csuse{Xbeforenotes@\firstXseries@}%
5360 \global\advance\skip\csname #1footins\endcsname by\csuse{Xafterrule@
#1}%
5361 \gdef\firstXseries@{#1}%
5362 }%
5363 {}%
5364 }%
5365 }%
5366 }
5367 %

```

The same thing is required for familiar notes and \prenotesX.

```

firstseriesX@68 \gdef\firstseriesX@{}
prepare@prenotesX69 \newcommand{\prepare@prenotesX}[1]{%
5370 \ifdimequal{Opt}{\prenotesX@}%
5371 {}%
5372 {%
5373 \IfStrEq{\firstseriesX@}{}{%
5374 \global\skip\csuse{footins#1}=\prenotesX@%
5375 \global\advance\skip\csname footins#1\endcsname by\csuse{afterruleX@
#1}%
5376 \gdef\firstseriesX@{#1}%
5377 }%
5378 {%
5379 \ifseriesbefore{#1}{\firstseriesX@}%
5380 {%
5381 \global\skip\csuse{footins#1}=\csuse{beforenotesX@\firstseriesX@}%
5382 \global\advance\skip\csname footins#1\endcsname by\csuse{afterruleX@
#1}%
5383 \gdef\firstseriesX@{#1}%
5384 }%
5385 {}%
5386 }%
5387 }%
5388 }
5389 %

```

XX Endnotes

First, check the noend option.

```
5390 \ifbool{noend@}{}{%Used instead of \ifnoend@ to prevent expansion problem
5391 %
```

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.

```
5392 \newcommand{\l@dend@open}[1]{%
5393   \global\booltrue{l@dend@#1}%
5394   \expandafter\immediate%
5395   \expandafter\openout%
5396   \csname l@d@#1end\endcsname%
5397   =\l@auxdir\jobname.#1end\relax%
5398 }%
5399 \newcommand{\l@dend@close}[1]{%
5400   \global\boolfalse{l@dend@#1}%
5401   \expandafter\immediate%
5402   \expandafter\closeout\csname l@d@#1end\endcsname%
5403 }%
5404
5405 %
```

`\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.

```
5406 \newcommand{\l@dend@stuff}{%
5407   \def\do##1{%
5408     \ifbool{l@dend@##1}{%
5409       {\l@dend@open{##1}}%
5410       \expandafter\immediate\expandafter\write\csname l@d@##1end\endcsname{\
string\l@d@section{the\section@num}\@percentchar}%
5411     }%
5412     \dolistloop{\@series}%
5413   }%
5414
5415 %
```

`\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.

```

5416 \global\newbool{parapparatus@#4}\def\endprint#1#2#3#4#5#6#7{%
5417 \iftoggle{Xendfirstnote@#4}{%
5418 \global\togglefalse{Xendfirstnote@#4}%
5419 \csuse{Xendtxtbeforenotes@#4}%
5420 }%
5421 \csuse{Xendbhooknote@#4}%
5422 \csuse{Xendnotefontsize@#4}%
5423 \hangindent=\csuse{Xendhangindent@#4}%
5424 \Xendstorelineinfo{#1}{#4}%
5425 \ifXendinsertsep@%
5426 \hskip\csuse{Xendafternote@#4}\relax%
5427 \ifcsdef{prevendline#4}%
5428 {\ifcsequal{prevendline#4}{lineinfo@}%
5429 {\ifcsequal{Xendsymnum@#4}{\csuse{Xendsep@#4}}}%
5430 {\csuse{Xendsep@#4}}}%
5431 }%
5432 {\csuse{Xendsep@#4}}%
5433 \else%
5434 \iftoggle{Xendparagraph@#4}%
5435 {\global\Xendinsertsep@true}%
5436 {}%
5437 \fi%
5438 \xdef\@currentseries{#4}%
5439 \def\do##1{%
5440 \setkeys[mac]{truefootnoteoption}{##1}%
5441 }%
5442 \notblank{#5}{\docsvlist{#5}}}%
5443 \IfStrEq{#6}{R}{\ledRcol@true}{}%
5444 \def\@this@crossref@start{#7:start}%
5445 \def\@this@crossref@end{#7:end}%
5446 \printlineendnote{#1}{#4}%
5447 \IfStrEq{#6}{R}{\ledRcol@false}{}%
5448 \nottoggle{Xendlemmadisablefontselection@#4}%
5449 {\select@lemmafont#1}%
5450 {}%
5451 \bgroup%
5452 \csuse{Xendlemmafont@#4}%
5453 \csuse{Xendwraplemma@#4}{#2}%
5454 \egroup%
5455 \ifboolexpr{%
5456 togl {nosep@}%
5457 or test{\ifcsequal{Xendlemmaseparator@#4}}}%
5458 }%

```

```

5459 {\hskip\csuse{Xendinplaceoflemmaseparator@#4}\relax}%
5460 {\nobreak%
5461 \hskip\csuse{Xendbeforelemmaseparator@#4}%
5462 \csuse{Xendlemmaseparator@#4}%
5463 \hskip\csuse{Xendafterlemmaseparator@#4}%
5464 \relax%
5465 }%
5466 \csuse{Xendwrapcontent@#4}{#3}%
5467 \nottoggle{Xendparagraph@#4}{\par}{}%
5468 \def\do##1{%
5469 \setkeys[mac]{falsefootnoteoption}{##1}%
5470 }%
5471 \notblank{#5}{\docsvlist{#5}}{}%
5472 }}%
5473
5474 \let\l@d@section=\@gobble
5475
5476 %

```

\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.

```

5477 \newcommand{\Xendstorelineinfo}[2]{%
5478 \l@dp@rsefootspec#1|}%
5479 \iftoggle{Xendnumberonlyfirstintwoline@#2}{%
5480 \xdef\lineinfo@{\l@dparsedstartpage - \l@dparsedstartline - \
5481 \l@dparsedstartsub - \l@dparsedendpage - \l@dparsedendline - \
5482 \l@dparsedendsub}%
5483 }%
5484 {%
5485 \xdef\lineinfo@{\l@dparsedstartpage - \l@dparsedstartline - \
5486 \l@dparsedstartsub}%
5487 }%
5488 }%
5489 %

```

\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.

```

5487 \newcommand{\printlineendnote}[2]{%
5488 \ifboolexpr{%
5489 togl {nonum@}%
5490 or togl {Xendnonumber@#2}%
5491 }%
5492 {\hspace{\csuse{Xendinplaceofnumber@#2}}}%
5493 {%
5494 \iftoggle{Xendnumberonlyfirstinline@#2}%
5495 {\ifcsdef{prevendline#2}%
5496 {\ifcsequal{prevendline#2}{lineinfo@}%
5497 {%
5498 \csuse{Xendbhookinplaceofnumber@#2}%

```

```

5499         \ifcempty{Xendsymmlinenumber@#2}%
5500         {\hspace{\csuse{Xendinplaceofnumber@#2}}}%
5501         {\printsymmlineendnotearea{#2}}%
5502         \csuse{Xendahookinplaceofnumber@#2}%
5503         }%
5504         {\printlineendnotearea{#1}{#2}}}%
5505         {\printlineendnotearea{#1}{#2}}}%
5506     }%
5507     {\printlineendnotearea{#1}{#2}}%We keep every time line
5508     \csxdef{prevendline#2}{\lineinfo@}%
5509     }%
5510 }%
5511 %

```

```

\printsymmlineendnotearea 12 \newcommand{\printsymmlineendnotearea}[1]{%
5513     \hspace{\csuse{Xendbeforemsymmlinenumber@#1}}%
5514     \csuse{Xendnotenumfont@#1}%
5515     \ifdimequal{\csuse{Xendboxmsymmlinenumber@#1}}{\z@}%
5516     {\csuse{Xendsymmlinenumber@#1}}%
5517     {\hbox to \csuse{Xendboxmsymmlinenumber@#1}%
5518       {\csuse{Xendsymmlinenumber@#1}\hfill}}%
5519     }%
5520     \hspace{\csuse{Xendaftersymmlinenumber@#1}}%
5521 }%
5522 %

```

\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.)

```

5523 \newcommand{\printlineendnotearea}[2]{%
5524     \csuse{Xendbhooklinenumber@#2}%
5525     \hspace{\csuse{Xendbeforenumber@#2}}%
5526     \bgroup%
5527     \csuse{Xendnotenumfont@#2}%
5528     \ifdimequal{\csuse{Xendboxlinenumber@#2}}{\Opt}%
5529     {\printendlines#1||\ifledRcol@{\Rlineflag}\fi}%
5530     {\leavevmode%
5531       \hbox to \csuse{Xendboxlinenumber@#2}%
5532       {%
5533         \IfSubStr{RC}{\csuse{Xendboxlinenumberalign@#2}}{\hfill}{}%
5534         \printendlines#1||\ifledRcol@{\Rlineflag}\fi%
5535         \IfSubStr{LC}{\csuse{Xendboxlinenumberalign@#2}}{\hfill}{}%
5536       }%
5537     \egroup%
5538     \hspace{\csuse{Xendafternumber@#2}}%
5539     \csuse{Xendahooklinenumber@#2}%
5540 }%
5541 %

```


XX.2 User level commands

XX.2.1 Inserting contents to endnotes

The `\Xendnotes` commands are defined above, when defining apparatus commands by series. Here, we define only `\toendnotes` command not specific to a series, in order to insert arbitrary code. The regular version writes an unexpanded argument, while the regular version writes a once-expanded argument.

```

\toendnotes* \newcommandx{\toendnotes}[2][1,usedefault]{%
\toendnotes* \ifboolexpr{bool{numbering} or bool{numberingR}}{%
5544 \def\do##1{%
5545 \expandafter\immediate\expandafter\write\csname l@d###1end\endcsname%
5546 {\unexpanded{#2}\@percentchar}%
5547 }%
5548 \ifstrempy{#1}%
5549 {\dolistloop{\@series}}%
5550 {\docsvlist{#1}}%
5551 }{\led@err@toendnotes@outsidenumbering}%
5552 }%
5553 \WithSuffix\newcommandx{\toendnotes*}[2][1,usedefault]{%
5554 \ifboolexpr{bool{numbering} or bool{numberingR}}{%
5555 \def\do##1{%
5556 \expandafter\immediate\expandafter\write\csname l@d###1end\endcsname%
5557 {#2\@percentchar}%
5558 }%
5559 \ifstrempy{#1}%
5560 {\dolistloop{\@series}}%
5561 {\docsvlist{#1}}%
5562 }{\led@err@toendnotes@outsidenumbering}%
5563 }%
5564 %

```

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.

```

5565 \newif\ifXendinsertsep@%
5566 \newcommand*{\doendnotes}[1]{%
5567 \l@dend@close{#1}%
5568 \begingroup
5569 \csxdef{prevpagenum@#1}{}%
5570 \csxdef{prevpagerange@#1}{}%
5571 \global\toggletrue{Xendfirstnote@#1}%
5572 \makeatletter
5573 \expandafter\let\csname #1end\endcsname=\endprint
5574 \input\l@auxdir\jobname.#1end%
5575 \global\Xendinsertsep@false%
5576 \endgroup}
5577 %

```

`\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.

```

5578 \newcommand*{\doendnotesbysection}[1]{%
5579   \l@dend@close{#1}%
5580   \csxdef{prevpagenum@#1}{}%
5581   \csxdef{prevpagerange@#1}{}%
5582   \global\expandafter\advance\csname #1end@bysection\endcsname by 1%
5583   \begingroup%
5584     \makeatletter%
5585     \def\l@d@section##1{%
5586       \global\toggletrue{Xendfirstnote@#1}%
5587       \ifnumequal{##1}{\csname #1end@bysection\endcsname}%
5588         {\cslet{#1end}{\endprint}}%
5589         {\cslet{#1end}{\@gobbleseven}}%
5590     }%
5591     \input\l@auxdir\jobname.#1end%
5592     \global\Xendinsertsep@false%
5593   \endgroup%
5594 }%
5595 %

```

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.

```

5596 }%
5597 %

```

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.

```

\setprintendlines5598 \newif\ifl@d@Xendlinenumber%
\ifl@d@Xendlinenumber5599 \newcommand*{\setprintendlines}[6]{%
5600   \let\@annot@start@print\relax%
5601   \let\@annot@end@print\relax%
5602   \l@d@pnumfalse%
5603   \l@d@dashfalse%
5604   \l@d@elinfalse%
5605   %

```

Should we print the line number? It depends on hooks and annotations.

```

5606   \l@d@Xendlinenumbertrue%
5607   \ifboolexpr{togl{Xendnolinenumber@\@currentseries}%
5608     or (%
5609     togl{Xendnolinenumberifannotation@\@currentseries}%

```

```

5610     and not test{\ifdefvoid{\annot@start}}%
5611     and not test{\ifdefvoid{\annot@end}}%
5612     )%
5613 }{%
5614   \l@d@Xendlinenumberfalse%
5615 }{%
5616 %

```

We print the second page number only if the ending page number is different

```

5617   \ifnum#4=#1 \else
5618     \l@d@pnumtrue
5619     \l@d@dashtrue
5620   \fi
5621 %

```

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.

```

5622   \ifboolexpr{%
5623     bool{\l@d@pnum}%
5624     or not test{\ifnumequal{#2}{#5}}%
5625     and bool{\l@d@Xendlinenumber}%
5626   }{%
5627     \l@d@elintrue%
5628     \l@d@dashtrue%
5629     \l@d@elintrue
5630     \l@d@dashtrue
5631     \unless\ifx\relax\annot@end%
5632       \def\@annot@end@print{%
5633         \l@wrapcs@ifnotemptybox{Xendwraplinenumannotation@\@currentseries
5634       }{\annot@end}%
5635     }%
5636     \fi%
5637   }%
5638 %

```

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.

```

5639   \ifl@d@elin%
5640     \def\@annot@start@print{%
5641       \l@wrapcs@ifnotemptybox{Xendwraplinenumannotation@\@currentseries}{\
5642     }{\annot@start}%
5643   }%
5644   \else%
5645     \ifx\annot@start\annot@end%
5646       \unless\ifx\@annot@start\relax%

```

```

5646     \def\@annot@start@print{%
5647         \l@wrapcs@ifnotemptybox{Xendwraplinenumannotation@\@currentseries
5648     }{\@annot@start}%
5649     }%
5650     \fi%
5651     \else%
5652     \ifx\@annot@end@print\relax%
5653     \def\@annot@start@print{%
5654         \l@wrapcs@ifnotemptybox{Xendwraplinenumannotation@\@currentseries
5655     }{%
5656         \ifx\annot@start\empty\else%
5657         \annot@start%
5658         \ifdefined\linangesep%
5659         \linangesep%
5660         \else%
5661         \csuse{Xendlinangeseparator@\@currentseries}%
5662         \fi%
5663         \fi%
5664         \annot@end%
5665         }%
5666         }%
5667         \else%
5668         \let\@annot@start@print\@annot@end@print%
5669         \let\@annot@end@print\relax%
5670         \fi%
5671         \fi%
5672         \fi%
5673     }%

```

We print the starting sub-line if it is nonzero.

```

5672     \l@d@ssubfalse
5673     \ifnum#3=0 \else
5674         \l@d@ssubtrue
5675     \fi
5676 %

```

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.

```

5677     \l@d@eslfalse
5678     \ifnum#6=0 \else
5679         \ifnum#6=#3
5680             \ifl@d@elin \l@d@esltrue \else \l@d@eslfalse \fi
5681         \else
5682             \l@d@esltrue
5683             \l@d@dashtrue
5684         \fi
5685     \fi%
5686 %

```

```

5687     \ifl@d@dash%
5688     \ifboolexpr{togl{fulllines@} or test{\ifcsempy{Xendtwolines@
5689     @currentseries}}}%
5690     {%

```

```

5691 \setistwofollowinglines{#1}{#2}{#4}{#5}%
5692 \ifboolexpr{%
5693   (%
5694     togl {Xendtwolinesbutnotmore@\@currentseries}%
5695     and not%
5696     (%
5697       bool {istwofollowinglines@}%
5698     )%
5699   )%
5700   or%
5701   (%
5702     (not test{\ifnumequal{#1}{#4}})%
5703     and togl{Xendtwolinesonlyinsamepage@\@currentseries}%
5704   )%
5705   }%
5706   {}%
5707   {%
5708     \l@d@dashfalse%
5709     \l@d@Xtwolinestrue%
5710     \l@d@elinfalse%
5711     \l@d@eslfalse%
5712     \ifcempty{Xendmoreethantwolines@\@currentseries}%
5713     {%
5714       {\ifistwofollowinglines@\else%
5715         \l@d@Xmoreethantwolinestrue%
5716       }%
5717     }%
5718   }%
5719 }%
5720 \fi%
5721 %

```

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

```

5722 \iftoggle{Xendnoidenticallinenumannotation@\@currentseries}{%
5723   \ifx\annot@start\annot@end%
5724     \let\annot@end@print\relax%
5725     \ifboolexpr{%Do not print the dash if we do not print the line number
5726       test{\ifdefempty{\linenumrep}}%
5727       or not bool {\l@d@Xendlinenum}%
5728   }{%
5729     \l@d@dashfalse%
5730   }{}%
5731   \fi%
5732 }{}%
5733 %

```

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.

```

5734 \iftoggle{Xendlinenumannotationonlyfirst@\@currentseries}{%
5735   \ifboolexpr{%
5736     (%

```

```

5737     togl{Xendlinenumannotationonlyfirstintwo@\@currentseries}%
5738     and test{\ifcsequal{annot@start}{prevannot@start@\@currentseries}}%
5739     and test{\ifcsequal{annot@end}{prevannot@end@\@currentseries}}%
5740   )%
5741   or%
5742   (%
5743     not togl{Xendlinenumannotationonlyfirstintwo@\@currentseries}%
5744     test{\ifcsequal{annot@start}{prevannot@start@\@currentseries}}%
5745   )%
5746   }{%
5747     \def\@annot@start@print{%
5748       \l@wrapcs@ifnotemptybox{Xendwraplinenumannotation@\@currentseries
5749     }{\csuse{Xendsymlinenumannotation@\@currentseries}}%
5750     }%
5751     \let\@annot@end@print\relax%
5752     \ifboolexpr{%Do not print the dash if do not print the line number
5753       test{\ifdefempty{\linenumrep}}%
5754       or not bool {l@d@Xendlinenum}}%
5755     }{%
5756       \l@d@dashfalse%
5757     }{%
5758     }%
5759     \global\cslet{prevannot@start@\@currentseries}{\annot@start}%
5760     \global\cslet{prevannot@end@\@currentseries}{\annot@end}%
5761   }%
5762   }%
5763   %
5764   %

```

End of \setprintendlines.

\printendlines Now we are ready to print it all.

```

5765 \def\printendlines#1|#2|#3|#4|#5|#6|#7|#8|{%
5766   \begingroup
5767   \setprintendlines{#1}{#2}{#3}{#4}{#5}{#6}%
5768   %

```

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.

```

5769   \ifdimequal{\csuse{Xendboxstartlinenum@\@currentseries}}{0pt}%
5770     {\bgroup}%
5771     {\leavevmode\hbox to \csuse{Xendboxstartlinenum@\@currentseries}\bgroup
5772     \hfill}%
5773   %

```

Then, print the starting page number-

```

5773   \ifboolexpr{%
5774     (%

```

```

5775     test{\ifcsstring{prevpagenum@ \@currentseries}{#1}}%
5776     and not%
5777     (togl{Xendpagenumberonlyfirstifsingle@ \@currentseries} and bool{
l@d@pnum})%
5778   )%
5779   or%
5780   (%
5781   test {\ifcsstring{prevpagerange@ \@currentseries}{#1-#4}}%
5782   )%
5783 }%
5784 {%
5785 \ifcsempy{Xendsympagenum@ \@currentseries}%
5786   {\hspace{\csuse{Xendinplaceofpagenumber@ \@currentseries}}}%
5787   {\csuse{Xendsympagenum@ \@currentseries}}%
5788 }%
5789 {%
5790 \wrap@edcrossref{\@this@crossref@start}{\printnpnum{#1}}%
5791 }%
5792 %

```

Then, determine what must be printed before the start line.

```

5793 \ifl@d@dash%
5794 \ifl@d@pnum%
5795   \csuse{Xendlineprefixsingle@ \@currentseries}%
5796 \else%
5797   \ifcsempy{Xendlineprefixmore@ \@currentseries}%
5798     {\csuse{Xendlineprefixsingle@ \@currentseries}}%
5799     {\csuse{Xendlineprefixmore@ \@currentseries}}%
5800 \fi%
5801 \else%
5802   \csuse{Xendlineprefixsingle@ \@currentseries}%
5803 \fi%
5804 %

```

Then print the starting line, followed, if needed, by the side flag and the starting sub line number, then the line number annotation.

```

5805 \ifcsstring{Xendlinenumannotationposition@ \@currentseries}{before}%
5806   {\@annot@start@print}%
5807   {}%
5808 \ifl@d@Xendlinenumber%
5809   \wrap@edcrossref{\@this@crossref@start}{%
5810     \ifledRcol%
5811       \linenumrepR{#2}%
5812     \else%
5813       \linenumrep{#2}%
5814     \fi%
5815   }%
5816 \fi%
5817 \iftoggle{Xendlineflag@ \@currentseries}{\ifledRcol@ \@Rlineflag\fi}{%
5818 \iftoggle{Xendnotboxingsubline@ \@currentseries}{\egroup}{}%Close the box
5819 \ifl@d@Xendlinenumber%
5820   \ifl@d@ssub%
5821     \csuse{Xendsublinesep@ \@currentseries}%
5822     \wrap@edcrossref{\@this@crossref@start}{%

```

```

5823     \ifledRcol@%
5824     \sublinenumrepR{#3}%
5825     \else%
5826     \sublinenumrep{#3}%
5827     \fi%
5828   }%
5829   \fi%
5830 \fi%
5831 \ifcsstring{Xendlinenumannotationposition@ \@currentseries}{after}%
5832   {\cannot@start@print}%
5833   }%
5834 %

```

Close the box.

```

5835   \iftoggle{Xendnotboxingsubline@ \@currentseries}{\egroup}%
5836 %

```

Open the box for the ending line number.

```

5837   \ifdimequal{\csuse{Xendboxendlinenum@ \@currentseries}}{0pt}%
5838   {\bgroup}%
5839   {\hbox to \csuse{Xendboxendlinenum@ \@currentseries}\bgroup}%
5840 %

```

Print the dash + the ending line number, or the line number range symbol.

```

5841   \ifl@d@Xtwolines%
5842   \ifl@d@Xendlinenum%
5843   \ifl@d@Xmorethantwolines%
5844   \wrap@edcrossref{\@this@crossref@end}{\csuse{Xendmorethantwolines@
5845   @currentseries}}%
5846   \else%
5847   \wrap@edcrossref{\@this@crossref@end}{\csuse{Xendtwolines@
5848   @currentseries}}%
5849   \fi%
5850   \fi%
5851   \else%
5852   \ifl@d@dash%
5853   \ifdefined\linerangesep%
5854   \linerangesep%
5855   \else%
5856   \csuse{Xendlinerangeseparator@ \@currentseries}%
5857   \fi%
5858   \fi%
5859 %

```

Print the ending page number.

```

5858   \ifl@d@pnum%
5859   \ifcsstring{prevpagerange@ \@currentseries}{#1-#4}%
5860   {%
5861   \ifcsempty{Xendsympagenum@ \@currentseries}%
5862   {\hspace{\csuse{Xendinplaceofpagenumber@ \@currentseries}}}%
5863   {\csuse{Xendsympagenum@ \@currentseries}}%
5864   }%
5865   {%
5866   \wrap@edcrossref{\@this@crossref@end}{\printpnum{#4}}%

```



```

5867 }%
5868 \fi%
5869 %

```

Print the ending line number, with, if needed, the line prefix, and followed by the side flag, the subline number, and line number annotation.

```

5870 \ifcsstring{Xendlinenumannotationposition@\@currentseries}{before}%
5871 {\@annot@end@print}%
5872 }%
5873 \ifl@d@Xendlinenumber%
5874 \ifl@d@elin%
5875 \ifl@d@pnum\csuse{Xendlineprefixsingle@\@currentseries}\fi%
5876 \wrap@edcrossref{\@this@crossref@end}{%
5877 \ifledRcol@%
5878 \linenumrepR{#5}%
5879 \else%
5880 \linenumrep{#5}%
5881 \fi%
5882 }%
5883 \iftoggle{Xendlineflag@\@currentseries}{\ifledRcol@\@Rlineflag\fi
}{}%
5884 \fi%
5885 \ifl@d@esl%
5886 \ifl@d@elin%
5887 \csuse{Xendsublinesep@\@currentseries}%
5888 \fi%
5889 \wrap@edcrossref{\@this@crossref@end}{%
5890 \ifledRcol@%
5891 \sublinenumrepR{#6}%
5892 \else%
5893 \sublinenumrep{#6}%
5894 \fi%
5895 }%
5896 \fi%
5897 \fi%
5898 \ifcsstring{Xendlinenumannotationposition@\@currentseries}{after}%
5899 {\@annot@end@print}%
5900 }%
5901 \fi%
5902 %

```

Close the ending line box.

```

5903 \ifdimequal{\csuse{Xendboxendlinenum@\@currentseries}}{0pt}%
5904 {}%
5905 {\hfill}%Prevent underfull hbox
5906 \egroup%
5907 %

```

And, finally, save, if needed, the current page number for the Xendpagenumberonlyfirst hooks.

```

5908 \iftoggle{Xendpagenumberonlyfirst@\@currentseries}%
5909 {\iftoggle{Xendpagenumberonlyfirstintwo@\@currentseries}%
5910 {\csxdef{prevpagerange@\@currentseries}{#1-#4}}%
5911 {\csxdef{prevpagenum@\@currentseries}{#4}}%

```

```

5912 }%
5913 {}%
5914 %

```

Now, the end of `\printendlines` macro.

```

5915 \endgroup%
5916 }%
5917 %
5918 %

```

`\printnpnum` A macro to print a page number in an endnote. Should not be override anymore

```

5919 \newcommand*{\printnpnum}[1]{\csuse{Xendbeforepagenumber@\@currentseries}
5920 }#1\csuse{Xendafterpagenumber@\@currentseries}}
5921 %

```

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@`.

```

5922 \newcommand{\newseries}[1]{%
5923   \def\do##1{\newseries@{##1}}%
5924   \docsvlist{#1}
5925 }
5926 %

```

`\@series` The `\series@` macro is an etoolbox list, which contains the name of all series.

```

5927 \newcommand{\@series}{}
5928 %

```

The command `\newseries@\series` creates a new series of the footnote.

`\newseries@` `\newcommand{\newseries@}[1]{`
`%`

XXI.1 Test if series is still existing

```

5931 \xifinlist{#1}{\@series}{\led@warn@SeriesStillExist{#1}}%
5932 {%
5933 %

```

XXI.2 Init specific to reledpar

When calling `\newseries@` after having loaded `reledpar`, we need to load specific setting.

```

5934 \ifdefined\newseries@par%
5935   \newseries@par{#1}%
5936 \fi%
5937 %

```

XXI.3 For critical footnotes

Critical footnotes are those which start with letters. We look for the \nocritical option of reledmac.

```
5938 \unless\ifnocritical@
5939 %
```

XXI.3.1 Options

```
5940 \newtoggle{Xlineflag@#1}
5941 \newtoggle{Xparindent@#1}
5942 \newtoggle{Xlemmadisablefontselection@#1}
5943 \csgdef{Xwrapcontent@#1}{}%
5944 \csgdef{Xbeforeinserting@#1}{}%
5945 \csgdef{Xhangindent@#1}{Opt}%
5946 \csgdef{Xragged@#1}{}%
5947 \csgdef{Xsizetwocol@#1}{0.45 \hsize}%
5948 \csgdef{Xsizethreecol@#1}{.3 \hsize}%
5949 \csgdef{Xcolalign@#1}{\raggedright}%
5950 \csgdef{Xnotenumfont@#1}{\normalfont}%
5951 \csgdef{Xnotefontsize@#1}{\footnotesize}%
5952 \csgdef{Xbhooknote@#1}{}%
5953 \csgdef{Xbhookgroup@#1}{}%
5954
5955 \csgdef{Xboxlinenum@#1}{Opt}%
5956 \csgdef{Xboxlinenumalign@#1}{L}%
5957
5958 \csgdef{Xboxstartlinenum@#1}{Opt}%
5959 \newtoggle{Xnotboxingsubline@#1}%
5960 \csgdef{Xboxendlinenum@#1}{Opt}%
5961
5962 \csgdef{Xboxsymlinenum@#1}{Opt}%
5963 \newtoggle{Xgroupbyline@#1}%
5964 \newtoggle{Xgroupbylineseparetwolines@#1}%
5965
5966 \newtoggle{Xnumberonlyfirstinline@#1}%
5967 \newtoggle{Xnumberonlyfirstintwolines@#1}%
5968
5969 \newtoggle{Xlinenumannotationonlyfirst@#1}%
5970 \newtoggle{Xlinenumannotationonlyfirstintwo@#1}%
5971
5972 \csgdef{Xtwolines@#1}{}%
5973 \csgdef{Xmorethantwolines@#1}{}%
5974 \csgdef{Xsublinesep@#1}{\fullstop}%
5975 \csgdef{Xpagelinesep@#1}{\csname Xsublinesep@#1\endcsname}%for
backward compatibility, call Xsublinesep@#1
5976 \newtoggle{Xtwolinesbutnotmore@#1}%
5977 \newtoggle{Xtwolinesonlyinsamepage@#1}%
5978 \newtoggle{Xonlypstart@#1}%
5979 \newtoggle{Xpstarteverytime@#1}%
5980
5981 \newtoggle{Xpstart@#1}%
5982 \newtoggle{Xpstartonlyfirst@#1}%
```

```

5983 \csdef{Xprevstart@#1}{0}%Not a hook, but a tool for the hook
Xpstartonlyfirst
5984 \csgdef{Xpstartseparator@#1}{}%
5985
5986 \newtoggle{Xstanza@#1}%
5987 \newtoggle{Xstanzaonlyfirst@#1}%
5988 \csdef{Xprevstanza@#1}{0}%Not a hook, but a tool for the hook
Xstanzaonlyfirst
5989
5990 \csgdef{Xstanzaseparator@#1}{}%
5991
5992 \csgdef{Xsymlinenum@#1}{}%
5993 \csgdef{Xsymlinenumannotation@#1}{}%
5994
5995 \newtoggle{Xnonote@#1}%
5996
5997 \newtoggle{Xnonumber@#1}%
5998 \newtoggle{Xnolinenumber@#1}%
5999 \newtoggle{Xnolinenumberifannotation@#1}%
6000 \csgdef{Xbeforenumber@#1}{Opt}%
6001 \csgdef{Xtxtbeforenumber@#1}{}%
6002 \newtoggle{Xnopagenumberifcurrent@#1}%
6003 \csgdef{Xbeforepagenumber@#1}{}%
6004 \csgdef{Xafternumber@#1}{0.5em}%
6005 \newtoggle{Xnonbreakableafternumber@#1}%
6006 \csgdef{Xbeforesymlinenum@#1}{\csuse{Xbeforenumber@#1}}%
6007 \csgdef{Xaftersymlinenum@#1}{\csuse{Xafternumber@#1}}%
6008 \csgdef{Xinplaceofnumber@#1}{1em}%
6009 \global\cslet{Xlemmaseparator@#1}{\rbracket}%
6010 \csgdef{Xbeforelemmaseparator@#1}{0em}%
6011 \csgdef{Xafterlemmaseparator@#1}{0.5em}%
6012 \csgdef{Xinplaceoflemmaseparator@#1}{1em}%
6013 \csgdef{Xbeforenotes@#1}{1.2em \@plus .6em \@minus .6em}%
6014 \csgdef{Xafterrule@#1}{Opt}%
6015
6016 \csgdef{Xtxtbeforenotes@#1}{%
6017 \newtoggle{Xtxtbeforenotes@#1@typeset}}%Not directly used by user,
but internal
6018 \newtoggle{Xtxtbeforenotesonlyonce@#1}%
6019
6020 \csgdef{Xmaxhnotes@#1}{0.8\vsizer}
6021 \newtoggle{Xnoteswidthliketwocolumns@#1}%
6022 \csgdef{Xparafootsep@#1}{}%
6023 \csgdef{Xafternote@#1}{1em plus.4em minus.4em}%
6024 \csgdef{Xlinerrangeseparator@#1}{\endashchar}%
6025
6026 \csgdef{Xlemmafont@#1}{}%
6027 \csgdef{Xwraplemma@#1}{%
6028 \csgdef{Xwidth@#1}{\hsizer}%
6029 \csgdef{Xwraplinenumannotation@#1}{\textsuperscript}%
6030 \csgdef{Xlinenumannotationposition@#1}{after}%
6031 \expandafter\csgdef{Xinnotemark@#1}##1{\ledinnotemark{##1}}%
6032 \newtoggle{Xnoidenticallinenumannotation@#1}%
6033 %

```

XXI.3.2 Create inserts, needed to add notes in foot

As regards inserts, see chapter 15 of *The TeXbook* by D. Knuth.

```

6034 \expandafter\newinsert\csname #1footins\endcsname%
6035 \unless\ifnoledgroup@%
6036 \expandafter\newinsert\csname mp#1footins\endcsname%
6037 \fi%
6038 %

```

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.

```

6039 \global\newbool{parapparatus@}{\expandafter\newcommand\expandafter
*}{\expandafter\newcommand}\csname #1footnote\endcsname[2][]{%
6040 \xdef\@currentseries{#1}%
6041 \nottoggle{Xnonote@#1}{%
6042 \advance \edtext@notes by 1%
6043 \if@edtext@secondarg@%
6044 \ifledRcol%
6045 \ifcsstring{Xonlyside@#1}{L}{\
led@error@note@called@onrightside{#1footnote}}}%
6046 \else%
6047 \ifcsstring{Xonlyside@#1}{R}{\
led@error@note@called@onleftside{#1footnote}}}%
6048 \fi%
6049 \begingroup%
6050 \newcommand{\content}{##2}%
6051 \ifcsstring{series@display#1}{paragraph}{%
6052 \patchcmd{\@tag}{\par}{\relax}{}{}%
6053 }{}%
6054 \ifnumberedpar@%
6055 \ifledRcol%
6056 \ifluatex%
6057 \footnotelang@lua[R]%
6058 \fi%
6059 \@ifundefined{xpg@main@language}{if polyglossia
6060 }{}%
6061 {\footnotelang@poly[R]}%
6062 \footnoteoptions@{R}{##1}{true}%
6063 \xright@appenditem{%
6064 \ifbool{indtl@innote}%
6065 {\unexpanded{\let\index\nindex}}%
6066 }{}%
6067 \ifbool{indtl@notenumber}%
6068 {\unexpanded{\let\index\nindex}}%There is no note
number... so
6069 }{}%
6070 \noexpand\Xnote@true%
6071 \unexpanded{\gdef\@currentnotetype{critical}}%
6072 \noexpand\prepare@Xprenotes{#1}%
6073 \noexpand\prepare@edindex@fornote{\ld@nums}%

```

```

6074 \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.
6075 \noexpand\setcounter{stanzaR}{\the\c@stanzaR}%Save
stanzaR counter for footnote
6076 \unexpanded{\def\@this@crossref@start}{\theedtext:
start}%
6077 \unexpanded{\def\@this@crossref@end}{\theedtext:end}%
6078 \expandonce{\@beforeinsertofthisedtext}% Internal for
now, no reason to make it public
6079 \noexpand\parse@annot\l@current@annotR|%
6080 \noexpand\csuse{v#1footnote}{#1}%
6081 {\l@d@nums}{\expandonce\@tag}{\expandonce\content}}
%
6082 \noexpand\Xnote@false%
6083 \unexpanded{\advance\@edindex@fornote@m@ne}%
6084 \ifbool{indtl@innote}%
6085 {\unexpanded{\let\index\orig@@index}}%
6086 {}%
6087 \ifbool{indtl@notenumber}%
6088 {\unexpanded{\let\index\orig@@index}}%
6089 {}%
6090 }to\inserts@listR
6091 \footnoteoptions@{R}{##1}{false}%
6092 \global\advance\insert@countR \@ne%
6093 \else%
6094 \ifluatex%
6095 \footnotelang@lua%
6096 \fi%
6097 \@ifundefined{xpg@main@language}%if polyglossia
6098 {}%
6099 {\footnotelang@poly}%
6100 \footnoteoptions@{L}{##1}{true}%
6101 \xright@appenditem{%
6102 \ifbool{indtl@innote}%
6103 {\unexpanded{\let\index\nindex}}%
6104 {}%
6105 \ifbool{indtl@notenumber}%
6106 {\unexpanded{\let\index\nindex}}%There is no note
number... so
6107 {}%
6108 \noexpand\Xnote@true%
6109 \unexpanded{\gdef\@currentnotetype{critical}}%
6110 \noexpand\prepare@Xprenotes{#1}%
6111 \noexpand\prepare@edindex@fornote{\l@d@nums}%
6112 \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.
6113 \ifl@pairing%
6114 \noexpand\setcounter{stanzaL}{\the\c@stanzaL}%Save
stanzaR counter for footnote
6115 \fi%

```

```

6116 \unexpanded{\def\@this@crossref@start}{\theedtext:
start}%
6117 \unexpanded{\def\@this@crossref@end}{\theedtext:end}%
6118 \expandonce{\@beforeinsertofthisedtext}%Internal for
now, no reason to make it public
6119 \noexpand\parse@annot\l@current@annot|%
6120 \noexpand\csuse{v#1footnote}%
6121 {#1}%
6122 {{\l@d@nums}{\expandonce\@tag}{\expandonce\content
}}%
6123 \noexpand\Xnote@false%
6124 \unexpanded{\advance\@edindex@fornote@m@ne}%
6125 \ifbool{indtl@innote}%
6126 {\unexpanded{\let\index\orig@@index}}%
6127 {}%
6128 \ifbool{indtl@notenumber}%
6129 {\unexpanded{\let\index\orig@@index}}%
6130 {}%
6131 }\to\inserts@list
6132 \global\advance\insert@count \one%
6133 \footnoteoptions@{L}{##1}{false}%
6134 \fi
6135 \else
6136 \csuse{v#1footnote}{#1}{{0|0|0|0|0|0|0}{##1}}%
6137 \fi%
6138 \endgroup%
6139 \else%
6140 \led@err@FootnoteNotInSecondArgEdtext{#1}%
6141 \fi%
6142 }{\@noneed@notesttrue}% End of \nottoggle{Xnonote@#1}{%
6143 \ignorespaces%
6144 }
6145 %

```

Create counter used to determine on which page the previous note was called.

```

6146 \expandafter\newcount\csname #1prevpage@num\endcsname%
6147 \expandafter\newcount\csname #1prevpage@numR\endcsname%
6148 %

```

We need to be able to modify reledmac's footnote macros and restore their

```

6149 \global\csletcs{#1@footnote}{#1footnote}
6150 %

```

XXI.3.4 Set standard display

```

6151 \Xarrangement@normal{#1}%
6152 %

```

End of for critical footnotes.

```

6153 \fi
6154 %

```

XXI.4 For familiar footnotes

Familiar footnotes are those which end with letters. We look for the `nofamiliar` option of `reledmac`.

```
6155 \unless\ifnofamiliar@
6156 %
```

XXI.4.1 Options

```
6157 \newtoggle{parindentX@#1}
6158 \csgdef{wrapcontentX@#1}{}%
6159 \csgdef{hangindentX@#1}{0pt}%
6160 \csgdef{beforeinsertingX@#1}{}%
6161 \csgdef{raggedX@#1}{}%
6162 \csgdef{hsizetwocolX@#1}{0.45 \hsize}%
6163 \csgdef{hsizethreecolX@#1}{.3 \hsize}%
6164 \csgdef{colalignX@#1}{\raggedright}%
6165 \csgdef{notenumfontX@#1}{\normalfont}%
6166 \csgdef{notefontsizeX@#1}{\footnotesize}%
6167 \csgdef{bhooknoteX@#1}{}%
6168 \csgdef{bhookgroupX@#1}{}%
6169 \csgdef{afterruleX@#1}{Opt}
6170 \csgdef{beforenotesX@#1}{1.2em \@plus .6em \@minus .6em}
6171 \csgdef{maxhnotesX@#1}{0.8\vsizex}%
6172 \newtoggle{noteswidthliketwocolumnsX@#1}%
6173 \csgdef{parafootsepX@#1}{}%
6174 \csgdef{afternoteX@#1}{1em plus.4em minus.4em}
6175 \csgdef{widthX@#1}{\hsizex}%
6176 \expandafter\csgdef{innotemarkX@#1}##1{\ledinnotemark{##1}}%
6177 \csgdef{txtbeforenotesX@#1}{}%
6178 \newtoggle{nonoteX@#1}%
6179 \newtoggle{txtbeforenotesX@#1@typeset}%Not directly used by user,
but internal
6180 \newtoggle{txtbeforenotesonlyonceX@#1}%
6181 % End of for familiar footnotes.
6182 % \subsubsection{Create inserts, needed to add notes in foot}
6183 % As regards inserts, see chapter 15 of the TeXBook by D. Knuth.
6184 % \begin{macrocode}
6185 \expandafter\newinsert\csname footins#1\endcsname%
6186 \unless\ifnoledgroup%
6187 \expandafter\newinsert\csname mpfootins#1\endcsname%
6188 \fi%
6189 %
```

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.

```
6190 \global\expandafter\newcommand\csname footnote#1\endcsname[2][{}]{%
6191 \xdef\@currentseries{#1}%
6192 \nottoggle{nonoteX@#1}{%
6193 \begingroup%
```



```

6195         \prepare@prenotesX{#1}%
6196         \newcommand{\content}{##2}%
6197 %

```

If we are using the `\csquotes` package, we reset the quotation level.

```

6198         \ifdefined\csq@qllevel%
6199         \csq@qllevel=0\relax%
6200         \fi%
6201 %

```

If we prepare parallel typesetting, we cannot just increase the footnote counter. Read `reledpar`'s handbook about that (V.1.2 p. 51). If we have a manual footnote mark, use it.

```

6202         \global\expandafter\advance\csname footnote#1@reading\
endcsname by \@ne%
6203         \ifstrepty{##1}{%
6204             \ifboolexpr{bool{!@dpairing} or bool{!@dprintingpages} or
bool{!@dprintingcolumns}}{%
6205                 \ifcsdef{footnote#1@reading\the\csname footnote#1@reading
\endcsname=typeset}%
6206                 {\setcounter{footnote#1}{\csuse{footnote#1@reading\the\
csname footnote#1@reading\endcsname=typeset}}}%
6207                 {\setcounter{footnote#1}{\the\csname footnote#1@reading\
endcsname}}}%
6208             }{%
6209                 \stepcounter{footnote#1}%
6210             }%
6211         }{%
6212 %

```

We also have to check consistency with `\onlysideX` setting.

```

6213         \ifledRcol%
6214         \ifcsstring{onlysideX@#1}{L}{\
led@error@note@called@onrightside{footnote#1}}{%
6215         \else%
6216         \ifcsstring{onlysideX@#1}{R}{\
led@error@note@called@onleftside{footnote#1}}{%
6217         \fi%
6218 %

```

And now, the feature not depending on whether we prepare parallel typesetting

```

6219         \ifstrepty{##1}%
6220         {\protected@csxdef{@thefnmark#1}{\csuse{thefootnote#1}}}%
6221         {\protected@csxdef{@thefnmark#1}{##1}}%
6222         \nottoggle{nomk@}%Nomk is set to true when using \
footnoteXnomk with \parpackage
6223         {\csuse{@footnotemark#1}}%
6224         {}%
6225         \ifluatex%
6226         \xdef\footnote@luatextextdir{\the\textdir}%
6227         \xdef\footnote@luatexpardir{\the\pardir}%
6228         \fi%
6229         \if@ledgroup%
6230         \led@set@index@fornote{#1}%

```

```

6231         \fi%
6232         \csuse{vfootnote#1}{#1}{\expandonce\content}{##1}\
m@mmf@prepare%
6233         \ifbool{indtl@innote}%
6234         {\let\index\orig@@index}%
6235         {}%
6236         \ifbool{indtl@notenumber}%
6237         {\let\index\orig@@index}%
6238         {}%
6239         \endgroup%
6240     }{}%End of the notoggleX@#1
6241 }
6242 %

```

Then define the counters. The \LaTeX counter `footnoteX` is the only one manipulated by the user. This is the one which is printed. The \TeX counter `\footnoteX@reading` is increased at each footnote. It is used for hyperlinks, for using `hyperlink` package, and for getting the correct footnote number when using parallel typesetting (V.1.2 p. 51).

```

6243     \newcounter{footnote#1}
6244     \global\expandafter\renewcommand\csname thefootnote#1\endcsname{\
arabic{footnote#1}}
6245     \expandafter\newcount\csname footnote#1@reading\endcsname%
6246 %

```

Create counter used to determine on which page the previous note was called.

```

6247     \expandafter\newcount\csname prevpage#1@num\endcsname%
6248     \expandafter\newcount\csname prevpage#1@numR\endcsname%
6249     \expandafter\newcount\csname footnote#1@parafootsepX\endcsname%For
the case we use \parafootsepX outside of numbered section
6250 %

```

Add `\let\footnoteX\@gobble` to `\no@expands`.

```

6251     \expandafter\gappto\expandafter\no@expands\expandafter{\expandafter\
let\csname footnote#1\endcsname\@gobble}%
6252 %

```

And now, define `\footnoteXmark` and `\footnoteXtext`, equivalent to classical `\footnotemark` and `\footnotetext`.

```

6253     \expandafter\newcommand\csname footnote#1mark\endcsname[1] []{%
6254         \begingroup%
6255         \prepare@prenotesX{#1}%
6256         \ifstrempy{##1}{%
6257             \stepcounter{footnote#1}%
6258         }{%
6259             \setcounter{footnote#1}{##1}%
6260         }%
6261         \protected@csxdef{@thefnmark#1}{\csuse{thefootnote#1}}%
6262         \csuse{@footnotemark#1}%
6263         \m@mmf@prepare%
6264     \endgroup%
6265 }%
6266     \expandafter\newcommand\csname footnote#1text\endcsname[2] []{%
6267         \begingroup%

```

```

6268         \csuse{vfootnote#1}{#1}{\expandonce{##2}}{##1}%
6269         \endgroup%
6270     }%
6271 %

```

Do not forget to initialize the series.

```

6272     \arrangementX@normal{#1}%
6273     \fi
6274 %

```

XXI.5 The endnotes

Endnotes are commands like `\Xendnote`, where `X` is a series letter. First, we check for the `noend` options.

```

6275     \unless\ifnoend@
6276 %

```

XXI.5.1 The auxiliary file

`\l@d@Xend` Endnotes of all varieties are saved up in a file, one by series, typically named `(jobname).Xend`.
`\ifl@dend@X` `\l@d@end` is the output stream number for this file, and `\ifl@dend@X` is a flag that is
`\l@dend@Xtrue` true when the file is open.
`\l@dend@Xfalse`

```

6277     \expandafter\newwrite\csname l@d@#1end\endcsname%
6278     \expandafter\newif\csname ifl@dend@#1\endcsname%
6279 %

```

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.

```

6280
6281     \global\expandafter\newcommand\csname #1endnote\endcsname[2][1,
usedefault]{%
6282         \advance \edtext@notes by 1%
6283         \bgroup%
6284         \newlinechar='40%
6285         \newcommand{\content}{##2}%
6286         \stepcounter{labidx}%
6287         \expandafter\immediate\expandafter\write\csname l@d@#1end\
endcsname{%
6288             \unexpanded{\def\sw@list@inedtext}%
6289             {\expandafter\unexpanded\expandafter{\sw@inthisedtext}}%
6290             \@percentchar\space%Explicit space, to add a linebreak in the
output file
6291             \noexpand\parse@annot\l@current@annot|\@percentchar\space%
6292             \expandafter\string\csname #1end\endcsname%
6293             {\ifnumberedpar@l@d@nums\fi}%
6294             {\ifnumberedpar@\expandonce\@tag\fi}%

```

```

6295         {\expandonce\content}%
6296         {#1}%
6297         {\unexpanded{##1}}%
6298         {\ifledRcol R\else L\fi}%
6299         {\theedtext}%
6300         \@percentchar%
6301     }%
6302     \egroup%
6303     \ignorespaces%
6304 }%
6305 %

```

XXI.5.3 Tools

The `\Xtoendnotes` command inserts any arbitrary content into the endnote file. It is an alias of the more generalist `\addtoendnotes`

```

6306     \global\expandafter\newcommand\csname #1toendnotes\endcsname[1]{%
6307     \toendnotes[#1]{##1}%
6308     }%
6309
6310     \expandafter\WithSuffix\expandafter\newcommand\csname #1toendnotes\
6311     endcsname*[1]{%
6312     \toendnotes*[#1]{##1}%
6313     }%
6314
6315 %

```

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.

```

6316     \global\cslet{#1end}{\@gobbleseven}
6317
6318 %

```

We need to store the number of times `\doendnotesbysection` is called for one series.

```

6319     \global\expandafter\newcount\csname #1end@bysection\endcsname%
6320 %

```

XXI.5.5 The options

```

6321     \csgdef{Xendwraplemma@#1}{%
6322     \csgdef{Xendwrapcontent@#1}{}%
6323     \csgdef{Xendtwolines@#1}{}%
6324     \csgdef{Xendmorethantwolines@#1}{}%
6325     \newtoggle{Xendtwolinesbutnotmore@#1}{%

```

```

6326 \newtoggle{Xendtwolinesonlyinsamepage@#1}{}%
6327 \newtoggle{Xendlemmadisablefontselection@#1}%
6328 \csgdef{Xendnotenumfont@#1}{\normalfont}%
6329 \csgdef{Xendnotefontsize@#1}{\footnotesize}%
6330 \csgdef{Xendbhooknote@#1}{}%
6331
6332 \csgdef{Xendsublinesep@#1}{\fullstop}%
6333
6334 \csgdef{Xendbeforenumber@#1}{Opt}%
6335 \csgdef{Xendafternumber@#1}{0.5em}%
6336
6337 \csgdef{Xendboxlinenum@#1}{Opt}%
6338 \csgdef{Xendboxlinenumalign@#1}{L}%
6339
6340 \csgdef{Xendboxstartlinenum@#1}{Opt}%
6341 \newtoggle{Xendnotboxingsubline@#1}%
6342 \csgdef{Xendboxendlinenum@#1}{Opt}%
6343
6344 \csgdef{Xendlemmaseparator@#1}{}%
6345 \csgdef{Xendbeforelemmaseparator@#1}{0em}%
6346 \csgdef{Xendafterlemmaseparator@#1}{0.5em}%
6347 \csgdef{Xendinplaceoflemmaseparator@#1}{0.5em}%
6348
6349 \newtoggle{Xendparagraph@#1}%
6350 \csgdef{Xendafternote@#1}{1em plus.4em minus.4em}%
6351 \csgdef{Xendsep@#1}{}%
6352
6353 \csgdef{Xendinplaceofnumber@#1}{Opt}%
6354 \newtoggle{Xendnonumber@#1}%
6355 \newtoggle{Xendnolinenum@#1}%
6356 \newtoggle{Xendnolinenumifannotation@#1}%
6357
6358 \csgdef{Xendhangindent@#1}{Opt}%
6359
6360 \newtoggle{Xendnumberonlyfirstinline@#1}%
6361 \newtoggle{Xendnumberonlyfirstintwolines@#1}%
6362
6363 \newtoggle{Xendlinenumannotationonlyfirst@#1}%
6364 \newtoggle{Xendlinenumannotationonlyfirstintwo@#1}%
6365
6366 \csgdef{Xendbeforesymmlinum@#1}{\csuse{Xendbeforenumber@#1}}%
6367 \csgdef{Xendaftersymmlinum@#1}{\csuse{Xendafternumber@#1}}%
6368
6369 \csgdef{Xendsymmlinum@#1}{}%
6370 \csgdef{Xendsymmlinumannotation@#1}{}%
6371
6372 \csgdef{Xendboxsymmlinum@#1}{Opt}%
6373
6374 \csgdef{Xendbhooklinenum@#1}{}%
6375 \csgdef{Xendehooklinenum@#1}{}%
6376 \csgdef{Xendbhookinplaceofnumber@#1}{}%
6377 \csgdef{Xendehookinplaceofnumber@#1}{}%
6378
6379 \csgdef{Xendlinerangeseparator@#1}{\endashchar}%

```

```

6380      \csgdef{Xendbeforepagenumber@#1}{p.}%
6381      \csgdef{Xendafterpagenumber@#1}{) }%
6382      \csgdef{Xendlineprefixsingle@#1}{}%
6383      \csgdef{Xendlineprefixmore@#1}{}%
6384
6385      \newtoggle{Xendlineflag@#1}
6386
6387      \csgdef{Xendlemmafont@#1}{}%
6388
6389      \csgdef{Xendlinenumannotationposition@#1}{after}%
6390      \csgdef{Xendwraplinenumannotation@#1}{\textsuperscript}%
6391      \newtoggle{Xendnoidenticallinenumannotation@#1}%
6392
6393      \newtoggle{Xendpagenumberonlyfirst@#1}%
6394      \newtoggle{Xendpagenumberonlyfirstifsingle@#1}%
6395      \newtoggle{Xendpagenumberonlyfirstintwo@#1}%
6396      \csgdef{Xendsympagenum@#1}{}%
6397      \csgdef{Xendinplaceofpagenumber@#1}{0pt}%
6398
6399      \csgdef{Xendtxtbeforenotes@#1}{}%
6400      \newtoggle{Xendfirstnote@#1}%Not a hook, but used to apply
6401      Xendtxtbeforenotes
6402
6403      %

```

End of endnotes declaration

```

6404      \fi%
6405      %

```

Dump series in \@series

```

6406      \listxadd{\@series}{#1}
6407      }
6408      }% End of \newseries
6409      %

```

XXI.6 Init standards series (A,B,C,D,E)

```

6410      \expandafter\newseries\expandafter{\default@series}
6411      %

```

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.

```

6412      \newcommand{\seriesatbegin}[1]{%
6413          \StrDel{\@series}{#1}[\@series]%
6414          \edef\@new{%
6415              \listxadd{\@new}{#1}%
6416              \listxadd{\@new}{\@series}%
6417              \xdef\@series{\@new}%

```

```
6418 }
6419 %
```

\seriesatend And `\seriesatend` moves the series to the end of the list.

```
6420 \newcommand{\seriesatend}[1]{%
6421   \StrDel{\@series}{#1}[\@series]%
6422   \edef\@new{%
6423     \listead{\@new}{\@series}%
6424     \listead{\@new}{#1}%
6425     \xdef\@series{\@new}%
6426   }
6427   %
```

XXII.2 Test series order

\ifseriesbefore `\ifseriesbefore{<seriesA>}{<seriesB>}{<true>}{<false>}` expands to `<true>` if `<seriesA>` is printed before `<seriesB>`, or to `<false>` otherwise.

```
6428 \newcommand{\ifseriesbefore}[4]{%
6429   \StrPosition{\@series}{#1}[\@first]%
6430   \StrPosition{\@series}{#2}[\@second]%
6431   \ifnumgreater{\@second}{\@first}{#3}{#4}%
6432 }
6433 %
```

XXII.2.1 Get the first series

In some specific case, we need to know the first series of the list of series.

```
\@getfirstseries34 \newcommand{\@getfirstseries}{%
6435   \ifdefempty{\@series}%
6436   {\xdef\@firstseries{}}%
6437   {\StrChar{\@series}{1}[\@firstseries]}%
6438 }%
6439 %
```

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.

```

6440 \newcommandx{\settoggle@series}[5][4,5,usedefault]{%
6441   \def\do##1{%
6442     \ifcsdef{etb@tgl@#2@##1}{%
6443       \global\settoggle{#2@##1}{#3}%
6444     }{%
6445       \led@warn@BadHookSeries{##1}{#2}%
6446     }%
6447     \ifstrequal{#4}{critical}{
6448       \csuse{Xarrangement@}\csuse{series@display##1}{##1}%
6449     }{}
6450     \ifstrequal{#4}{familiar}{%
6451       \csuse{arrangementX@}\csuse{series@displayX##1}{##1}%
6452     }{}%
6453   }%
6454   \ifstreempty{#1}{%
6455     \dolistloop{\@series}%
6456     \ifstreempty{#5}{}{%
6457       \docsvlist{#5}%
6458     }
6459   }%
6460   {%
6461     \docsvlist{#1}%
6462   }%
6463 }
6464 %

```

`\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.

```

6465 \newcommandx{\setcommand@series}[6][4,5,6=0,usedefault]{%
6466   \def\do##1{
6467     \ifcsdef{#2@##1}{%
6468       \expandafter\global\expandafter\renewcommand\csname#2@##1\endcsname
6469       [#6]{#3}%
6470     }{%
6471       \led@warn@BadHookSeries{##1}{#2}%
6472     }%
6473     \ifstrequal{#4}{critical}{%
6474       \csuse{Xarrangement@\csuse{series@display##1}}{##1}%
6475     }{
6476       \ifstrequal{#4}{familiar}{%
6477         \csuse{arrangementX@\csuse{series@displayX##1}}{##1}%
6478       }{
6479         \ifstreempty{#1}{%
6480           \dolistloop{\@series}%
6481           \ifstreempty{#5}{}%
6482           \docsvlist{#5}
6483         }
6484       }%
6485     }%
6486     \docsvlist{#1}%
6487   }%
6488 }%
6489 %

```

XXII.3.3 Tools to generate options commands

`\newhookcommand@series` `\newhookcommand@series\command` names is a generic command to add new commands for hooks, like `\Xhsizetwocol`. The first argument is the name of the hook, the second a comma-separated list of pseudo-series where the hook can be used, like `appref` in the case of `\Xtwolines`. The second argument is also used to create commands named `\<hookname><pseudoseris>`, like `\Xtwolinesappref`. The third argument

```

6490 \newcommandx{\newhookcommand@series}[3][2,3=0,usedefault]{%
6491   \global\expandafter\newcommand\expandafter*\csname #1\endcsname[2][]{%
6492     \setcommand@series{##1}{#1}{##2}[][#2][#3]%

```

```

6493 }%
6494 \ifstrempy{#2}{}{%
6495   \def\do##1{%
6496     \global\expandafter\newcommand\expandafter*\csname #1##1\endcsname
6497     [1]{%
6498       \csuse{#1}[##1]{###1}%
6499     }%
6500   \docsvlist{#2}%
6501 }%
6502 }
6503 %

```

\newhooktoggle@series \newhooktoggle@series\command names is a generic command to add new commands for a new toggle hook, like \Xnumberonlyfirstinline. The second argument is also used to create commands named \<hookname><pseudoserries>, like \Xtwolinesbutnotmoreappref.

```

6504 \newcommandx{\newhooktoggle@series}[2][2,usedefault]{%
6505   \global\expandafter\newcommandx\expandafter*\csname #1\endcsname[2][1,2={
6506     true},usedefault]{%
6507     \settoggle@series{##1}{#1}{##2}[][#2]%
6508   }%
6509   \ifstrempy{#2}{}{%
6510     \def\do##1{%
6511       \global\expandafter\newcommand\expandafter*\csname #1##1\endcsname{%
6512         \csuse{#1}[##1]%
6513       }%
6514     \docsvlist{#2}%
6515   }%
6516 }
6517 %

```

\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

```

6518 \newcommand{\newhooktoggle@series@reload}[2]{%
6519   \global\expandafter\newcommandx\expandafter*\csname #1\endcsname[2][1,2={
6520     true},usedefault]{%
6521     \settoggle@series{##1}{#1}{##2}[#2]%
6522   }%
6523 }%

```

\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.

```

6524 \newcommand{\newhookcommand@series@reload}[2]{%
6525   \global\expandafter\newcommand\expandafter*\csname #1\endcsname[2][]{%
6526     \setcommand@series{##1}{#1}{##2}[#2]%
6527   }%
6528 }
6529 %

```

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.

```

6530 \unless\ifnocritical@
6531   \newhookcommand@series{Xwrapcontent}%
6532   \newhookcommand@series{Xbeforeinserting}%
6533   \newhookcommand@series{Xlemmafont}%
6534   \newhookcommand@series{Xwraplemma}%
6535   \newhooktoggle@series{Xparindent}
6536   \newhookcommand@series{Xhangindent}
6537   \newhookcommand@series{Xragged}
6538   \newhookcommand@series{Xhsizetwocol}
6539   \newhookcommand@series{Xhsizethreecol}
6540   \newhookcommand@series{Xcolalign}%
6541   \newhookcommand@series{Xnotenumfont}
6542   \newhookcommand@series{Xbhooknote}
6543   \newhookcommand@series@reload{Xbhookgroup}{critical}
6544   \newhookcommand@series{Xboxsymlinenum}%
6545   \newhookcommand@series{Xsymlinenum}
6546   \newhookcommand@series{Xsymlinenumannotation}%
6547   \newhookcommand@series{Xbeforenumber}
6548   \newhookcommand@series{Xtxtbeforenumber}
6549   \newhooktoggle@series{Xnopagenumberifcurrent}%
6550   \newhookcommand@series{Xbeforepagenumber}%
6551   \newhookcommand@series{Xafternumber}
6552   \newhookcommand@series{Xbeforesymlinenum}
6553   \newhookcommand@series{Xaftersymlinenum}
6554   \newhookcommand@series{Xinplaceofnumber}
6555   \newhookcommand@series{Xlemmaseparator}
6556   \newhookcommand@series{Xbeforelemmaseparator}
6557   \newhookcommand@series{Xafterlemmaseparator}
6558   \newhookcommand@series{Xinplaceoflemmaseparator}
6559   \newhookcommand@series{Xtxtbeforenotes}
6560   \newhooktoggle@series{Xtxtbeforenotesonlyonce}%
6561   \newhookcommand@series@reload{Xafterrule}{critical}
6562   \newhooktoggle@series{Xnumberonlyfirstinline}
6563   \newhooktoggle@series{Xnumberonlyfirstintwolines}
6564   \newhooktoggle@series{Xlinenumannotationonlyfirst}%
6565   \newhooktoggle@series{Xlinenumannotationonlyfirstintwo}%
6566   \newhooktoggle@series{Xgroupbyline}%
6567   \newhooktoggle@series{Xgroupbylineseparatetwolines}%
6568   \newhooktoggle@series{Xnonumber}
6569   \newhooktoggle@series{Xnolinenumber}%
6570   \newhooktoggle@series{Xnolinenumberifannotation}%
6571
6572   \newhooktoggle@series{Xpstart}
6573   \newhooktoggle@series{Xpstartonlyfirst}%
6574   \newhooktoggle@series{Xpstarteverytime}%
6575   \newhookcommand@series{Xpstartseparator}%
6576
6577   \newhooktoggle@series{Xstanza}%
6578   \newhooktoggle@series{Xstanzaonlyfirst}%

```

```

6579 \newhookcommand@series{Xstanzaseparator}%
6580
6581 \newhooktoggle@series{Xnonote}%
6582
6583 \newhooktoggle@series{Xonlypstart}
6584 \newhooktoggle@series{Xnonbreakableafternumber}
6585 \newhooktoggle@series{Xlemmadisablefontselection}
6586 \newhookcommand@series@reload{Xmaxhnotes}{critical}
6587 \newhookcommand@series@reload{Xbeforenotes}{critical}
6588 \newhooktoggle@series@reload{Xnoteswidthliketwocolumns}{critical}%
6589 \newhookcommand@series@reload{Xnotefontsize}{critical}
6590
6591 \newhookcommand@series{Xboxlinenum}%
6592 \newhookcommand@series{Xboxlinenumalign}%
6593
6594 \newhookcommand@series{Xboxstartlinenum}%
6595 \newhooktoggle@series{Xnotboxingsubline}%
6596 \newhookcommand@series{Xboxendlinenum}%
6597
6598 \newhookcommand@series{Xafternote}%
6599 \newhookcommand@series{Xparafootsep}
6600 \newhookcommand@series@reload{Xwidth}{critical}%
6601
6602 \newhookcommand@series{Xinnotemark}[] [1]%
6603
6604 \ifundef{\Xhsize}%
6605 {%
6606   \newcommandx{\Xhsize}[2] [1,usedefault]{%
6607     \led@warning@Xhsize@deprecated%
6608     \Xwidth[#1]{#2}%
6609   }%
6610 }%
6611 {}%
6612 \fi
6613 %

```

Now, hooks for critical notes which also apply to crossreferencing and line numbering at the sides of the page.

```

6614 \newhooktoggle@series{Xlineflag}[appref,SEref]
6615 \newhookcommand@series{Xtwolines}[appref,SEref]
6616 \newhookcommand@series{Xmorethantwolines}[appref,SEref]
6617 \newhookcommand@series{Xsublinesep}[appref,SEref,side]%
6618 \newhookcommand@series{Xpagelinesep}[appref,SEref,side]%
6619 \newhooktoggle@series{Xtwolinesbutnotmore}[appref,SEref]
6620 \newhooktoggle@series{Xtwolinesonlyinsamepage}[appref,SEref]
6621 \newhookcommand@series{Xlinerrangeseparator}[appref,SEref]
6622 \newhookcommand@series{Xlinenumannotationposition}[side,appref,SEref]%
6623 \newhookcommand@series{Xwraplinenumannotation}[side,ref,appref,SEref]%
6624 \newhooktoggle@series{Xnoidenticallinenumannotation}[side,ref,appref,SEref]
6625 %

```

XXII.3.5 Options for familiar notes

Before generating the optional commands for familiar notes, we check the `\nofamiliar` option.

```

6626 \unless\ifnofamiliar@
6627   \newhookcommand@series{wrapcontentX}%
6628   \newhookcommand@series{beforeinsertingX}%
6629   \newhooktoggle@series{parindentX}
6630   \newhookcommand@series{hangindentX}
6631   \newhookcommand@series{raggedX}
6632   \newhookcommand@series{hsizetwocolX}
6633   \newhookcommand@series{hsizethreecolX}
6634   \newhookcommand@series{colalignX}%
6635   \newhookcommand@series{notenumfontX}
6636   \newhookcommand@series{bhooknoteX}
6637   \newhookcommand@series@reload{bhookgroupX}{familiar}
6638   \newhookcommand@series@reload{beforenotesX}{familiar}
6639   \newhookcommand@series@reload{maxhnotesX}{familiar}
6640   \newhooktoggle@series@reload{noteswidthliketwocolumnsX}{familiar}%
6641   \newhookcommand@series@reload{afterruleX}{familiar}
6642   \newhookcommand@series@reload{notefontsizeX}{familiar}
6643   \newhookcommand@series{afternoteX}
6644   \newhookcommand@series{parafootsepX}
6645   \newhookcommand@series{txtbeforenotesX}%
6646   \newhooktoggle@series{txtbeforenotesonlyonceX}%
6647   \newhookcommand@series@reload{widthX}{familiar}%
6648   \newhookcommand@series{innotemarkX}[] [1]%
6649   \newhooktoggle@series{nonoteX}%
6650   \ifundef{\hsizeX}%
6651     {%
6652       \newcommandx{\hsizeX}[2][1,usedefault]{%
6653         \led@warning@hsizeX@deprecated%
6654         \widthX[#1]{#2}%
6655       }%
6656     }%
6657   {%
6658 \fi
6659 %

```

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.

```

6660 \unless\ifnoend@
6661   \newhookcommand@series{Xendwraplemma}
6662   \newhookcommand@series{Xendwrapcontent}
6663   \newhookcommand@series{Xendnotenumfont}
6664   \newhookcommand@series{Xendlemmafnt}%
6665   \newhookcommand@series{Xendbhooknote}
6666
6667   \newhookcommand@series{Xendboxlinenum}%
6668   \newhookcommand@series{Xendboxlinenumalign}%
6669

```

```

6670 \newhookcommand@series{Xendboxstartlinenum}%
6671 \newhooktoggle@series{Xendnotboxingsubline}%
6672 \newhookcommand@series{Xendboxendlinenum}%
6673
6674 \newhookcommand@series{Xendnotefontsize}
6675 \newhooktoggle@series{Xendlemmadisablefontselection}
6676 \newhookcommand@series{Xendlemmaseparator}
6677 \newhookcommand@series{Xendbeforelemmaseparator}
6678 \newhookcommand@series{Xendafterlemmaseparator}
6679 \newhookcommand@series{Xendinplaceoflemmaseparator}
6680
6681 \newhookcommand@series{Xendbeforenumber}%
6682 \newhookcommand@series{Xendafternumber}%
6683
6684 \newhooktoggle@series{Xendparagraph}
6685 \newhookcommand@series{Xendafternote}
6686 \newhookcommand@series{Xendsep}
6687
6688 \newhookcommand@series{Xendinplaceofnumber}%
6689 \newhooktoggle@series{Xendnonumber}%
6690 \newhooktoggle@series{Xendnolinenum}%
6691 \newhooktoggle@series{Xendnolinenumifannotation}%
6692
6693 \newhooktoggle@series{Xendnumberonlyfirstinline}%
6694 \newhooktoggle@series{Xendnumberonlyfirstintwelines}%
6695
6696 \newhooktoggle@series{Xendlinenumannotationonlyfirst}%
6697 \newhooktoggle@series{Xendlinenumannotationonlyfirstintwo}%
6698
6699 \newhookcommand@series{Xendsymmlinenum}%
6700 \newhookcommand@series{Xendbeforesymmlinenum}%
6701 \newhookcommand@series{Xendaftersymmlinenum}%
6702 \newhookcommand@series{Xendboxsymmlinenum}%
6703
6704 \newhookcommand@series{Xendsymmlinenumannotation}%
6705
6706 \newhookcommand@series{Xendbhooklinenum}%
6707 \newhookcommand@series{Xendahooklinenum}%
6708 \newhookcommand@series{Xendbhookinplaceofnumber}%
6709 \newhookcommand@series{Xendahookinplaceofnumber}%
6710
6711 \newhookcommand@series{Xendhangindent}%
6712
6713 \newhooktoggle@series{Xendpagenumberonlyfirst}%
6714 \newhooktoggle@series{Xendpagenumberonlyfirstifsingle}%
6715 \newhooktoggle@series{Xendpagenumberonlyfirstintwo}%
6716 \newhookcommand@series{Xendsympagenum}%
6717 \newhookcommand@series{Xendinplaceofpagenumber}%
6718
6719 \newhookcommand@series{Xendtxtbeforenotes}%
6720
6721 \fi
6722 \newhooktoggle@series{Xendlineflag}[apprefwithpage,Serefwithpage]
6723 \newhookcommand@series{Xendtwolines}[apprefwithpage,Serefwithpage]

```

```

6724 \newhookcommand@series{Xendmoreethantwolines}[apprefwithpage,SErefwithpage]
6725 \newhooktoggle@series{Xendtwolinesbutnotmore}[apprefwithpage,SErefwithpage]
6726 \newhooktoggle@series{Xendtwolinesonlyinsamepage}[apprefwithpage,
SErefwithpage]
6727 \newhookcommand@series{Xendlinerrangeseparator}[apprefwithpage,SErefwithpage
]
6728 \newhookcommand@series{Xendbeforepagenumber}[apprefwithpage,SErefwithpage,
SErefonlypage]
6729 \newhookcommand@series{Xendafterpagenumber}[apprefwithpage,SErefwithpage]
6730 \newhookcommand@series{Xendlineprefixsingle}[apprefwithpage,SErefwithpage]
6731 \newhookcommand@series{Xendlineprefixmore}[apprefwithpage,SErefwithpage]
6732 \newhookcommand@series{Xendsublinesep}[apprefwithpage,SErefwithpage]
6733
6734 \newhookcommand@series{Xendlinenumannotationposition}[apprefwithpage,
SErefwithpage]%
6735 \newhookcommand@series{Xendwraplinenumannotation}[apprefwithpage,
SErefwithpage]%
6736 \newhooktoggle@series{Xendnoidenticallinenumannotation}[apprefwithpage,
SErefwithpage]%
6737 %

```

XXII.4 Hooks for a particular footnote

`\newhooktoggle@specific` `\newhooktoggle@specific` is a generic command to create boolean hook specific to a note.

```

6738 \newcommand{\newhooktoggle@specific}[1]{%
6739 \newtoggle{#1@}%
6740 \listgadd{\hooktoggle@specific}{#1}%
6741 \define@key[mac]{truefootnoteoption}{#1}[]{\global\settoggle{#1@}{true}}%
When enabling footnote option
6742 \define@key[mac]{falsefootnoteoption}{#1}[]{\global\settoggle{#1@}{false
}}
6743 }
6744 %

```

`\newhookarg@specific` `\newhookarg@specific` is a generic command to create argumen hook specific to a note.

```

6745 \newcommand{\newhookarg@specific}[1]{%
6746 \listgadd{\hookarg@specific}{#1}%
6747 \define@key[mac]{truefootnoteoption}{#1}{\csgdef{#1@}{#1}}%When enabling
footnote option
6748 \define@key[mac]{falsefootnoteoption}{#1}{\global\csundef{#1@}}%When
disabling footnote option
6749 }
6750 %

```

`\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.

```

6751 \def\hooktoggle@specific{}%
6752 \newcommand{\add@hooktoggle@specific@to@cs}[1]{%
6753   \def\do##1{%
6754     \iftoggle{##1@}{%
6755       \ifcsdef{#1}{%
6756         \csgappto{#1}{\toggletrue{##1@}}%
6757       }{%
6758         \csgdef{#1}{\toggletrue{##1@}}%
6759       }%
6760     }{%
6761       \ifcsdef{#1}{%
6762         \csgappto{#1}{\togglefalse{##1@}}%
6763       }{%
6764         \csgdef{#1}{\togglefalse{##1@}}%
6765       }%
6766     }%
6767   }%
6768   \dolistloop{\hooktoggle@specific}%
6769 }%
6770 %

```

`\hookarg@specific`
`\add@hookarg@specific@to@cs`

The same, but for optional argument of critical footnotes with assigned value.

```

6771 \def\hookarg@specific{}%
6772 \newcommand{\add@hookarg@specific@to@cs}[1]{%
6773   \def\do##1{%
6774     \ifcsvoid{##1@}{}%
6775     \ifcsdef{#1}{%
6776       \csxappto{#1}{%
6777         \noexpand\csdef%
6778           {##1@}{\csname##1@\endcsname}%
6779       }%
6780     }{%
6781       \csxdef{#1}{%
6782         \noexpand\csdef%
6783           {##1@}{\csname##1@\endcsname}%
6784       }%
6785     }%
6786   }%
6787 }%
6788 \dolistloop{\hookarg@specific}%
6789 }%
6790 %

```

And now, we define some hooks specific to a note.

```

6791 \newhooktoggle@specific{fulllines}%
6792 \newhooktoggle@specific{nonum}
6793 \newhooktoggle@specific{nosep}
6794 \newhooktoggle@specific{noprefix}%
6795 \newhooktoggle@specific{prefixmore}%
6796 \newhookarg@specific{linerangesep}
6797 %

```


`\linerangesep@` `\linerangesep@` is defined by the option `linerangesep` of critical notes to change temporarily the line range separator for a specific line. As we have to define it before typesetting the line and undefine it after, we use the family of `xkeyval` package's key.

```
6798 %
```

`\nomk@` `\nomk@` toggle is used by `reledpar` to remove the footnote mark in the text when using `\footnoteXmk`. Read `reledpar` handbook.

```
6799 \newtoggle{nomk@}%
6800 %
```

XXII.5 Alias

`\Xnolemmaseparator` `\Xnolemmaseparator[series]` is just an alias for `\Xlemmaseparator[series]{}`.

```
6801 \newcommand*{\Xnolemmaseparator}[1][1]{\Xlemmaseparator[#1]{}}
6802 %
```

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 \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 \LaTeX are first, then familiar familiar footnotes and finally the critical footnotes.

```
6803 \newcommand*{\l@ddoxtrafeet}{%
6804   \IfStrEq{familiar-critical}{\@fnpos}
6805     {\do@feetX\do@Xfeet}%
6806     {%
6807       \IfStrEq{critical-familiar}{\@fnpos}%
6808         {\do@Xfeet\do@feetX}%
6809         {%
6810           \setbox\@outputbox \vbox{%
6811             \unvbox\@outputbox%
6812             \do@feet@custom@order}{\@fnpos}%
6813           }%
6814         }%
6815     }%
6816 }%
6817
6818 %
```

`\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, L^AT_EX could perform an infinite run.

```

6819 \newcommand{\do@feet@custom@order}[2]{%
6820   \def\do##1{%
6821     \edef\@notesseries{\@firstoftwo##1}%
6822     \edef\@notetype{\@secondoftwo##1}%
6823     \ifdefstring{\@notetype}{critical}%
6824       {\csuse{#1append@Xnotes}{\@notesseries}}%
6825       {\ifdefstring{\@notetype}{familiar}%
6826         {\csuse{#1append@notesX}{\@notesseries}}%
6827         {}%
6828       }%
6829   }%
6830   \expandafter\docsvlist\expandafter{#2}%
6831 }%
6832 %

```

`\do@Xfeet` `\do@Xfeet` is the code extending `\@makecol` to cater to the extra critical feet.

```

6833 \newcommand*{\do@Xfeet}{%
6834   \setbox\@outputbox \vbox{%
6835     \unvbox\@outputbox
6836     \@opXfeet}}
6837 %

```

`\@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
6838 \newcommand{\append@Xnotes}[1]{%
6839   \ifvoid\csuse{#1footins}\else%
6840     \global\skip\csuse{#1footins}=\csuse{Xbeforenotes@#1}%
6841     \global\advance\skip\csuse{#1footins} by\csuse{Xafterrule@#1}%
6842     \print@Xnotes{#1}%
6843     \fi%
6844   }%
6845   %

```

The normal way to add one series, `\print@Xnotes`, is replaced by `reledpar` when using `\Pages`.

```

6846 \newcommand\print@Xnotes[1]{%
6847   \xdef\@currentseries{#1}%
6848   \csuse{#1footstart}{#1}%
6849   \csuse{#1footgroup}{#1}%
6850 }%
6851 %

```

We print all series of notes by looping on them. We check before printing them that they are not voided.

```

6852 \newcommand*{\@opXfeet}{%
6853   \unless\ifnocritical@%
6854     \gdef\firstXseries@{}%
6855     \def\do##1{%

```

```

6856     \append@Xnotes{##1}%
6857   }%
6858   \dolistloop{\@series}%
6859 \fi%
6860 }%
6861 %

```

\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.

```

6862 \newcommand*{\l@ddodoreintrafeet}{%
6863   \IfStrEq{familiar-critical}{\@fnpos}
6864   {\@doreinfeetX\X@doreinfeet}%
6865   {%
6866     \IfStrEq{critical-familiar}{\@fnpos}%
6867     {\X@doreinfeet\@doreinfeetX}%
6868     {\@doreinfeetX\X@doreinfeet}%
6869   }%
6870 }
6871
6872 %

```

\X@doreinfeet \X@doreinfeet is the code for catering for the extra critical footnotes within \@reinserts.

```

6873 \newcommand*{\X@doreinfeet}{%
6874   \unless\ifnocritical@%
6875   \def\do##1{%
6876     \ifvoid\csuse{##1footins}\else%
6877     \insert\csuse{##1footins}{\unvbox\csuse{##1footins}}%
6878     \fi}%
6879   \dolistloop{\@series}
6880 \fi%
6881 }
6882
6883 %

```

\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.

```

6884 \newcommand{\append@notesX}[1]{%
6885   \ifvoid\csuse{footins#1}\else%
6886     \global\skip\csuse{footins#1}=\csuse{beforenotesX@#1}%
6887     \global\advance\skip\csuse{footins#1} by\csuse{afterruleX@#1}%
6888     \print@notesX{#1}%
6889   \fi%
6890 }%
6891 %

```

The normal way to print one series of notes. \print@Xnotes is replaced by reledpar when using \Pages.

```

6892 \newcommand\print@notesX[1]{%
6893   \xdef\@currentseries{#1}%
6894   \csuse{footstart#1}{#1}%

```

```

6895 \csuse{footgroup#1}{#1}%
6896 }%
6897 %

```

We print all the series of notes by looping over them. We check before printing them that they are not voided.

```

6898 \newcommand*{\do@feetX}{%
6899   \unless\ifnofamiliar@%
6900   \gdef\firstseriesX@{}%
6901   \setbox\@outputbox \vbox{%
6902     \unvbox\@outputbox%
6903     \def\do##1{%
6904       \append@notesX{##1}%
6905     }%
6906     \dolistloop{\@series}}%
6907   \fi%
6908 }%
6909
6910 \newcommand{\@doreinfeetX}{%
6911   \unless\ifnofamiliar@%
6912   \def\do##1{%
6913     \ifvoid\csuse{footins##1}\else
6914       \insert%
6915         \csuse{footins##1}
6916         {\unvbox\csuse{footins##1}}%
6917     \fi%
6918   }%
6919   \dolistloop{\@series}%
6920 \fi%
6921 }%
6922 %
6923 %

```

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.)

```

6924 \ifclassloaded{memoir}{%
6925 %

```

memoir is loaded so we use memoir's built in hooks.

```

6926 \g@addto@macro{\m@mdoextrafeet}{\l@ddoxtrafeet}%
6927 \g@addto@macro{\m@mdodoreinextrafeet}{\l@ddodoreinextrafeet}%
6928 }{%
6929 %

```

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.

```

6930 \ifboolexpr{%
6931   test{\@ifpackageloaded{fancyhdr}}%
6932   and test {\ifdef{\latex@makecol}}%
6933 }{%
6934   \patchcmd%
6935     {\latex@makecol}%
6936     {\xdef\@freelist{\@freelist\@midlist}}%
6937     {\xdef\@freelist{\@freelist\@midlist}\l@ddoxtrafeet}%
6938     {}%
6939     {\led@error@fail@patch@makecol}%
6940 }{%
6941   \patchcmd%
6942     {\@makecol}%
6943     {\xdef\@freelist{\@freelist\@midlist}}%
6944     {\xdef\@freelist{\@freelist\@midlist}\l@ddoxtrafeet}%
6945     {}%
6946     {\led@error@fail@patch@makecol}%
6947 }%
6948
6949 \patchcmd%
6950   {\@reinserts}%
6951   {\ifvbox}%
6952   {\l@ddodoreinextrafeet\ifvbox}%
6953   {}%
6954   {\led@error@fail@patch@reinserts}%
6955 }
6956
6957 %

```

It turns out that \@doclearpage also needs modifying.

\if@led@nofoot We have to check if there are any leftover feet.

```

6958 \newif\if@led@nofoot
6959
6960 %

```

```

6961 \@ifclassloaded{memoir}{%
6962 %

```

If the memoir class is loaded, we hook into its modified \@doclearpage.

```

\@mem@extranofeet 6963 \g@addto@macro{\@mem@extranofeet}{%%
6964   \def\do#1{%
6965     \unless\ifnocritical@%
6966       \ifvoid\csuse{#1footins}\else\@mem@nofootfalse\fi%
6967     \fi%
6968     \unless\ifnofamiliar@%
6969       \ifvoid\csuse{footins#1}\else\@mem@nofootfalse\fi%
6970     \fi%
6971   }
6972   \dolistloop{\@series}%

```

```

6973 }%
6974 }{%
6975 %

```

As memoir is not loaded we have patch \@doclearpage.

```

\@led@testifnofoot 76 \newcommand*{\@led@testifnofoot}{%
\@doclearpage 77 \@led@nofoottrue%
6978 \ifvoid\footins\else%
6979 \@led@nofootfalse%
6980 \fi%
6981 \def\do##1{%
6982 \unless\ifnocritical@%
6983 \ifvoid\csuse{##1footins}\else%
6984 \@led@nofootfalse%
6985 \fi%
6986 \fi%
6987 \unless\ifnofamiliar@%
6988 \ifvoid\csuse{footins##1}\else%
6989 \@led@nofootfalse%
6990 \fi%
6991 \fi%
6992 }%
6993 \dolistloop{\@series}%
6994 }%
6995
6996 \pretocmd%
6997 {\@doclearpage}%
6998 {\@led@testifnofoot}%
6999 {}%
7000 {\led@error@fail@patch@@doclearpage}%
7001
7002 \patchcmd%
7003 {\@doclearpage}%
7004 {\ifvoid\footins}%
7005 {\if@led@nofoot}%
7006 {}%
7007 {\led@error@fail@patch@@doclearpage}%
7008
7009 }
7010
7011 %

```

XXIV Page numbering in parallel typsetting

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. 124). The `sameparallelpagenumber` 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 \LaTeX features to determine whether a page is left (even numbered) or right (odd numbered). Consequently, we have to do it by patching \the page , 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 \LaTeX 's \@outputpage macro, as is the page counter in this macro. However, this increase will take account of the options.

$\text{\par@patch@thepage}$ $\text{\par@patch@thepage}$ patches \the page in order to use the value of \par@page counter and not the value of page. It must be called after any redefinition of \the page . That is why we insert it at the end of the \LaTeX macro \pagenumbering , which is called by some \xxxmatter commands. In cases when we are using the memoir class, we insert it at the end of \@mempnum . When using \pagenumbering , we also need to reset the \par@page counter. Consequently, we put $\text{\par@patch@thepage}$ and counter reset in $\text{\par@patch@pagenumbering}$. We also call $\text{\par@patch@thepage}$ at the beginning of the document.

```

7012 \newcommand{\par@patch@thepage}{%
7013   \ifboolexpr{%
7014     bool{sameparallelpagenumber}%
7015     or bool{prevpgnotnumbered}%
7016   }%
7017   {%
7018     \patchcmd{\the page}%
7019       {page}{\par@page}%
7020       {}%
7021       {\led@error@fail@patch@thepage}%
7022   }{}%
7023 }%
7024
7025 \newcommand{\par@patch@pagenumbering}{%
7026   \setcounter{\par@page}{1}%
7027   \par@patch@thepage%
7028 }%
7029
7030 \ifl@dmemoir%
7031   \apptocmd{\@mempnum}%
7032     {\par@patch@pagenumbering}%
7033     {}%
7034     {\led@error@fail@patch@@mempnum}%
7035
7036 \else%
7037   \apptocmd{\pagenumbering}%
7038     {\par@patch@pagenumbering}%
7039     {}%
7040     {\led@error@fail@patch@pagenumbering}%
7041 \fi%
7042
7043 \AtBeginDocument{\par@patch@thepage}%
7044
7045 %

```

`\@outputpage` As its name says, `\@outputpage` is a \TeX macro called in the output routine. It is this macro which increases the page counter. We patch it in order to increase, conditionally, the `par@page` counter, and to set correctly the page number sent to the auxiliary files (`.1`, `.2`, etc.).

```

7046 \AtBeginDocument{%
7047   \set@this@c@page%
7048   \apptocmd{\@outputpage}{%
7049     \ifsameparallelpagenumber%
7050       \ifl@printingpages%
7051         \ifodd\c@page\else%
7052           \stepcounter{par@page}%
7053         \fi%
7054       \else%
7055         \stepcounter{par@page}%
7056       \fi%
7057     \else%
7058       \stepcounter{par@page}%
7059     \fi%
7060   }%
7061   {}%
7062   {\led@error@fail@patch@outputpage}%
7063 }%
7064 \catcode`\#=12%
7065 \AtBeginDocument{%
7066   \apptocmd{\setcounter}{%
7067     \IfStrEq{#1}{page}{\set@this@c@page}{}%
7068   }%
7069   {}%
7070   {\led@error@fail@patch@setcounter}%
7071   \apptocmd{\addtocounter}{%
7072     \IfStrEq{#1}{page}{\set@this@c@page}{}%
7073   }%
7074   {}%
7075   {\led@error@fail@patch@addtocounter}%
7076   \@ifpackageloaded{calc}{%
7077     \apptocmd{\stepcounter}{%Because calc.sty does not use addtocounter in
stepcounter
7078       \IfStrEq{#1}{page}{\set@this@c@page}{}%
7079     }%
7080     {}%
7081     {\led@error@fail@patch@steptocounter}%
7082   }{}%
7083 }%
7084 \catcode`\#=6%
7085 %

```

`\thepar@page` The `par@page` counter.

```

7086 \newcounter{par@page}%
7087 \setcounter{par@page}{1}%
7088 %

```


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.

```
7089 \listcreate{\labelref@list}
7090 %
```

`\zz@@@` Two convenience macros to zero three / four labeling counters in one go.

```
\zz@@@
7091 \newcommand*{\zz@@@}{000|000|000}% Set three counters to zero in one go
7092 \newcommand*{\zz@@@}{000|000|000|000}% Set four counters to zero in one go
7093
7094 %
```

`\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.³⁶

This version of the original EDMAC `\label` uses `\@bsphack` and `\@esphack` to eliminate extra space problems and also use the \TeX write methods for the `.aux` file.

Jesse Billett³⁷ found that the original code could be off by several pages. This version, hopefully cures that, and also allows for non-arabic page numbering.

```
7095 \newcommand*{\edlabel}[1]{%
7096   \leavevmode%
7097   \@bsphack%
7098   \ifboolexpr{bool{ledRcol} or bool{ledRcol@}}{%
7099     \ifXnote@%
7100     \protected@write\@auxout{%
7101       {\string\l@dmake@labelsR\space\thepage|\l@dparsedstartline|\l@dparsedstartsub||\annot@start|\the\c@pstartR|{#1}}%
7102     \ifdef\hypertarget{%
7103       {\Hy@raisedlink{\hypertarget{#1}{}}}%
7104     }%
7105   \else%
7106     \write\linenum@outR{\string\@lab}%
7107     \ifx\labelref@listR\empty%
7108       \xdef\label@refs{\zz@@@}%

```

³⁶The remaining macros in this section were kindly revised by Wayne Sullivan, who substantially improved their efficiency and flexibility.

³⁷(jdb43@cam.ac.uk) via the ctt thread "ledmac cross referencing", 25 August 2003.

```

7109 \else%
7110 \gl@p\labelref@listR\to\label@refs%
7111 \fi%
7112 \ifvmode%
7113 \advancelabel@refs%
7114 \fi%
7115 %

```

Use code from the kernel `\label` command to write the correct page number. Also define an `hypertarget` if `hyperref` package is loaded.

```

7116 \protected@write\@auxout{}\%
7117 {\string\l@dmake@labelsR\space\thepage|\label@refs|\the\c@pstartR
|{#1}}%
7118 \ifdef{\hypertarget}%
7119 {\Hy@raisedlink{\hypertarget{#1}{}}}%
7120 {}%
7121 \fi%
7122 }{%
7123 \ifXnote@%
7124 \ifl@dpairing%pstart or pstartL?
7125 \protected@write\@auxout{}\%
7126 {\string\l@dmake@labels\space\thepage|\l@dparsedstartline|\
l@dparsedstartsub||\annot@start|\the\c@pstartL|{#1}}%
7127 \ifdef{\hypertarget}%
7128 {\Hy@raisedlink{\hypertarget{#1}{}}}%
7129 {}%
7130 \else%
7131 \protected@write\@auxout{}\%
7132 {\string\l@dmake@labels\space\thepage|\l@dparsedstartline|\
l@dparsedstartsub||\annot@start|\the\c@pstart|{#1}}%
7133 \ifdef{\hypertarget}%
7134 {\Hy@raisedlink{\hypertarget{#1}{}}}%
7135 {}%
7136 \fi%
7137 \else%
7138 \write\linenum@out{\string\@lab}%
7139 \ifx\labelref@list\empty%
7140 \xdef\label@refs{\zz@@@}%
7141 \else%
7142 \gl@p\labelref@list\to\label@refs%
7143 \fi%
7144 \ifvmode%
7145 \advancelabel@refs%
7146 \fi%
7147 \ifl@dpairing%Pstart or PstartL?
7148 \protected@write\@auxout{}\%
7149 {\string\l@dmake@labels\space\thepage|\label@refs|\the\c@pstartL
|{#1}}%
7150 \ifdef{\hypertarget}%
7151 {\Hy@raisedlink{\hypertarget{#1}{}}}%
7152 {}%
7153 \else%
7154 \protected@write\@auxout{}\%
7155 {\string\l@dmake@labels\space\thepage|\label@refs|\the\c@pstart

```

```

|{#1}}%
7156     \ifdef{\hypertarget}%
7157         {\Hy@raisedlink{\hypertarget{#1}{}}}%
7158         {}%
7159     \fi%
7160 \fi%
7161 }%
7162 \@esphack}%
7163
7164 %

```

`\advance@label@refs` In cases where `\edlabel` is the first element in a paragraph, we have a problem with line counts, because line counts change only at the first horizontal box of the paragraph. Hence, we need to test `\edlabel` if it occurs at the start of a paragraph. To do so, we use `\ifvmode`. If the test is true, we must advance by one unit the amount of text we write into the `.aux` file. We do so with the `\advance@label@refs` command.

```

7165 \newcounter{line}%
7166 \newcounter{subline}%
7167 \newcounter{absline}%
7168 \newcommand{\advance@label@refs}{%
7169     \setcounter{line}{\expandafter\labelrefsparseline\label@refs}%
7170     \stepcounter{line}%
7171     \setcounter{absline}{\expandafter\labelrefsparseabsline\label@refs}%
7172     \stepcounter{absline}%
7173     \ifsublines@%
7174         \setcounter{subline}{\expandafter\labelrefsparsesubline\label@refs}%
7175         \stepcounter{subline}{1}%
7176         \def\label@refs{\theline|\thesubline|\theabsline}%
7177     \else%
7178         \def\label@refs{\theline|0|\theabsline}%
7179     \fi%
7180 }
7181 \def\labelrefsparseline#1|#2|#3{#1}%
7182 \def\labelrefsparsesubline#1|#2|#3{#2}%
7183 \def\labelrefsparseabsline#1|#2|#3{#3}%
7184 %

```

`\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.

```

7185 \newcommand*{\l@dmake@labels}{%
7186 \def\l@dmake@labels#1|#2|#3|#4|#5|#6|#7{%
7187     \expandafter\ifx\csname the@label\csuse{XR@prefix}#7\endcsname%
7188     \relax%
7189     \else%

```

```

7190 \led@warn@DuplicateLabel{\csuse{XR@prefix}#7}%
7191 \fi
7192 \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
7193 \csgdef{the@label\csuse{XR@prefix}#7}{#1|#2|#3|#4|#5|#6|\relax}%
7194 \ignorespaces}
7195
7196 %

```

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.

```

7197 \AtBeginDocument{%
7198 \def\l@dmake@labels#1|#2|#3|#4|#5|#6|#7{%
7199 }
7200
7201 %

```

\@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.

```

7202
7203 \newcommand*{\@lab}{%
7204 \ifledRcol
7205 \xright@appenditem{\linenumr@p{\line@numR}}|
7206 \ifsublines@ \sublinenumr@p{\subline@numR}\else 0\fi|
7207 \the\absline@numR|
7208 \current@annot%
7209 }%
7210 \to\labelref@listR
7211 \else
7212 \xright@appenditem{%
7213 \linenumr@p{\line@num}|
7214 \ifsublines@ \sublinenumr@p{\subline@num}\else 0\fi|
7215 \the\absline@num|
7216 \current@annot%
7217 }%
7218 \to\labelref@list
7219 \fi}
7220 %

```

\applabel \applabel, if called in \edtext will insert automatically both a starting and an ending label for the current edtext lines.

```

7221 \newcommand*{\applabel}[1]{%
7222 \if@edtext@secondarg@%
7223 %

```

Label should not be already defined.

```

7224 \ifcsundef{the@label#1}{%
7225 \csdef{the@label#1}{applabel}%
7226 }%
7227 {%
7228 \led@warn@DuplicateLabel{#1 (applabel)}%
7229 }%
7230 %

```

Parse the `\edtext` line numbers and annotations.

```

7231 \expandafter\l@dp@rsefootspec\l@d@nums|%
7232 \expandafter\parse@annot\l@current@annot|%
7233 %

```

Use the \LaTeX standard hack for label.

```

7234 \@bspack%
7235 %

```

And now, write the data in the auxiliary file.

```

7236 \ifledRcol%
7237 \protected@write\@auxout{%
7238 {\string\l@dmake@labelsR\space\l@dparsedstartpage|
\l@dparsedstartline|\l@dparsedstartsub||\annot@start|\the\c@pstartR|{#1:
start}}}%
7239 \ifdef{\hypertarget}%
7240 {\Hy@raisedlink{\hypertarget{#1:start}}}%
7241 {}%
7242 \protected@write\@auxout{%
7243 {\string\l@dmake@labelsR\space\l@dparsedendpage|\l@dparsedendline
||\l@dparsedendsub||\annot@end|\the\c@pstartR|{#1:end}}}%
7244 \else%
7245 \ifl@dpairing%pstart or pstartL?
7246 \protected@write\@auxout{%
7247 {\string\l@dmake@labels\space\l@dparsedstartpage|
\l@dparsedstartline|\l@dparsedstartsub||\annot@start|\the\c@pstartL|{#1:
start}}}%
7248 \ifdef{\hypertarget}%
7249 {\Hy@raisedlink{\hypertarget{#1:start}}}%
7250 {}%
7251 \protected@write\@auxout{%
7252 {\string\l@dmake@labels\space\l@dparsedendpage|
\l@dparsedendline|\l@dparsedendsub||\annot@end|\the\c@pstartL|{#1:end}}}%
7253 \else%
7254 \protected@write\@auxout{%
7255 {\string\l@dmake@labels\space\l@dparsedstartpage|
\l@dparsedstartline|\l@dparsedstartsub||\annot@start|\the\c@pstart|{#1:start
}}}%
7256 \ifdef{\hypertarget}%
7257 {\Hy@raisedlink{\hypertarget{#1:start}}}%
7258 {}%
7259 \protected@write\@auxout{%
7260 {\string\l@dmake@labels\space\l@dparsedendpage|
\l@dparsedendline|\l@dparsedendsub||\annot@end|\the\c@pstart|{#1:end}}}%

```

```

7261     \fi%
7262     \fi%
7263 %

```

Use the L^AT_EX standard hack for label.

```

7264     \@esphack%
7265 %

```

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.

```

7266     \ifdef{\hypertarget}%
7267     {%
7268         \ifcsundef{applabel@thisedtext@the\@edtext@level}{\global\
cslet{applabel@thisedtext@the\@edtext@level}{\relax}}{}%
7269         \listcsgadd{applabel@thisedtext@the\@edtext@level}{#1}%
7270     }%
7271     {}%
7272 %

```

Warning if \applabel is called outside of \edtext.

```

7273     \else%
7274         \led@warn@AppLabelOutSecondArgEdtext{#1}%
7275     \fi%
7276 %

```

End of \applabel

```

7277 }%
7278 %

```

\edlabelS \edlabelS and \edlabelE are just used to mark the beginning and the end of a passage.

```

\edlabelE
\edlabelSE
7279 \newcommand{\edlabelS}[1]{%
7280     \edlabel{#1:start}%
7281 }
7282 \newcommand{\edlabelE}[1]{%
7283     \edlabel{#1:end}%
7284 }
7285 \newcommand{\edlabelSE}[1]{%
7286     \edlabelS{#1}%
7287     \edlabelE{#1}%
7288 }
7289 %

```

\wrap@edcrossref \wrap@edcrossref is called around all reledmac crossref commands, except those which start with x. It adds the hyperlink.

```

7290 \newrobustcmd{\wrap@edcrossref}[2]{%
7291     \ifdef{\hyperlink}%
7292         {\hyperlink{#1}{#2}}%
7293         {#2}%
7294     }
7295 %

```

\edpageref If the specified label exists, \edpageref gives its page number.

\xpageref For this reference command, as for the other two, a special version with prefix x is provided for use in places where the command is to be scanned as a number, as in \linenum. These special versions have two limitations: they do not print error messages if the reference is unknown, and they can't appear as the first label or reference command in the file; you must ensure that a \edlabel or a normal reference command appears first, or these x-commands will always return zeros.

ØT_EX already defines a \pageref, so changing the name to \edpageref.

```
7296 \newcommand*\edpageref}[1]{\l@dref@undefined{#1}\wrap@edcrossref{#1}{\l@
7297 \l@dgetref@num{1}{#1}}}%
7298
7299 %
```

\edlineref If the specified label exists, \lineref gives its line number.

\xlineref

```
7300 \newcommand*\edlineref}[1]{%
7301   \l@dref@undefined{#1}%
7302   \wrap@edcrossref{#1}{%
7303     \providetoggle{label@#1@ledRcol}%Required for the first run, when the
7304     label has not yet been parsed on the .aux file
7305     \iftoggle{label@#1@ledRcol}%
7306       {\linenumrepR{\l@dgetref@num{2}{#1}}}%
7307       {\linenumrep{\l@dgetref@num{2}{#1}}}%
7308     \xflagref{#1}%
7309   }%
7310 \newcommand*\xlineref}[1]{\l@dgetref@num{2}{#1}}%
7311
7312 %
```

\sublineref If the specified label exists, \sublineref gives its sub-line number.

\xsublineref

```
7313 \newcommand*\sublineref}[1]{%
7314   \l@dref@undefined{#1}%
7315   \wrap@edcrossref{#1}{%
7316     \providetoggle{label@#1@ledRcol}%Required for the first run, when the
7317     label has not yet been parsed on the .aux file
7318     \iftoggle{label@#1@ledRcol}%
7319       {\sublinenumrepR{\l@dgetref@num{3}{#1}}}%
7320       {\sublinenumrep{\l@dgetref@num{3}{#1}}}%
7321     }%
7322 \newcommand*\xsublineref}[1]{\l@dgetref@num{3}{#1}}%
7323
7324 %
```

\xabslineref If the specified label exists, \xabslineref gives its absolute line number. This is generally used only by reledmac internal macros.

```
7325 \newcommand*\xabslineref}[1]{\l@dgetref@num{4}{#1}}%
7326 %
```

\annotationref If the specified label exists, **\edannotationref** gives the line number annotation with which it is associated.

\xannotationref

```

7327 \newcommand*{\annotationref}[1]{%
7328   \l@dref@undefined{#1}%
7329   \wrap@edcrossref{#1}{%
7330     \l@wrap@ifnotemptybox{\Xwraplinenumannotation@ref}{%
7331       \l@dgetref@num{5}{#1}%
7332     }%
7333   }%
7334 }%
7335 \newcommand*{\xannotationref}[1]{%
7336   \l@dgetref@num{5}{#1}%
7337 }%
7338 %

```

\pstartref If the specified label exists, **\pstartref** gives its pstart number.

\xpstartref

```

7339 \newcommand*{\pstartref}[1]{\l@dref@undefined{#1}\wrap@edcrossref{#1}{\l@dgetref@num{6}{#1}}}%
7340 \newcommand*{\xpstartref}[1]{\l@dgetref@num{6}{#1}}%
7341 %
7342 %

```

\xflagref **\xflagref** finds the side flag of any ref defined with **\edlabel**.

```

7343 \newcommand*{\xflagref}[1]{\l@dgetref@num{7}{#1}}%
7344 %

```

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.

```

7345 \newcommand*{\l@dref@undefined}[1]{%
7346   \expandafter\ifx\csname the@label#1\endcsname\relax
7347     \led@warn@RefUndefined{#1}%
7348   \fi}
7349 %
7350 %

```

\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.

```

7351 \newcommand*{\l@dgetref@num}[2]{%
7352   \expandafter
7353   \ifx\csname the@label#2\endcsname \relax
7354     000%

```



```

7355 \else
7356   \expandafter\expandafter\expandafter
7357   \l@dlabel@parse\csname the@label#2\endcsname|#1%
7358 \fi}
7359
7360 %

```

\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.)

```

7361 \newcommand*\l@dlabel@parse{ }
7362 \def\l@dlabel@parse#1|#2|#3|#4|#5|#6|#7|#8{%
7363   \ifcase #8%
7364     \or #1%
7365     \or #2%
7366     \or #3%
7367     \or #4%
7368     \or #5%
7369     \or #6%
7370     \or #7%
7371   \fi}
7372 %

```

\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.

```

7373 \newcommand*\xxref}[2]{%
7374   {%
7375     \expandafter\ifx\csname the@label#1\endcsname\relax%
7376     \let\@tempa\zz@@@%
7377     \def\@tempc{}%
7378     \else%
7379     \def\@tempa{%
7380       \l@dgetref@num{1}{#1}|%
7381       \l@dgetref@num{2}{#1}|%
7382       \l@dgetref@num{3}{#1}%
7383     }%
7384     \edef\@tempc{%
7385       \l@dgetref@num{5}{#1}%
7386     }%
7387     \fi%
7388     \expandafter\ifx\csname the@label#2\endcsname\relax%

```

```

7389 \let\@tempb\zz@@@%
7390 \def\@tempd{}%
7391 \else%
7392 \def\@tempb{%
7393 \l@getref@num{1}{#2}|%
7394 \l@getref@num{2}{#2}|%
7395 \l@getref@num{3}{#2}%
7396 }%
7397 \edef\@tempd{%
7398 \l@getref@num{5}{#2}%
7399 }%
7400 \fi%
7401 \global\appto\@beforeinsertofthisedtext{%
7402 \def\@this@crossref@start{#1}%
7403 }%
7404 \global\appto\@beforeinsertofthisedtext{%
7405 \def\@this@crossref@end{#2}%
7406 }%
7407 \linenum{\@tempa|\@tempb}%
7408 \edef\@tempe{\@tempc|\@tempd}%
7409 \expandafter\lineannot\expandafter{\@tempe}%
7410 }%
7411 }%
7412
7413
7414 %

```

\appref \appref, \SEref, \apprefwithpage, \SErefwithpage and \SEonlypage print cross-ref to some start / end lines defined by specific commands. It prints the lines as they should be printed in the apparatus (critical notes for not suffixed versions, endnotes for suffixed versions).

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.

```

7415 \def\Xtwolines@appref{}%
7416 \def\Xtwolines@SEref{}%
7417
7418 \def\Xmorethantwolines@appref{}%
7419 \def\Xmorethantwolines@SEref{}%
7420
7421 \def\Xlinerangeseparator@appref{\endashchar}%
7422 \def\Xlinerangeseparator@SEref{\endashchar}%
7423
7424 \def\Xsublinesep@appref{\fullstop}%
7425 \def\Xsublinesep@SEref{\fullstop}%
7426
7427 \def\Xpagelinesep@appref{\fullstop}%
7428 \def\Xpagelinesep@SEref{\fullstop}%
7429
7430
7431 \newtoggle{Xtwolinesbutnotmore@appref}%
7432 \newtoggle{Xtwolinesbutnotmore@SEref}%
7433

```

```

7434 \newtoggle{Xtwolinesonlyinsamepage@appref}%
7435
7436 \newtoggle{Xtwolinesonlyinsamepage@SEref}%
7437
7438 \newtoggle{Xlineflag@appref}%
7439 \toggletrue{Xlineflag@appref}%Here exception
7440 \newtoggle{Xlineflag@SEref}%
7441 \toggletrue{Xlineflag@SEref}%Here exception
7442
7443 \newtoggle{Xlinenumannotationonlyfirst@SEref}%Never changed, just for the
test in \setprintlines
7444 \newtoggle{Xlinenumannotationonlyfirst@appref}%Idem
7445 \newtoggle{Xnolinenumber@SEref}%Idem
7446 \newtoggle{Xnolinenumber@appref}%Idem
7447 \newtoggle{Xnolinenumberifannotation@SEref}%Idem
7448 \newtoggle{Xnolinenumberifannotation@appref}%Idem
7449
7450
7451 \def\Xlinenumannotationposition@appref{after}%
7452 \def\Xlinenumannotationposition@SEref{after}%
7453
7454 \def\Xwraplinenumannotation@appref{\textsuperscript}%
7455 \def\Xwraplinenumannotation@SEref{\textsuperscript}%
7456 \newtoggle{Xnoidenticallylinenumannotation@appref}%
7457 \newtoggle{Xnoidenticallylinenumannotation@SEref}%
7458
7459 \def\Xendtwolines@apprefwithpage{}%
7460 \def\Xendtwolines@SErefwithpage{}%
7461
7462 \def\Xendmorethantwolines@apprefwithpage{}%
7463 \def\Xendmorethantwolines@SErefwithpage{}%
7464
7465 \def\Xendlinerangeseparator@apprefwithpage{\endashchar}
7466 \def\Xendlinerangeseparator@SErefwithpage{\endashchar}
7467 \def\Xendlinerangeseparator@SErefonlypage{\endashchar}
7468
7469 \def\Xendbeforepagenumber@apprefwithpage{p.}%
7470 \def\Xendbeforepagenumber@SErefwithpage{p.}%
7471 \def\Xendbeforepagenumber@SEonlypage{p.}%
7472
7473 \def\Xendafterpagenumber@apprefwithpage{} }%
7474 \def\Xendafterpagenumber@SErefwithpage{} }%
7475
7476
7477 \def\Xendlineprefixsingle@apprefwithpage{}%
7478 \def\Xendlineprefixsingle@SErefwithpage{}%
7479
7480 \def\Xendlineprefixmore@apprefwithpage{}%
7481 \def\Xendlineprefixmore@SErefwithpage{}%
7482
7483 \newtoggle{Xendtwolinesbutnotmore@apprefwithpage}%
7484 \newtoggle{Xendtwolinesbutnotmore@SErefwithpage}%
7485
7486 \def\Xendsublinesep@apprefwithpage{\fullstop}%

```

```

7487 \def\Xendsublinesep@SErefwithpage{\fullstop}%
7488
7489 \newtoggle{Xendtwolinesonlyinsamepage@apprefwithpage}%
7490 \newtoggle{Xendtwolinesonlyinsamepage@SErefwithpage}%
7491
7492 \newtoggle{Xendlineflag@apprefwithpage}
7493 \toggletrue{Xendlineflag@apprefwithpage}%Here, exception
7494 \newtoggle{Xendlineflag@SErefwithpage}
7495 \toggletrue{Xendlineflag@SErefwithpage}%Here, exception
7496
7497
7498 \def\Xendlinenumannotationposition@apprefwithpage{after}%
7499 \def\Xendlinenumannotationposition@SErefwithpage{after}%
7500
7501 \def\Xendlinenumannotationposition@apprefwithpage{after}%
7502 \def\Xendlinenumannotationposition@apprefwithpage@SErefwithpage{after}%
7503
7504 \def\Xendwraplinenumannotation@apprefwithpage{\textsuperscript}%
7505 \def\Xendwraplinenumannotation@SErefwithpage{\textsuperscript}%
7506
7507 \newtoggle{Xendnoidenticallinenumannotation@apprefwithpage}%
7508 \newtoggle{Xendnoidenticallinenumannotation@SErefwithpage}%
7509
7510
7511 \newtoggle{Xendnolinenum@SErefwithpage}%Never changed, only for test in
    \setprintendlines
7512 \newtoggle{Xendnolinenum@apprefwithpage}%Idem
7513 \newtoggle{Xendnolinenum@ifannotation@SErefwithpage}%Idem
7514 \newtoggle{Xendnolinenum@ifannotation@apprefwithpage}%Idem
7515
7516 %

```

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.

```

7517
7518 \newtoggle{Xendlinenumannotationonlyfirst@SEref}%Never changed, just for
    the test in \setprintendlines
7519 \newtoggle{Xendlinenumannotationonlyfirst@SErefwithpage}%Idem
7520 \newtoggle{Xendlinenumannotationonlyfirst@appref}%Idem
7521 \newtoggle{Xendlinenumannotationonlyfirst@apprefwithpage}%Idem
7522
7523 \gdef\Xboxstartlinenum@appref{0pt}
7524 \gdef\Xboxstartlinenum@SEref{0pt}
7525 \newtoggle{Xnotboxingsubline@appref}%
7526 \newtoggle{Xnotboxingsubline@SEref}%
7527
7528 \gdef\Xboxendlinenum@appref{0pt}
7529 \gdef\Xboxendlinenum@SEref{0pt}
7530
7531 \gdef\Xendboxstartlinenum@apprefwithpage{0pt}
7532 \gdef\Xendboxstartlinenum@SErefwithpage{0pt}
7533 \newtoggle{Xendnotboxingsubline@apprefwithpage}%
7534 \newtoggle{Xendnotboxingsubline@SErefwithpage}%

```

```

7535 \gdef\Xendboxendlinenum@apprefwithpage{0pt}
7536 \gdef\Xendboxendlinenum@SErefwithpage{0pt}
7537
7538
7539 \newtoggle{Xendpagenumberonlyfirst@apprefwithpage}
7540 \newtoggle{Xendpagenumberonlyfirst@SErefwithpage}
7541
7542 \newtoggle{Xendpagenumberonlyfirstifsingle@apprefwithpage}
7543 \newtoggle{Xendpagenumberonlyfirstifsingle@SErefwithpage}
7544
7545 \newtoggle{Xendpagenumberonlyfirstintwo@apprefwithpage}
7546 \newtoggle{Xendpagenumberonlyfirstintwo@SErefwithpage}
7547
7548 \gdef\Xendsympagenum@apprefwithpage{}
7549 \gdef\Xendsympagenum@SErefwithpage{}
7550
7551 \gdef\Xendinplaceofpagenumber@apprefwithpage{}
7552 \gdef\Xendinplaceofpagenumber@SErefwithpage{}
7553
7554 %

```

Now, declare the default values of \@apprefprefixsingle and \@apprefprefixmore, \@SErefprefix, \@SErefprefixmore and the commands which defines them.

```

7555 \newcommand\@apprefprefixsingle{}%
7556 \newcommand\@SErefprefixsingle{}%
7557
7558 \newcommand\@apprefprefixmore{}%
7559 \newcommand\@SErefprefixmore{}%
7560
7561 \newcommand{\setapprefprefixsingle}[1]{%
7562   \gdef\@apprefprefixsingle{#1}%
7563 }
7564 \newcommand{\setSErefprefixsingle}[1]{%
7565   \gdef\@SErefprefixsingle{#1}%
7566 }
7567
7568 \newcommand{\setapprefprefixmore}[1]{%
7569   \gdef\@apprefprefixmore{#1}%
7570 }
7571 \newcommand{\setSErefprefixmore}[1]{%
7572   \gdef\@SErefprefixmore{#1}%
7573 }
7574
7575 %

```

And not \setSErefonlypageprefixsingle and \setSErefonlypageprefixmore.

```

7576 \newcommand{\setSErefonlypageprefixsingle}[1]{%
7577   \gdef\SErefonlypage@prefixsingle{#1}%
7578 }%
7579 \newcommand{\setSErefonlypageprefixmore}[1]{%
7580   \gdef\SErefonlypage@prefixmore{#1}%
7581 }%
7582 %

```

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.

```

7583
7584 \newcommandx{\appref}[2][1,usedefault]{\reformatted@{#1}{#2}{appref}}
7585 \newcommandx{\Seref}[2][1,usedefault]{\reformatted@{#1}{#2}{Seref}}
7586
7587 \newcommandx{\apprefwithpage}[2][1,usedefault]{\reformattedwithpage@
7588 {#1}{#2}{appref}}
7589 \newcommandx{\Serefwithpage}[2][1,usedefault]{\reformattedwithpage@
7590 {#1}{#2}{Seref}}
7591
7592 \newcommand{\reformatted@}[3]{%
7593   \def\do##1{%
7594     \setkeys[mac]{truefootnoteoption}{##1}%
7595   }%
7596   \notblank{#1}{\docsvlist{#1}}{}%
7597   \xdef\@currentseries{#3}%
7598   \iftoggle{noprefix@}{}%
7599   {%
7600     \ifcempty{@#3prefixmore}%
7601     {\@apprefprefixsingle}%
7602     {%
7603       \IfEq{\xlineref{#2:start}}{\xlineref{#2:end}}%
7604       {\iftoggle{prefixmore@}%
7605        {\csuse{@#3prefixmore}}%
7606        {\csuse{@#3prefixsingle}}}%
7607     }%
7608     {\csuse{@#3prefixmore}}%
7609   }%
7610 }%
7611 \ifboolexpr{%
7612   test{\ifcsundef{the@label#2:start}}%
7613   or test{\ifcsundef{the@label#2:end}}%
7614 }%
7615 {\led@warn@pairRefUndefined{#2}\nfss@text{\reset@font\bfseries ??}}%
7616 {%
7617   \def\@this@crossref@start{#2:start}%
7618   \def\@this@crossref@end{#2:end}%
7619   \xdef\annot@start{\xannotationref{#2:start}}%
7620   \xdef\annot@end{\xannotationref{#2:end}}%
7621   \printlines\xpageref{#2:start}|\xlineref{#2:start}|\xsublineref{#2:
start}|\xpageref{#2:end}|\xlineref{#2:end}|\xsublineref{#2:end}|\relax|\
xflagref{#2:start}|}%
7622 }%
7623 \def\do##1{%
7624   \setkeys[mac]{falsefootnoteoption}{##1}%
7625 }%
7626 \notblank{#1}{\docsvlist{#1}}{}%

```

```

7627 }%
7628
7629 \newcommand{\reformattedwithpage@}[3]{%
7630   \def\do##1{%
7631     \setkeys[mac]{truefootnoteoption}{##1}%
7632   }%
7633   \notblank{#1}{\docsvlist{#1}}{%
7634     \xdef\@currentseries{#3withpage}%
7635     \ifboolexpr{%
7636       test{\ifcsundef{the@label#2:start}}%
7637       or test{\ifcsundef{the@label#2:end}}%
7638     }%
7639     {\led@warn@pairRefUndefined{#2}\nfss@text{\reset@font\bfseries ??}}%
7640     {%
7641       \def\@this@crossref@start{#2:start}%
7642       \def\@this@crossref@end{#2:end}%
7643       \xdef\annot@start{\xannotationref{#2:start}}%
7644       \xdef\annot@end{\xannotationref{#2:end}}%
7645       \printendlines\xpageref{#2:start}|\xlineref{#2:start}|\xsublineref{#2:
start}|\xpageref{#2:end}|\xlineref{#2:end}|\xsublineref{#2:end}|\relax|\
xflagref{#2:start}}%
7646     }%
7647     \def\do##1{%
7648       \setkeys[mac]{falsefootnoteoption}{##1}%
7649     }%
7650     \notblank{#1}{\docsvlist{#1}}{%
7651 }%
7652
7653 \newcommand{\reformattedonlypage@}[3]{%
7654   \def\do##1{%
7655     \setkeys[mac]{truefootnoteoption}{##1}%
7656   }%
7657   \notblank{#1}{\docsvlist{#1}}{%
7658     \xdef\@currentseries{#3onlypage}%
7659     \ifboolexpr{%
7660       test{\ifcsundef{the@label#2:start}}%
7661       or test{\ifcsundef{the@label#2:end}}%
7662     }%
7663     {\led@warn@pairRefUndefined{#2}\nfss@text{\reset@font\bfseries ??}}%
7664     {\ifnumequal{\xpageref{#2:end}}{\xpageref{#2:start}}%
7665       {%
7666         \ifcsvoid{#3onlypage@prefixsingle}%
7667         {}%
7668         {\csletcs{Xendbeforepagenumber@#3onlypage}{#3onlypage@prefixsingle
}}%
7669         \printnnum{%
7670           \wrap@edcrossref{#2:start}{\xpageref{#2:start}}%
7671         }%
7672       }%
7673     }%
7674     \ifcsvoid{#3onlypage@prefixmore}%
7675     {%
7676       {\csletcs{Xendbeforepagenumber@#3onlypage}{#3onlypage@prefixmore}}%
7677     }%

```

```

7678 \printnpnum{%
7679 \wrap@edcrossref{#2:start}{\xpageref{#2:start}}}%
7680 \linerangesep@%
7681 \wrap@edcrossref{#2:end}{\xpageref{#2:end}}}%
7682 }%
7683 \else%
7684 \printnpnum{%
7685 \wrap@edcrossref{#2:start}{\xpageref{#2:start}}}%
7686 \csuse{Xendlinerangeseparator@{\currentseries}}%
7687 \wrap@edcrossref{#2:end}{\xpageref{#2:end}}}%
7688 }%
7689 \fi%
7690 }%
7691 }%
7692 \def\do##1{%
7693 \setkeys[mac]{falsefootnoteoption}{##1}%
7694 }%
7695 \notblank{#1}{\docsvlist{#1}}{}%
7696 }%
7697 %

```

\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`.

```

7698 \newcommand*{\edmakelabel}[2]{\expandafter\xdef\csname the@label#1\
endcsname{#2}}
7699
7700 %

```

(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. 139 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`.

```

7701 \pretocmd{\XR@test}{%
7702 {\XR@test@mac+++#1#2#3#4+++}}%
7703 {}%
7704 {}%
7705 %

```


`\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`.

```
7706 \long\def\xR@test@mac+++#1+++{\XR@test@mac@test#1}
7707 %
```

`\XR@test@mac@test` And finally, `\XR@test@mac@test` does the job. This code is based on the `\XR@test` macro of the `xr` package. However, not that the `\XR@prefix` is not called here, but it is integrated directly in `\l@dmake@labels` and `\l@dmake@labelsR`.

```
7708 \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
7709 \ifx#1\l@dmake@labels%
7710 \l@dmake@labels#2%
7711 \else
7712 \ifx#1\l@dmake@labelsR%
7713 \l@dmake@labelsR #2%
7714 \fi%
7715 \fi%
7716 }%
7717 %
```

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.

```
7718 \pretocmd{\@xympar}%
7719 {\ifnumberedpar@
7720 \led@warn@NoMarginpars
7721 \esphack
7722 \else}%
7723 {}%
7724 {}%
7725
7726 \apptocmd{\@xympar}%
7727 {\fi}%
7728 {}
7729 {}
7730
7731 %
```

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

```

7732 \newcount\sidenote@margin
7733 \newcommand*{\sidenotemargin}[1]{%
7734   \l@dgetsidenote@margin{#1}%
7735   \ifnum\@l@tempcntb>\m@ne
7736     \ifledRcol
7737       \global\sidenote@marginR=\@l@tempcntb
7738     \else
7739       \global\sidenote@margin=\@l@tempcntb
7740     \fi
7741   \fi}}
7742 \newcommand*{\l@dgetsidenote@margin}[1]{%
7743   \def\@tempa{#1}\def\@tempb{left}%
7744   \ifx\@tempa\@tempb
7745     \@l@tempcntb \z@
7746   \else
7747     \def\@tempb{right}%
7748     \ifx\@tempa\@tempb
7749       \@l@tempcntb \@ne
7750     \else
7751       \def\@tempb{outer}%
7752       \ifx\@tempa\@tempb
7753         \@l@tempcntb \tw@
7754       \else
7755         \def\@tempb{inner}%
7756         \ifx\@tempa\@tempb
7757           \@l@tempcntb \thr@@
7758         \else
7759           \led@warn@BadSidenotemargin
7760           \@l@tempcntb \m@ne
7761         \fi
7762       \fi
7763     \fi
7764   \fi}
7765 \sidenotemargin{right}
7766 %
7767 %

```

\l@dlp@rbox We need two boxes to store sidenote texts.

```

\l@drp@rbox
7768 \newbox\l@dlp@rbox
7769 \newbox\l@drp@rbox
7770
7771 %

```

\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
7772 \newdimen\ledlsnotewidth \ledlsnotewidth=\marginparwidth
\ledrsnotesep

```

```

\ledlsnotefontsetup
\ledrsnotefontsetup

```

```

7773 \newdimen\ledrsnotewidth \ledrsnotewidth=\marginparwidth
7774 \newdimen\ledlsnotesep \ledlsnotesep=\linenumsep
7775 \newdimen\ledrsnotesep \ledrsnotesep=\linenumsep
7776 \newcommand*\ledlsnotefontsetup{\raggedleft\footnotesize}
7777 \newcommand*\ledrsnotefontsetup{\raggedright\footnotesize}
7778
7779 %

```

`\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@
7780 \newcommand*\ledleftnote}[1]{%
7781   \ledsidenote@{\l@dlsnote{#1}}%
7782 }%
7783 \newcommand*\ledrightnote}[1]{%
7784   \ledsidenote@{\l@drsnote{#1}}%
7785 }%
7786 \newcommand*\ledsidenote}[1]{%
7787   \ledsidenote@{\l@dcsnote{#1}}%
7788 }%
7789 \newcommand*\ledinnernote}[1]{%
7790   \ledsidenote@{\l@disnote{#1}}%
7791 }%
7792 \newcommand*\ledouternote}[1]{%
7793   \ledsidenote@{\l@dosnote{#1}}%
7794 }%
7795 \newcommand*\ledsidenote@[1]{%
7796   \edtext{}{%
7797     \advance\edtext@notes@start by 1%
7798     #1%
7799   }%+
7800 }%
7801 %

```

`\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
7802 \newif\ifrightnoteup
7803 \rightnoteuptrue
7804
7805 \newcommand*\l@dlsnote}[1]{%
7806   \begingroup%
7807   \newcommand{\content}{#1}%
7808   \ifnumberedpar@
7809     \ifledRcol%
7810       \xright@appenditem{\noexpand\l@dlsnote{\expandonce\content}}{%
7811         \to\inserts@listR
7812       }%
7813     \else%
7814       \xright@appenditem{\noexpand\l@dlsnote{\expandonce\content}}{%
7815         \to\inserts@list

```

```

7816 \global\advance\insert@count \@ne%
7817 \fi
7818 \fi%
7819 \ignorespaces%
7820 \endgroup%
7821 }%
7822
7823 \newcommand*{\l@drsnote}[1]{%
7824 \begingroup%
7825 \newcommand{\content}{#1}%
7826 \ifnumberedpar@
7827 \ifledRcol%
7828 \xright@appenditem{\noexpand\vl@drsnote{\expandonce\content}}{%
7829 \to\inserts@listR
7830 \global\advance\insert@countR \@ne%
7831 \else%
7832 \xright@appenditem{\noexpand\vl@drsnote{\expandonce\content}}{%
7833 \to\inserts@list
7834 \global\advance\insert@count \@ne%
7835 \fi
7836 \fi\ignorespaces%
7837 \endgroup%
7838 }%
7839
7840 \newcommand*{\l@dcnote}[1]{%
7841 \begingroup%
7842 \newcommand{\content}{#1}%
7843 \ifnumberedpar@
7844 \ifledRcol%
7845 \xright@appenditem{\noexpand\vl@dcnote{\expandonce\content}}{%
7846 \to\inserts@listR
7847 \global\advance\insert@countR \@ne%
7848 \else%
7849 \xright@appenditem{\noexpand\vl@dcnote{\expandonce\content}}{%
7850 \to\inserts@list
7851 \global\advance\insert@count \@ne%
7852 \fi
7853 \fi\ignorespaces%
7854 \endgroup%
7855 }%
7856
7857 \newcommand*{\l@disnote}[1]{%
7858 \begingroup%
7859 \newcommand{\content}{#1}%
7860 \ifnumberedpar@%
7861 \ifledRcol%
7862 \xright@appenditem{\noexpand\vl@disnote{\expandonce\content}}{%
7863 \to\inserts@listR%
7864 \global\advance\insert@countR \@ne%
7865 \else%
7866 \xright@appenditem{\noexpand\vl@disnote{\expandonce\content}}{%
7867 \to\inserts@list%
7868 \global\advance\insert@count \@ne%
7869 \fi%

```

```

7870 \fi\ignorespaces%
7871 \endgroup%
7872 }%
7873
7874 \newcommand*{\l@dosnote}[1]{%
7875 \begingroup%
7876 \newcommand{\content}{#1}%
7877 \ifnumberedpar%
7878 \ifledRcol%
7879 \xright@appenditem{\noexpand\l@dosnote{\expandonce\content}}{%
7880 \to\inserts@listR%
7881 \global\advance\insert@countR \@ne%
7882 \else%
7883 \xright@appenditem{\noexpand\l@dosnote{\expandonce\content}}{%
7884 \to\inserts@list%
7885 \global\advance\insert@count \@ne%
7886 \fi%
7887 \fi\ignorespaces%
7888 \endgroup%
7889 }%
7890
7891 %

```

\vl@dlsnote Put the left/right text into boxes, but just save the moveable text. **\l@dcsnotetext**, **\vl@drsnote** **\l@dcsnotetext@l** and **\l@dcsnotetext@r** are etoolbox's lists which will store the content of sidenotes. We store the content in lists, because we need to loop later on them, in case many sidenote co-exist for the same line. That is there some special test **\vl@disnote** to do, in order to:

- Store the content of **\ledsidenote** to **\l@dcsnotetext** in any cases.
- Store the content of **\rightsidenote** to:
 - **\l@dcsnotetext** if **\ledsidenote** is to be put on right.
 - **\l@dcsnotetext@r** if **\ledsidenote** is to be put on left.
- Store the content of **\leftsidenote** to:
 - **\l@dcsnotetext** if **\ledsidenote** is to be put on left.
 - **\l@dcsnotetext@l** if **\ledsidenote** is to be put on right.

\vl@disnote and **\vl@dosnote** just call **\vl@dlsnote** or **\vl@drsnote**, depending on the page.

```

7892 \newcommand*{\vl@dlsnote}[1]{%
7893 \ifledRcol%
7894 \@l@tempcntb=\sidenote@marginR%
7895 \ifnum\@l@tempcntb>\@ne%
7896 \advance\@l@tempcntb by\page@numR%
7897 \fi%
7898 \else%
7899 \@l@tempcntb=\sidenote@margin%
7900 \ifnum\@l@tempcntb>\@ne%
7901 \advance\@l@tempcntb by\page@num%

```

```

7902 \fi%
7903 \fi%
7904 \ifodd\@l@tempcntb%
7905 \listgadd{\l@dcsnotetext@l}{#1}%
7906 \else%
7907 \listgadd{\l@dcsnotetext}{#1}%
7908 \fi
7909 }
7910 \newcommand*{\vl@drsnote}[1]{%
7911 \ifledRcol@%
7912 \@l@tempcntb=\sidenote@marginR%
7913 \ifnum\@l@tempcntb>\@ne%
7914 \advance\@l@tempcntb by\page@numR%
7915 \fi%
7916 \else%
7917 \@l@tempcntb=\sidenote@margin%
7918 \ifnum\@l@tempcntb>\@ne%
7919 \advance\@l@tempcntb by\page@num%
7920 \fi%
7921 \fi%
7922 \ifodd\@l@tempcntb%
7923 \listgadd{\l@dcsnotetext}{#1}%
7924 \else%
7925 \listgadd{\l@dcsnotetext@r}{#1}%
7926 \fi%
7927 }
7928 \newcommand*{\vl@dcsnote}[1]{\listgadd{\l@dcsnotetext}{#1}}
7929
7930 \newcommand{\vl@disnote}[1]{%
7931 \ifledRcol@%
7932 \@tempcnta=\page@numR%
7933 \else%
7934 \@tempcnta=\page@num%
7935 \fi%
7936 \ifodd\@tempcnta% ODD => right page => inner side = left side
7937 \vl@dlsnote{#1}%
7938 \else%
7939 \vl@drsnote{#1}%
7940 \fi%
7941 }%
7942
7943 \newcommand{\vl@dosnote}[1]{%
7944 \ifledRcol@%
7945 \@tempcnta=\page@numR%
7946 \else%
7947 \@tempcnta=\page@num%
7948 \fi%
7949 \ifodd\@tempcnta% ODD => right page => outer side = right side
7950 \vl@drsnote{#1}%
7951 \else%
7952 \vl@dlsnote{#1}%
7953 \fi%
7954 }%
7955

```

7956 %

`\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.

```
7957 \newcommand*{\setl@dlp@rbox}[1]{%
7958   \begingroup%
7959   \parindent\z@\hspace=\ledlsnotewidth%
7960   \ledlsnotefontsetup%We kept it outside of the vbox, because can affect
the ragging
7961   \global\setbox\l@dlp@rbox%
7962   \ifleftnoteup%
7963     =\vbox to\z@{\ledlsnotefontsetup\vss #1}}%We put \
ledlsnotefontsetup inside footnote because required for color command. Note
the {} to keep setting local.
7964   \else%
7965     =\vbox to 0.70\baselineskip{\ledlsnotefontsetup\strut#1\vss}}%
7966   \fi%
7967   \endgroup%
7968 }
7969
7970 \newcommand*{\setl@drp@rbox}[1]{%
7971   \begingroup%
7972   \parindent\z@\hspace=\ledrsnotewidth%
7973   \ledrsnotefontsetup%We kept it outside of the vbox, because can affect
the ragging
7974   \global\setbox\l@drp@rbox%
7975   \ifrightrightnoteup%
7976     =\vbox to\z@{\ledrsnotefontsetup\vss#1}}%We put \ledrsnotefontsetup
inside footnote because required for color command. Note the {} to keep
setting local.
7977   \else%
7978     =\vbox to0.7\baselineskip{\ledrsnotefontsetup\strut#1\vss}}%
7979   \fi%
7980   \endgroup%
7981 }%
7982 \newif\ifleftnoteup
7983 \leftnoteuptrue
7984 %
```

`\@sidenotesep` This macro is used to separate sidenotes of the same line.

```
7985 \newcommand{\setsidenotesep}[1]{\gdef\@sidenotesep{#1}}
7986 \newcommand{\@sidenotesep}{, }
7987 %
```

`\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.

```

7988 \newcommand*{\affixside@note}{%
7989     \prepare@edindex@for@note{\the\page@num|\the\line@num|\the\subline@num|\
the\page@num|\the\line@num|\the\subline@num|}%
7990     \def\sidenotecontent@{}%
7991     \numgdef{\itemcount@}{0}%
7992     \def\do##1{%
7993         \ifnumequal{\itemcount@}{0}%
7994             {%
7995                 \appto\sidenotecontent@{##1}}% Not print not separator before
the 1st note
7996             {\appto\sidenotecontent@{\@sidenotessep ##1}%
7997             }%
7998             \numgdef{\itemcount@}{\itemcount@+\@ne}%
7999     }%
8000     \dolistloop{\l@dcnotes@text}%
8001     \ifnumgreater{\itemcount@}{1}{\led@err@ManySidenotes}{}%
8002 %

```

And we do the same for left and right notes (not movable).

```

8003 \gdef\@templ@ed{}%
8004 \gdef\@templ@n{\l@dcnotes@text\l@dcnotes@text@l\l@dcnotes@text@r}%
8005 \ifx\@templ@ed\@templ@n \else%
8006     \if@twocolumn%
8007         \if@firstcolumn%
8008             \setl@dlp@rbox{##1}{\sidenotecontent@}%
8009         \else%
8010             \setl@drp@rbox{\sidenotecontent@}%
8011         \fi%
8012     \else%
8013         \@l@dttempcntb=\sidenote@margin%
8014         \ifnum\@l@dttempcntb>\@ne%
8015             \advance\@l@dttempcntb by\page@num%
8016         \fi%
8017         \ifodd\@l@dttempcntb%
8018             \setl@drp@rbox{\sidenotecontent@}%
8019             \gdef\sidenotecontent@{}%
8020             \numgdef{\itemcount@}{0}%
8021             \dolistloop{\l@dcnotes@text@l}%
8022             \ifnumgreater{\itemcount@}{1}{\led@err@ManyLeftnotes}{}%
8023             \setl@dlp@rbox{\sidenotecontent@}%
8024         \else%
8025             \setl@dlp@rbox{\sidenotecontent@}%
8026             \gdef\sidenotecontent@{}%
8027             \numgdef{\itemcount@}{0}%
8028             \dolistloop{\l@dcnotes@text@r}%
8029             \ifnumgreater{\itemcount@}{1}{\led@err@ManyRightnotes}{}%
8030             \setl@drp@rbox{\sidenotecontent@}%
8031         \fi%
8032     \fi%
8033 \fi%
8034 \advance\@edindex@for@note@m@ne%
8035 }
8036 %

```


XXVII Minipages and such

We can put footnotes into minipages. The preparatory code has been set up earlier, all that remains is to ensure that it is available inside a minipage box. This requires some alteration to the kernel code, specifically the `\@iiiminipage` and `\endminipage` macros. We will arrange this so that additional series can be easily added.

`\l@dfheetbeginmini` These will be the hooks in `\@iiiminipage` and `\endminipage`.
`\l@dfheetendmini` They can be extended to handle other things if necessary.

```

8037 \ifnoledgroup@else%
8038 \newcommand*{\l@dfheetbeginmini}{\@ledgrouptrue\l@dedbeginmini\l@dfambeginmini}
8039 \newcommand*{\l@dfheetendmini}{%
8040   \IfStrEq{critical-familiar}{\@mpfnpos}%
8041   {\l@dedendmini\l@dfamendmini}%
8042   {%
8043     \IfStrEq{familiar-critical}{\@mpfnpos}%
8044     {\l@dfamendmini\l@dedendmini}%
8045     {\do@feet@custom@order{mp@}{\@mpfnpos}}}%
8046   }%
8047 }%
8048 %

```

`\l@dedbeginmini` These handle the initiation and closure of critical footnotes in a minipage environment.

```

\l@dedendmini
\mp@append@Xnotes
8049 \newcommand*{\l@dedbeginmini}{%
8050   \unless\ifnocritical@%
8051   \def\do##1{%
8052     \csletcs{v##1footnote}{mpv##1footnote}%
8053   }%
8054   \dolistloop{\@series}%
8055   \fi%
8056 }
8057 \newcommand*{\l@dedendmini}{%
8058   \unless\ifnocritical@%
8059   \ifl@dpairing%
8060     \ifledRcol%
8061       \flush@notesR%
8062     \else%
8063       \flush@notes%
8064     \fi%
8065   \fi
8066   \def\do##1{%
8067     \mp@append@Xnotes{##1}%
8068   }%
8069   \dolistloop{\@series}%
8070   \fi%
8071 }%
8072 \newcommand{\mp@append@Xnotes}[1]{%
8073 \ifvoid\csuse{mp#1footins}\else%
8074   \ifl@dpairing%
8075     \ifparledgroup%
8076     \ifledRcol%

```

```

8077 \dimdef{\parledgroup@beforenotesR}{\parledgroup@beforenotesR+\skip
\@nameuse{mp#1footins}}%
8078 \else%
8079 \dimdef{\parledgroup@beforenotesL}{\parledgroup@beforenotesL+\
skip\@nameuse{mp#1footins}}%
8080 \fi%
8081 \fi%
8082 \fi%
8083 \ifcsstring{series@display#1}{paragraph}{}{%
8084 \setbox\@nameuse{mp#1footins}=\vbox{%
8085 \csuse{Xnotefontsize@#1}%
8086 \ifcsdef{Xhsize\csuse{series@display#1}@#1}{%
8087 \hsize\csuse{Xhsize\csuse{series@display#1}@#1}%
8088 }{}%
8089 \noindent\csuse{Xtxtbeforenotes@#1}%
8090 \unvbox\@nameuse{mp#1footins}%
8091 \@parboxrestore%
8092 }%
8093 }%
8094 \csuse{mp#1footgroup}{#1}%
8095 \fi%
8096 }%
8097 %

```

`\l@dfambeginmini` These handle the initiation and closure of familiar footnotes in a minipage environment.

```

\l@dfamendmini
\mp@append@notesX
8098 \newcommand*{\l@dfambeginmini}{%
8099 \unless\ifnofamiliar@%
8100 \def\do##1{\csletcs{vfootnote##1}{mpvfootnote##1}}%
8101 \dolistloop{\@series}%
8102 \fi%
8103 }%
8104
8105 \newcommand*{\l@dfamendmini}{%
8106 \unless\ifnofamiliar@%
8107 \def\do##1{%
8108 \mp@append@notesX{##1}%
8109 }%
8110 \dolistloop{\@series}%
8111 \fi%
8112 }%
8113 \newcommand{\mp@append@notesX}[1]{%
8114 \ifvoid\csuse{mpfootins#1}\else%
8115 \ifcsstring{series@displayX#1}{paragraph}{}{%
8116 \setbox\@nameuse{mpfootins#1}=\vbox{%
8117 \csuse{notefontsizeX@#1}%
8118 \ifcsdef{hsize\csuse{series@display#1}X@#1}{%
8119 \hsize\csuse{hsize\csuse{series@display#1}X@#1}%
8120 }{}%
8121 \noindent\csuse{txtbeforenotesX@#1}%
8122 \unvbox\@nameuse{mpfootins#1}%
8123 \@parboxrestore%
8124 }%
8125 }%

```

```

8126 \csuse{mpfootgroup#1}{#1}%
8127 \fi%
8128 }%
8129 %

```

\@iiminipage This is our extended form of the kernel \@iiminipage defined in ltboxes.dtx.

```

8130 \patchcmd%
8131 {\@iiminipage}%
8132 {\let\@footnotetext\@mpfootnotetext}%
8133 {\let\@footnotetext\@mpfootnotetext\l@dfeetbeginmini}%
8134 {}%
8135 {\led@error@fail@patch@\@iiminipage}%
8136 %

```

\endminipage This is our extended form of the kernel \endminipage defined in ltboxes.dtx.

```

8137 \patchcmd%
8138 {\endminipage}%
8139 {\footnoterule}%
8140 {\footnoterule\l@advance@parledgroup@beforenormalnotes}%
8141 {}%
8142 {\led@error@fail@patch@\endminipage}
8143
8144 \patchcmd%
8145 {\endminipage}%
8146 {\@minipagefalse}%
8147 {\l@dfeetendmini\@minipagefalse}%
8148 {}%
8149 {\led@error@fail@patch@\endminipage}
8150
8151 %

```

\l@dunboxmpfoot \l@dunboxmpfoot insert normal footnotes for ledgroup.
ggroup@beforenormalnotes

```

8152 \newcommand*{\l@dunboxmpfoot}{%
8153 \vskip\skip\@mpfootins
8154 \normalcolor
8155 \footnoterule
8156 \l@advance@parledgroup@beforenormalnotes
8157 \unvbox\@mpfootins%
8158 }
8159 %

```

When using parallel ledgroup, we need to store the vertical space added before footnote, in order to compensate them between left and right pages.

```

8160 \newcommand{\l@advance@parledgroup@beforenormalnotes}{%
8161 \ifparledgroup
8162 \ifl@pairing
8163 \ifledRcol
8164 \dimgdef{\parledgroup@beforenotesR}{\parledgroup@beforenotesR+
skip\@mpfootins}
8165 \else

```

```

8166 \dimgdef{\parledgroup@beforenotesL}{\parledgroup@beforenotesL+\
skip\@mpfootins}
8167 \fi
8168 \fi
8169 \fi
8170 }
8171 %

```

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.

```

8172 \newenvironment{ledgroup}{%
8173 \resetprevpage@num%
8174 \def\@mpfn{mpfootnote}\def\thempfn{\thempfootnote}\c@mpfootnote\z@%
8175 \let\@footnotetext\@mpfootnotetext
8176 \l@dfeetbeginmini%
8177 }{%
8178 \par
8179 \unskip
8180 \ifvoid\@mpfootins\else
8181 \l@dunboxmpfoot
8182 \fi
8183 \l@dfeetendmini%
8184 \@ledgroupfalse%
8185 }
8186 }
8187
8188
8189 %

```

ledgroupsize `\begin{ledgroupsize}[\langle pos \rangle]{\langle width \rangle}`

This environment puts footnotes at the end, even if that happens to be in the middle of a page, or crossing a page boundary. It is a sort of unboxed, variable `\langle width \rangle` minipage. The optional `\langle pos \rangle` controls the sideways position of numbered text.

```

8190 \newenvironment{ledgroupsize}[2][1]{%
8191 %

```

Set the various text measures.

```

8192 \hsize #2\relax
8193 %

```

Initialize fills for centering.

```

8194 \let\ledllfill\hfil
8195 \let\ledrlfill\hfil
8196 \def\@tempa{#1}\def\@tempb{1}%
8197 %

```

Left adjusted numbered lines

```

8198 \ifx\@tempa\@tempb
8199 \let\ledllfill\relax
8200 \else
8201 \def\@tempb{r}%

```

```

8202 \ifx\@tempa\@tempb
8203 %

```

Right adjusted numbered lines

```

8204 \let\ledrlfill\relax
8205 \fi
8206 \fi
8207 %

```

Set up the footnoting.

```

8208 \def\@mpfn{mpfootnote}\def\thempfn{\thempfootnote}\c@mpfootnote\z@
8209 \let\@footnotetext\@mpfootnotetext
8210 \l@dfetbeginmini%
8211 }{
8212 \par
8213 \unskip
8214 \ifvoid\@mpfootins\else
8215 \l@dunboxmpfoot
8216 \fi
8217 \l@dfetendmini%
8218 }
8219 %
8220 %

```

Close the \ifnoledgroup@\else.

```

8221 \fi%
8222 %

```

\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.

```

8223 \newif\ifledgroupnotesL@
8224 \newif\ifledgroupnotesR@
8225 %

```

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.

```

8226 \AtBeginDocument{%
8227 \unless\ifl@imakeidx%
8228 \@ifpackageloaded{imakeidx}{\led@error@PackageAfterEledmac{imakeidx}}{}
8229 %
8230 \unless\ifl@indextools%
8231 \@ifpackageloaded{indextools}{\led@error@PackageAfterEledmac{indextools}}{}%
8232 \fi%

```

```

8233 \unless\ifl@footmisc%
8234 \ifpackageloaded{footmisc}{\led@error@PackageAfterEledmac{footmisc}}{}
8235 %
8236 \fi%
8237 }
8238 %

```

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
8238 \newcommand{\pagelinesep}{-}
8239 \newcommand{\edindexlab}{$&}
8240 \newcounter{labidx}
8241 \setcounter{labidx}{0}
8242
8243 %

```

\doedindexlabel This macro sets an \edlabel.

```

8244 \newcommand{\doedindexlabel}{%
8245 \stepcounter{labidx}%
8246 \edlabel{\edindexlab\thelabidx}%
8247 }
8248
8249 %

```

\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.

```

8250 \newcounter{pageline}%
8251 \renewcommand{\thepageline}{%
8252 \thepage%
8253 \pagelinesep%
8254 \xlineref{\edindexlab\thelabidx}%
8255 }
8256 %

```

\thestartpageline These macros make up the page/line start/end number when the \edindex command is called in critical notes.

```

\thepageline
\theendpageline
8257 \newcommand{\thestartpageline}{%
8258 \l@dparsedstartpage%
8259 \pagelinesep%
8260 \l@dparsedstartline%
8261 }
8262 \newcommand{\theendpageline}{%
8263 \l@dparsedendpage%
8264 \pagelinesep%
8265 \l@dparsedendline%
8266 }
8267 %

```

XXVIII.3 Code specific to \edindex in critical footnotes

`\@edindex@fornote@` This counter is incremented at the beginning of each note (either a footnote or a side-note), and decremented at the end of each note. If its value is greater than 0, that means we are inside a note.

```
8268 \newcount\@edindex@fornote@
8269 %
```

`\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 |.

```
8270 \newcommand{\prepare@edindex@fornote}[1]{%
8271   \l@dp@rsefootspec#1|}%
8272   \advance\@edindex@fornote@\@ne%
8273 }
8274 %
```

`\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.

```
8275 \newcommand{\get@edindex@ledinnote@command}{%
8276   \ifxindy@%
8277     \gdef\@ledinnote@command{%
8278       ledinnote\thelabidx%
8279     }%
8280     \ifxindyhyperref@%
8281       \immediate\write\eledmac@xindy@out{%
8282         (define-attributes ("ledinnote\thelabidx"))^^J
8283         \space\space(markup-locref^^J
8284         \eledmacmarkuplocrefdepth^^J
8285         :open "\string\ledinnote[\edindexlab\thelabidx]{\@index@command
8286         }\@currentseries}{\@currentnotetype}"^^J
8287         :close "}"^^J
8288         :attr "ledinnote\thelabidx"^^J
8289       )
8290     }%
8291   \else%
8292     \immediate\write\eledmac@xindy@out{%
8293       (define-attributes ("ledinnote\thelabidx"))^^J
8294       \space\space(markup-locref^^J
8295       \eledmacmarkuplocrefdepth^^J
8296       :open "\string\ledinnote{\@index@command}{\@currentseries}{\@
8297       @currentnotetype}"^^J
8298       :close "}"^^J
8299       :attr "ledinnote\thelabidx"^^J
8300     )
8301   }%
8302 \fi%
8303 %
```

If we do not use xindy option, `\@ledinnote@command` will produce something like `ledinnote{formattingcommand}`.

```

8302 \else%
8303   \gdef\@ledinnote@command{%
8304     ledinnote[\edindexlab\thelabidx]{\@index@command}{\@currentseries}{\@currentnotetype}%
8305   }%
8306   \fi%
8307 }
8308 %

```

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 |.

```

8309 \def\get@index@command#1|#2+{%
8310   \gdef\@index@txt{#1}%
8311   \gdef\@index@command{#2}%
8312   \xdef\@index@parenthesis{}%
8313   \IfBeginWith{\@index@command}{(}{%
8314     \StrGobbleLeft{\@index@command}{1}{\@index@command@}%
8315     \global\let\@index@command\@index@command@%
8316     \xdef\@index@parenthesis{(%}%
8317   }{%
8318     \IfBeginWith{\@index@command}{)}{%
8319       \StrGobbleLeft{\@index@command}{1}{\@index@command@}%
8320       \global\let\@index@command\@index@command@%
8321       \xdef\@index@parenthesis{)%}%
8322     }{%
8323     }%
8324   }%

```

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.

```

8325 \newcommandx{\ledinnote}[5][1,usedefault]{%
8326   \IfStrEq{#4}{critical}{%
8327     \def\tmp{Xinnotemark}%
8328   }{%
8329     \def\tmp{innotemarkX}%
8330   }%

```



```

8331 \ifboolexpr{%
8332   test{\ifdefequal{\iftrue}{\ifHy@hyperindex}}%
8333   or%
8334   bool {xindyhyperref}%
8335 }%
8336 {%
8337   \csuse{#2}{\hyperlink{#1}{\csuse{\tmp @#3}{#5}}}%
8338 }%
8339 {%
8340   \csuse{#2}{\csuse{\tmp @#3}{#5}}%
8341 }%
8342 }%
8343 \newcommand{\ledinnotemark}[1]{#1\emph{n}}%
8344 %

```

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
8345 \newcommandx{\@wredindex}[2][1=\expandonce\jobname,usedefault]{%#1 = the
index name, #2 = the text
8346   \begingroup%
8347   \let\emph\@firstofone%
8348   \let\textbf\@firstofone%
8349   \let\textit\@firstofone%
8350   \let\textmd\@firstofone%
8351   \let\textnormal\@firstofone%
8352   \let\textrm\@firstofone%
8353   \let\textsc\@firstofone%
8354   \let\textsf\@firstofone%
8355   \let\textsl\@firstofone%
8356   \let\texttt\@firstofone%
8357   \let\textup\@firstofone%
8358   \xdef\@tmp{#2}%To be used in IfSubStr instead of #2 directly. Avoid
some expansion bugs (for example with \edindex{textsc{something}})
8359   \endgroup%
8360   \ifl@imakeidx%
8361     \ifnum\@edindex@fornote@>\z@%
8362       \IfSubStr[1]{\@tmp}{|}{\get@index@command#2+}{\get@index@command#2|+}
%
8363       \get@edindex@ledinnote@command%
8364       \expandafter\imki@wrindexentry{#1}{\@index@txt|(\@ledinnote@command
}{\thestartpageline}%
8365       \expandafter\imki@wrindexentry{#1}{\@index@txt|)\@ledinnote@command
}{\theendpageline}%

```

```

8366 \else%
8367 \get@edindex@hyperref{#2}%
8368 \imki@wrindexentry{#1}{\@index@txt\@edindex@hyperref}{\thepage}%
8369 \fi%
8370 \else%
8371 \ifnum\@edindex@fornote@>\z@%
8372 \IfSubStr[1]{\@tmp}{|}{\get@index@command#2+}{\get@index@command#2|+}%
8373 %
8374 \get@edindex@ledinnote@command%
8375 \expandafter\protected@write\@indexfile{}%
8376 {\string\indexentry{\@index@txt|(\@ledinnote@command){\thestartpageline}%
8377 }%
8378 \expandafter\protected@write\@indexfile{}%
8379 {\string\indexentry{\@index@txt|)\@ledinnote@command}{\theendpageline}%
8380 }%
8381 \else%
8382 \protected@write\@indexfile{}%
8383 {\string\indexentry{#2}{\thepage}%
8384 }%
8385 \fi%
8386 \endgroup
8387 \@esphack%
8388 }
8389 %

```

Need to add the definition of `\edindex` to `\makeindex`, and initialise `\edindex` to do nothing.

```

8390 \pretocmd{\makeindex}{%
8391 \def\edindex{%
8392 \ifboolexpr{bool{numbering} or bool{numberingR} or bool{
8393 l@printingpages} or bool{l@dprintingcolumns}}{%
8394 \@bsphack%
8395 \doedindexlabel%
8396 \begingroup%
8397 \@sanitize%
8398 \wredindex%
8399 }%
8400 {%
8401 \led@warn@edinde@outsidenumbering%
8402 \index%
8403 }%
8404 }%
8405 }%
8406 {\led@error@fail@patch@makeindex}%
8407 \newcommand{\edindex}[1]{\@bsphack\@esphack}
8408 \newcommandx{\dummy@edindex}[2][1=\expandonce\jobname,usedefault]{}%
8409 \newcommand{\disable@edindex}{\let\old@edindex\edindex\let\edindex\
8410 dummy@edindex}%
8411 \newcommand{\restore@edindex}{\let\edindex\old@edindex}%
8412 %

```

XXVIII.7 Hyperlink

`\hyperlinkformat` `\hyperlinkformat` command is to be used to have both a internal hyperlink and a format, when indexing.

```

8412 \newcommand{\hyperlinkformat}[3]{%
8413   \ifstrempy{#1}%
8414     {\hyperlink{#2}{#3}}%
8415     {\csuse{#1}{\hyperlink{#2}{#3}}}%
8416   }%
8417 %

```

`\hyperlinkR` `\hyperlinkR` command is to be used to create a internal hyperlink and `\ledRflag`, when indexing.

```

8418 \newcommand{\hyperlinkR}[2]{%
8419   \hyperlink{#1}{#2\@Rlineflag}%
8420 }%
8421 %
8422 %

```

`\hyperlinkformatR` `\hyperlinkformatR` command is to be used to create a internal hyperlink, a format and a `\@Rlineflag`, when indexing.

```

8423 \newcommand{\hyperlinkformatR}[3]{%
8424   \hyperlinkformat{#1}{#2}{#3\@Rlineflag}%
8425 }%
8426 %
8427 %

```

`\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`).

```

8428 \newcommand{\get@edindex@hyperref}[1]{%
8429 %

```

We have to disable temporary spaces to work through a `xstring` bug (or feature?)

```

8430   \edef\temp@{%
8431     \catcode`\ =9 %space need for catcode
8432     \detokenize{#1}%For active character in unicode
8433     \catcode`\ =10 % space need for catcode
8434   }%
8435 %

```

Now, we define `\@edindex@hyperref` if the `hyperindex` of `hyperref` is enabled.

```

8436   \ifdefequal{\iftrue}{\ifHy@hyperindex}{%
8437     \IfSubStr{\temp@}{|}%
8438     {\get@index@command#1+%
8439     \ifledRcol%
8440       \gdef\@edindex@hyperref{|\@index@parenthesis %space kept
8441       hyperlinkformatR{\@index@command}%
8442       {\edindexlab\thelabidx}}%
8443     \else%
8444       \gdef\@edindex@hyperref{|\@index@parenthesis %space kept

```

```

8445     hyperlinkformat{\@index@command}%
8446     {\edindexlab\thelabidx}}%
8447   \fi%
8448 }%
8449 {\get@index@command#1|+%
8450   \ifledRcol%
8451     \gdef\@edindex@hyperref{|hyperlinkR{\edindexlab\thelabidx}}%
8452   \else%
8453     \gdef\@edindex@hyperref{|hyperlink{\edindexlab\thelabidx}}%
8454   \fi%
8455 }%
8456 }%
8457 %

8458 % If we use both xindy and hyperref, first get the \protect\cs{
index@command} command.
8459 % Then define \protect\cs{@edindex@hyperref} in the form \verb+eledmacXXX+
8460 % \begin{macrocode}
8461 {\ifxindyhyperref%
8462   \IfSubStr{\temp@}{|}%
8463     {\get@index@command#1+%
8464     {\get@index@command#1|+%
8465     \gdef\@edindex@hyperref{|eledmac\thelabidx}%
8466 %

```

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`.

```

8467   \IfStrEq{\@index@parenthesis}{(}%
8468   {%
8469     \csxdef{xindyparenthesis@\@index@txt}{\thelabidx}%
8470     \gdef\@edindex@hyperref{| (eledmac\thelabidx}%
8471   }%
8472   }%
8473 %

```

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`.

```

8474   \IfStrEq{\@index@parenthesis}{)}%
8475   {%
8476     \xdef\@edindex@hyperref{|)eledmac\csuse{xindyparenthesis@\@index@txt}}%
8477     \global\csundef{xindyparenthesis@\@index@txt}%
8478   }%
8479 %

```

Write in the `.xdy` file the attributes of the location.

```

8480   {%
8481     \immediate\write\eledmac@xindy@out{%
8482       (define-attributes ("eledmac\thelabidx"))^^J
8483       \space\space(markup-locref^^J
8484       \eledmacmarkuplocrefdepth^^J

```

```

8485         :open "\string\hyperlink%
8486             \ifledRcol R\fi%
8487             {\edindexlab\thelabidx}%
8488             {\ifdefempty{\@index@command}%
8489             {}%
8490             {\@backslashchar\@index@command}%
8491             {"^^J
8492         :close "}}"^^J
8493         :attr "eledmac\thelabidx"^^J
8494     )
8495 }%
8496 }%
8497 %

```

And now, in any other case.

```

8498 \else%
8499     \gdef\@index@txt{#1}%
8500     \gdef\@edindex@hyperref{}%
8501     \fi%
8502 }%
8503 }
8504 %

```

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.

```

8505 \newcommand{\led@set@index@fornote}[1]{%
8506     \ifbool{indtl@innote}%
8507     {\let\index\nindex}%
8508     {}%
8509     \ifbool{indtl@notenumber}%
8510     {%
8511         \renewcommand{\index}[2][\indtl@jobname]{%
8512             \orig@@index[#1]{%
8513                 ##2|innotenumber{\this@footnoteX@reading}%
8514             }%
8515         }%
8516     }%
8517     {}%
8518 }%
8519 %

```

\led@reinit@index@fornote The `\led@reinit@index@fornote` just reset the default value of `\index`.

```

8520 \newcommand{\led@reinit@index@fornote}{%
8521     \ifbool{indtl@innote}%
8522     {\let\index\orig@@index}%
8523     {}%
8524     \ifbool{indtl@notenumber}%

```

```

8525 {\let\index\orig@@index}%
8526 {}%
8527 }%
8528 %

```

XXIX Glossaries

Here, we define the `\gls`-like commands prefixed by `ed`, only if the package `glossaries` is loaded.

```

8529 \AtBeginDocument{%
8530   \@ifpackageloaded{glossaries}{%
8531     \@ifpackageloaded{glossaries-extra}{\
8532       \error@require{glossaries-extra}%
8533       \newif\ifl@noexpands@edgl%A boolean switched to true in \no@expands
8534     }%

```

First those which arguments are [*options*]{*label*}[*insert*].

```

8534   \gdef\edglsomo{%
8535     \listgadd{\edglsomo}{gls}%
8536     \listgadd{\edglsomo}{Gls}%
8537     \listgadd{\edglsomo}{GLS}%
8538     \listgadd{\edglsomo}{glspl}%
8539     \listgadd{\edglsomo}{Glspl}%
8540     \listgadd{\edglsomo}{GLSpl}%
8541     \listgadd{\edglsomo}{glstext}%
8542     \listgadd{\edglsomo}{Glstext}%
8543     \listgadd{\edglsomo}{GLStext}%
8544     \listgadd{\edglsomo}{Glsfirst}%
8545     \listgadd{\edglsomo}{GLSfirst}%
8546     \listgadd{\edglsomo}{glsplural}%
8547     \listgadd{\edglsomo}{Glsplural}%
8548     \listgadd{\edglsomo}{GLSplural}%
8549     \listgadd{\edglsomo}{glsfirstplural}%
8550     \listgadd{\edglsomo}{Glsfirstplural}%
8551     \listgadd{\edglsomo}{GLSfirstplural}%
8552     \listgadd{\edglsomo}{glsname}%
8553     \listgadd{\edglsomo}{Glsname}%
8554     \listgadd{\edglsomo}{GLSname}%
8555     \listgadd{\edglsomo}{glssymbol}%
8556     \listgadd{\edglsomo}{Glsymbol}%
8557     \listgadd{\edglsomo}{GLSsymbol}%
8558     \listgadd{\edglsomo}{glsdesc}%
8559     \listgadd{\edglsomo}{Glsdesc}%
8560     \listgadd{\edglsomo}{GLSdesc}%
8561     \listgadd{\edglsomo}{glsuseri}%
8562     \listgadd{\edglsomo}{Glsuseri}%
8563     \listgadd{\edglsomo}{GLSuseri}%
8564     \listgadd{\edglsomo}{glsuserii}%
8565     \listgadd{\edglsomo}{Glsuserii}%
8566     \listgadd{\edglsomo}{GLSuserii}%
8567     \listgadd{\edglsomo}{glsuseriii}%
8568     \listgadd{\edglsomo}{Glsuseriii}%

```

```

8569 \listgadd{\edglsomo}{GLSuseriii}%
8570 \listgadd{\edglsomo}{glsuseriv}%
8571 \listgadd{\edglsomo}{Glsuseriv}%
8572 \listgadd{\edglsomo}{GLSuseriv}%
8573 \listgadd{\edglsomo}{glsuserv}%
8574 \listgadd{\edglsomo}{Glsuserv}%
8575 \listgadd{\edglsomo}{GLSuserv}%
8576 \listgadd{\edglsomo}{glsuservi}%
8577 \listgadd{\edglsomo}{Glsuservi}%
8578 \listgadd{\edglsomo}{GLSuservi}%
8579 \renewcommand{\do}[1]{%
8580 \expandafter\DeclareRobustCommandx\csname ed#1\endcsname[3][1,3,
usedefault]{%
8581 \doedindexlabel%
8582 \ifl@noexpands@edgl%
8583 \csname#1\endcsname[counter=pageline,noindex,##1]{##2}{##3}%
8584 \else%
8585 \ifdef{\hypertarget}{%
8586 \Hy@raisedlink@left{\hypertarget{pageline.\thepageline}{}}%
8587 }{}%
8588 \csname#1\endcsname[counter=pageline,##1]{##2}{##3}%
8589 \fi%
8590 }%
8591 \expandafter\WithSuffix\expandafter\DeclareRobustCommandx\csname ed
#1\endcsname*[3][1,3,usedefault]{%
8592 \doedindexlabel%
8593 \ifl@noexpands@edgl%
8594 \csname#1\endcsname*[counter=pageline,noindex,##1]{##2}{##3}%
8595 \else%
8596 \csname#1\endcsname*[counter=pageline,##1]{##2}{##3}%
8597 \ifdef{\hypertarget}{%
8598 \Hy@raisedlink@left{\hypertarget{pageline.\thepageline}{}}%
8599 }{}%
8600 \fi%
8601 }%
8602 }%
8603 \dolistloop{\edglsomo}%
8604 %

```

Then those which arguments are [*options*]{*label*}{*link text*}.

```

8605 \gdef\edglsomm{}%
8606 \listgadd\edglsomm{glsdisp}%
8607 \listgadd\edglsomm{glslink}%
8608 \renewcommand{\do}[1]{%
8609 \expandafter\DeclareRobustCommandx\csname ed#1\endcsname[3][1,
usedefault]{%
8610 \doedindexlabel%
8611 \ifl@noexpands@edgl%
8612 \csname#1\endcsname[counter=pageline,noindex,##1]{##2}{##3}%
8613 \else%
8614 \ifdef{\hypertarget}{%
8615 \Hy@raisedlink@left{\hypertarget{pageline.\thepageline}{}}%
8616 }{}%
8617 \csname#1\endcsname[counter=pageline,##1]{##2}{##3}%

```

```

8618     \fi%
8619   }%
8620   \expandafter\WithSuffix\expandafter\DeclareRobustCommand\csname ed
#1\endcsname*[3][1,usedefault]{%
8621     \doedindexlabel%
8622     \ifl@noexpands@edgl%
8623       \csname#1\endcsname*[counter=pageline,noindex,##1]{##2}{##3}%
8624     \else%
8625       \ifdef{\hypertarget}{%
8626         \Hy@raisedlink@left{\hypertarget{pageline.\thepageline}{}}%
8627       }{}%
8628       \csname#1\endcsname*[counter=pageline,##1]{##2}{##3}%
8629     \fi%
8630   }%
8631 }%
8632 \dolistloop{\edglsomm}%
8633 %

```

Then those which arguments are $[\langle options \rangle] \{ \langle label \rangle \}$.

```

8634 \gdef\edglsom{}%
8635 \listgadd\edglsom{glsadd}%
8636 \renewcommand{\do}[1]{%
8637   \expandafter\DeclareRobustCommand\csname ed#1\endcsname[2][1,
usedefault]{%
8638     \doedindexlabel%
8639     \ifl@noexpands@edgl%
8640       \csname#1\endcsname[counter=pageline,noindex,##1]{##2}%
8641     \else%
8642       \ifdef{\hypertarget}{%
8643         \Hy@raisedlink@left{\hypertarget{pageline.\thepageline}{}}%
8644       }{}%
8645       \csname#1\endcsname[counter=pageline,##1]{##2}%
8646     \fi%
8647   }%
8648   \expandafter\WithSuffix\expandafter\DeclareRobustCommand\csname ed
#1\endcsname*[2][1,usedefault]{%
8649     \doedindexlabel%
8650     \ifl@noexpands@edgl%
8651       \csname#1\endcsname*[counter=pageline,noindex,##1]{##2}%
8652     \else%
8653       \ifdef{\hypertarget}{%
8654         \Hy@raisedlink@left{\hypertarget{pageline.\thepageline}{}}%
8655       }{}%
8656       \csname#1\endcsname*[counter=pageline,##1]{##2}%
8657     \fi%
8658   }%
8659 }%
8660 \dolistloop{\edglsom}%
8661 %

```

`\l@noexpands@edgl` is a macro to switch `\ifl@noexpands@edgl` to true in `\no@expands`. Its definition is made dynamically.

```

\l@noexpands@edgl462 \def\l@noexpands@edgl{\l@noexpands@edgltrue}%

```



```

8663 }{\let\l@noexpands@edgl\relax}%
8664 }%
8665 %

```

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.

```

8666 \def\@hangingsymbol{}
8667 \newcommand*{\sethangingsymbol}[1]{%
8668   \gdef\@hangingsymbol{#1}%
8669 }%
8670 \newif\ifinstanza
8671 %

```

`\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.

```

8672 \newif\ifinserthangingsymbol
8673 \newcommand{\inserthangingsymbol}{%
8674   \ifinserthangingsymbol%
8675     \ifinstanza%
8676       \@hangingsymbol%
8677     \fi%
8678   \fi%
8679 }
8680 %

```

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.

```

8681 \newcommand*{\ampersand}{\char`\&}
8682
8683 %

```

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.

```
8684 \chardef\body=\catcode`\@
8685 \catcode`\@=11
8686 \chardef\next=\catcode`\&
8687 \catcode`\&=\active
8688
8689 %
```

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.

```
8690 \newcount\stanza@count
8691 \newlength{\stanzaindentbase}
8692 \setlength{\stanzaindentbase}{20pt}
8693
8694 %
```

`\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.

```
8695 \def\strip@szacnt#1,#2|{\def\@tempb{#1}\def\@tempa{#2|}}
8696 \newcommand*{\setstanzavalues}[2]{\def\@tempa{#2,,|}%
8697   \stanza@count\z@
8698   \def\next{\expandafter\strip@szacnt\@tempa
8699     \ifx\@tempb\empty\let\next\relax\else
8700     \expandafter\mathchardef\csname #1@number\stanza@count
8701     \@endcsname\@tempb\relax
8702     \advance\stanza@count\@ne\fi\next}%
8703   \next}
8704
8705 %
```

`\setstanzaindents` In the original EDMAC, `\setstanzavalues{sza}{\langle...⟩}` had to be called to set the in-
`\setstanzapenalties` dents, and similarly `\setstanzavalues{szp}{\langle...⟩}` to set the penalties. `\setstanzaindents` and `\setstanzapenalties` macros are a convenience to give the user one less thing to worry about (misspelling the first argument).

```
8706 \newcommand*{\setstanzaindents}[1]{\setstanzavalues{sza}{#1}}
8707 \newcommand*{\setstanzapenalties}[1]{\setstanzavalues{szp}{#1}}
8708 %
8709 %
```

`\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.

```

8710 \newcounter{stanzaindentsrepetition}
8711 \newcounter{startstanzaindentsrepetition}%
8712 \setcounter{startstanzaindentsrepetition}{1}%
8713 \newcount\stanza@modulo
8714
8715 \newcommand*{\managestanza@modulo}[0]{%
8716   \advance\stanza@modulo\@ne%
8717   \ifnum\stanza@modulo>\value{stanzaindentsrepetition}%
8718     \stanza@modulo\@ne%
8719   \fi%
8720 }
8721 %

```

`\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.

```

8722 \newcommand{\stanzaindent}[1]{%
8723   \hspace{\dimexpr#1\stanzaindentbase-\parindent\relax}%
8724   \ignorespaces%
8725 }%
8726 \WithSuffix\newcommand\stanzaindent*[1]{%
8727   \stanzaindent{#1}%
8728   \global\advance\stanza@modulo-\@ne%
8729   \ifnum\stanza@modulo=0%
8730     \global\stanza@modulo=\value{stanzaindentsrepetition}%
8731   \fi%
8732   \ignorespaces%
8733 }%
8734 %

```

XXX.5 Numbering stanza

Here, macro for numbering stanza. First, the stanza counter.

```

\thestanza 8735 \newcounter{stanza}
8736 \renewcommand{\thestanza}{%
8737   \textbf{\arabic{stanza}}%
8738 }
8739 %

```

`\ifnumberstanza` Then, macro to activate automatically numbering of stanza.

```

8740 \newif\ifnumberstanza%
8741 %

```

`\@insertstanza` Now, macro called at the first line of verse of a stanza.

```

8742 \newcommand{\@insertstanza}[0]{%
8743   \ifnumberstanza%
8744   \ifl@dpairing%
8745   \ifledRcol%
8746   \stanzanumwrapper{\thestanzaR}%
8747   \else%
8748   \stanzanumwrapper{\thestanzaL}%
8749   \fi%
8750   \else%
8751   \stanzanumwrapper{\thestanza}%
8752   \fi%
8753   \setline{1}%
8754   \fi%
8755 }%
8756 %

```

`\@advancestanza` Also a command to advance the counter of stanza.

```

8757 \newcommand{\@advancestanza}[0]{%
8758   \ifnumberstanza%
8759   \ifl@dpairing%
8760   \ifledRcol%
8761   \addtocounter{stanzaR}{1}%
8762   \else%
8763   \addtocounter{stanzaL}{1}%
8764   \fi%
8765   \else%
8766   \addtocounter{stanza}{1}%
8767   \fi%
8768   \fi%
8769 }%
8770 %

```

`\stanzanumwrapper` And finally, the wrapper for stanza number

```

8771 \newcommand{\stanzanumwrapper}[1]{%
8772   \flagstanza{#1}%
8773 }%
8774 %

```

XXX.6 Stanza number in note

Here, the command called when printing stanza number in notes.

```

8775 \newcommand{\printstanza}[0]{%
8776   \ifboolexpr{bool{l@dpairing} or bool{l@dprintingpages} or bool{
l@dprintingcolumns}}{%
8777     \ifledRcol%
8778     \thestanzaR%
8779     \else%
8780     \thestanzaL%
8781     \fi%

```

```

8782 }{%
8783   \thestanza%
8784 }%
8785 }
8786 %

```

XXX.7 Main work

`\stanza@line` Now we arrive at the main works. `\stanza@line` sets the indentation for the line and starts a numbered paragraph—each line is treated as a paragraph. `\stanza@hang` sets the hanging indentation to be used if the stanza line requires more than one print line.

`\stanza@indent` If it is known that each stanza line will fit on one print line, it is advisable to set the hanging indentation to zero. `\sza@penalty` places the specified penalty following each stanza line. By default, this facility is turned off so that no penalty is included. However, the user may initiate these penalties to indicate good and bad places in the stanza for page breaking.

`\sza@penalty`

```

8787 \newcommandx{\stanza@line}[2][1,2,usedefault]{%
8788   \stanza@indent%
8789   \pstart[#1][#2]\stanza@hang\ignorespaces%
8790 }%
8791 \newcommand{\stanza@indent}{%
8792   \ifboolexpr{%
8793     test {\ifnumequal{\value{stanzaindentsrepetition}}{0}}%
8794     or%
8795     test {\ifnumless{\number\stanza@count}{\value{
startstanzaindentsrepetition}}}%
8796   }{%
8797     \ifcsdef{sza@\number\stanza@count @}{%
8798       {%
8799         \parindent=\csname sza@\number\stanza@count @\endcsname\
stanzaindentbase%
8800       }{%
8801         \led@err@StanzaIndentNotDefined%
8802       }%
8803     }{%
8804       \ifcsdef{sza@\number\stanza@modulo @}{%
8805         \parindent=\csname sza@%
8806         \the\numexpr%
8807         \stanza@modulo%
8808         - 1%
8809         + \value{startstanzaindentsrepetition}%
8810         \relax%
8811         @\endcsname\stanzaindentbase%
8812         \managestanza@modulo%
8813       }%
8814       {%
8815         \led@err@StanzaIndentNotDefined%
8816       }%
8817     }%
8818   }%
8819
8820 \def\stanza@hang{%

```

```

8821 \leavevmode\startlock%
8822 \hangindent\if@RTL-\fi\expandafter%
8823 \csname sza@0@\endcsname\stanzaindentbase%
8824 \hangafter\@ne%
8825 }%
8826
8827 \def\sza@penalty{\count@\csname szp@\number\stanza@count @\endcsname
8828 \ifnum\count@>\@M\advance\count@-\@M\penalty-\else
8829 \penalty\fi\count@}
8830 %

```

\@startstanza Now we have the components of the \stanza macro, which appears at the start of a
 \stanza group of lines. This macro initializes the count and checks to see if hanging indentation
 \@stopstanza and penalties are to be included. Hanging indentation suspends the line count, so that
 \AtEveryStopStanza the enumeration is by verse line rather than by print line. If the print line count is desired,
 \AtEveryStanza invoke \let\startlock\relax and do the same for \endlock. Here and above we
 \AtStartEveryStanza have used \xdef to make the stored macros take up a bit less space, but it also makes
 \BeforeEveryStopStanza them more obscure to the reader. Lines of the stanza are delimited by ampersands &.
 \newverse The last line of the stanza must end with \&.

```

8831 \xdef\@startstanza[#1][#2]{%
8832 \noexpand\instanzatrue\expandafter
8833 \begingroup%
8834 \catcode`\noexpand\&\active%
8835 \unexpanded{\newdimen\parindent@beforestanza%
8836 \parindent@beforestanza=\parindent}%Keep in memory the standard \
parindent
8837 \global\stanza@count\@ne\stanza@modulo\@ne
8838 \noexpand\ifnum\expandafter\noexpand\csname sza@0@\endcsname=\z@%
8839 \def\noexpand\stanza@hang{\unexpanded{\leavevmode\null}}%
8840 \let\noexpand\endlock\relax%
8841 \noexpand\else%
8842 \interlinepenalty\@M%
8843 \unexpanded{%
8844 \ifboolexpr{bool {RTL} and bool{xetex}}{%
8845 \leftskip\z@ plus 1fil\relax%
8846 }{%
8847 \rightskip\z@ plus 1fil\relax%
8848 }%
8849 }%
8850 \noexpand\fi%
8851 \noexpand\ifnum\expandafter\noexpand\csname szp@0@\endcsname=\z@%
8852 \let\noexpand\sza@penalty%
8853 \relax\noexpand\fi%
8854 \def\noexpand&{%
8855 \noexpand\newverse[] []}%
8856 \def\noexpand\&{\noexpand\@stopstanza}%
8857 \noexpand\@advancestanza%
8858 \noexpand\stanza@line[#1][#2]%
8859 \noexpand\@insertstanza%
8860 \let\par\relax\ignorespaces%No paragraph in verses
8861 }
8862
8863 \newcommandx{\stanza}[2][1,2,usedefault]{%

```

```

8864 \ifboolexpr{%
8865   not test{\ifdefvoid{\at@every@stanza}}%
8866   and test{\ifstrempy{#1}}%
8867   and test{\ifstrempy{#2}}}%
8868   {\@startstanza[] [\at@every@stanza]\at@start@every@stanza}%
8869   {\@startstanza[#1] [#2]\at@start@every@stanza}%
8870 }%
8871
8872 \newcommandx{\@stopstanza}[2][1,2,usedefault]{%
8873   \unskip%
8874   \endlock%
8875   \ifboolexpr{%
8876     not test{\ifdefvoid{\at@every@stop@stanza}}%
8877     and test{\ifstrempy{#1}}%
8878     and test{\ifstrempy{#2}}}%
8879     {\before@every@stop@stanza\pend[] [\at@every@stop@stanza]}%
8880     {\before@every@stop@stanza\pend[#1] [#2]}%
8881   \endgroup%
8882   \instanzafalse%
8883   \global\undef\parindent@beforestanza%
8884 }
8885
8886 \newcommand{\AtEveryStopStanza}[1]{%
8887   \ifstrempy{#1}%
8888   {\gdef\at@every@stop@stanza{}}%
8889   {\gdef\at@every@stop@stanza{\noindent#1}}%
8890 }%
8891 \WithSuffix\newcommand\AtEveryStopStanza*[1]{%
8892   \ifstrempy{#1}%
8893   {\gdef\at@every@stop@stanza{}}%
8894   {\gdef\at@every@stop@stanza{#1}}%
8895 }%
8896 \def\at@every@stop@stanza{}%
8897
8898 \newcommand{\AtEveryStanza}[1]{%
8899   \ifstrempy{#1}%
8900   {\gdef\at@every@stanza{}}%
8901   {\gdef\at@every@stanza{\noindent#1}}%
8902 }%
8903 \WithSuffix\newcommand\AtEveryStanza*[1]{%
8904   \ifstrempy{#1}%
8905   {\gdef\at@every@stanza{}}%
8906   {\gdef\at@every@stanza{#1}}%
8907 }%
8908
8909
8910
8911 \newcommand{\AtStartEveryStanza}[1]{%
8912   \ifstrempy{#1}%
8913   {\gdef\at@start@every@stanza{}}%
8914   {\gdef\at@start@every@stanza{#1}}%
8915 }%
8916 \def\at@start@every@stanza{}%
8917

```

```

8918 \newcommand{\BeforeEveryStopStanza}[1]{%
8919   \ifstrempy{#1}%
8920     {\gdef\before@every@stop@stanza{}}%
8921     {\gdef\before@every@stop@stanza{#1}}%
8922 }%
8923 \def\before@every@stop@stanza{%
8924
8925 \newcommandx*\newverse}[4][1,2,3,4,usedefault]{%
8926   \unskip%
8927   \endlock\pend[#1][#3]\sza@penalty\global%
8928   \advance\stanza@count\one\stanza@line[#2][#4]%
8929 }
8930
8931 %

```

\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`.

```

8932 \newcommand*\flagstanza[2][\stanzaindentbase]{%
8933   \hskip -#1\llap{#2}\hskip #1\ignorespaces}
8934
8935 %

```

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.

```

8936 \catcode`\&=\next
8937 \catcode`\@=\body
8938 \setstanzavalues{szp}{0}
8939
8940 %

```

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
8941 \ifantilabe@%
8942   \newdimen\firsthalf@antilabe@wd%
8943   \newcommand\get@firsthalf@antilabe@wd{%
8944     \ifboolexpr{bool {instanza} or bool {inastanzaL} or bool {inastanzaR}}{
%
8945       \vbadness=10000%
8946       \splittopskip=\z@%
8947       \ifl@dpairing%
8948         \ifledRcol%
8949         \setbox0=\copy\csname l@dRcolrawbox\the\l@dnumstartsR\endcsname%

```



```

8950     \else%
8951         \setbox0=\copy\csname l@dLcolrawbox\the\l@dnumstartsL\endcsname%
8952     \fi%
8953     \else%
8954         \setbox0=\copy\raw@text%
8955     \fi%
8956     \dimen0=\dimexpr\ht0+\dp0-\baselineskip\relax%
8957     \ifdim\dimen0>0pt%
8958         \setbox1=\vsplit0 to \dimen0%
8959     \fi%
8960     \setbox0=\vbox{\unvbox0\global\setbox1=\lastbox}%
8961     \setbox1=\hbox{\noindent\unhbox1}%
8962     \global\firsthalf@antilabe@wd=\wd1%
8963     }{}%
8964 }%
8965 \def\beforeantilabe{}%
8966 \def\afterantilabe{}%
8967 \newcommand{\antilabe}[1]{%
8968     \beforeantilabe\hbox to\firsthalf@antilabe@wd{#1\hfill}\afterantilabe%
8969 }%
8970 \else%
8971     \let\get@firsthalf@antilabe@wd\relax%
8972 \fi%
8973 %

```

XXXI Apparatus of Manuscripts

XXXI.1 User level macro

\msdata The user level **\msdata** command only writes the manuscripts data in a numbered auxiliary file. There are two associated etoolbox counters.

```

\msdata@cR
8974 \def\msdata@c{}%
8975 \def\msdata@cR{}%
8976 \newcommand{\msdata}[1]{%
8977     \leavevmode%
8978     \unless\ifstopmsdata@inserted%
8979         \stopmsdata%
8980         \led@warning@msdatawithoutstop%
8981     \fi%
8982     \global\stopmsdata@inserted@false%
8983     \unless\ifledRcol%
8984         \numgdef{\msdata@c}{\msdata@c+1}%
8985         \ifdef{\hypertarget}{%
8986             \edlabel{\msdata@c:start:msdata}%
8987         }{}%
8988         \protected@write\linenum@out{}{%
8989             \string\@msd{#1}%
8990         }%
8991     \else%
8992         \numgdef{\msdata@cR}{\msdata@cR+1}%
8993         \ifdef{\hypertarget}{%
8994             \edlabel{\msdata@cR:start:msdata}%

```

```

8995 }{}%
8996 \protected@write\linenum@outR{}{%
8997   \string\@msd{#1}%
8998 }%
8999 \fi%
9000 }%
9001 %

```

\stopmsdata The user level `\stopmsdata` command only writes information about the end of manuscripts data in numbered auxiliary file.

```

9002 \newcommand{\stopmsdata}[0]{%
9003   \leavevmode%
9004   \unless\ifledRcol%
9005     \protected@write\linenum@out{}{%
9006       \string\@stopmsd%
9007     }%
9008     \ifdef{\hypertarget}{%
9009       \edlabel{\msdata@c:end:msdata}%
9010     }{}%
9011   \else%
9012     \protected@write\linenum@outR{}{%
9013       \string\@stopmsd%
9014     }%
9015     \ifdef{\hypertarget}{%
9016       \edlabel{\msdata@cR:end:msdata}%
9017     }{}%
9018   \fi%
9019   \global\stopmsdata@inserted@true%
9020 }%
9021 %

```

\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`

```

9022 \newif\ifstopmsdata@inserted@%
9023 %

```

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?

```

9024 \newcommand{\setmsdataseries}[1]{%
9025   \gdef\@msdata@series{#1}%
9026 }%
9027 \def\@msdata@series{A}%
9028 %

```

\setmsdataposition The label for the manuscripts data.

```

9029 \def\ms@data@position{msdata-regular}%
9030 \newcommand{\setmsdataposition}[1]{%
9031   \gdef\ms@data@position{#1}%
9032 }%
9033 %

```

\setmsdatalabel The label for the manuscripts data.

```

9034 \def\ms@data@label{Ms.}%
9035 \newcommand{\setmsdatalabel}[1]{%
9036   \gdef\ms@data@label{#1}%
9037 }%
9038 %

```

XXXI.3 Counters and lists

\@msd@c \@msd@c is a counter incremented at each \@msd read in auxiliary file.

```

9039 \numdef{\@msd@c}{0}
9040 \numdef{\@msd@cR}{0}
9041 %

```

\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.

```

9042 \numdef{\add@msd@c}{0}%
9043 \numdef{\add@msd@cR}{0}%
9044 %

```

\@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.

```

9045 \def\@msdata@list{}%
9046 %

```

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.

```

9047 \newcommand{\@msd}[1]{%
9048   \unless\ifledRcol%
9049     \global\numdef{\@msd@c}{\@msd@c+\@ne}%
9050     \csgdef{\@msdata@\@msd@c @data}{#1}%
9051     \csxdef{\@msdata@\@msd@c @linenumber}{\the\line@num}%
9052     \csxdef{\@msdata@\@msd@c @sublinenumber}{\the\subline@num}%
9053     \csxdef{\@msdata@\@msd@c @abslinenumber}{\the\absline@num}%
9054     \xright@appenditem{\the\absline@num}\to\actionlines@list%
9055     \xright@appenditem{-1010}\to\actions@list%
9056   \else%

```

```

9057 \global\newcommand{\@msd@cR}{\@msd@cR+\@one}%
9058 \csgdef{\@msd@data@\@msd@cR @dataR}{\#1}%
9059 \csxdef{\@msd@data@\@msd@cR @linenumberR}{\the\line@numR}%
9060 \csxdef{\@msd@data@\@msd@cR @sublinenumberR}{\the\subline@numR}%
9061 \csxdef{\@msd@data@\@msd@cR @abslinenumberR}{\the\absline@numR}%
9062 \xright@appenditem{\the\absline@numR}\to\actionlines@listR%
9063 \xright@appenditem{-1010}\to\actions@listR%
9064 \fi%
9065 }%
9066 %

```

\@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.

```

9067 \newcommand{\@stopmsd}[0]{%
9068   \unless\ifledRcol%
9069     \ifcsundef{\@msd@data@\@msd@c @stoplinenumberR}{%
9070       \csxdef{\@msd@data@\@msd@c @stopabslinenumberR}{\the\absline@numR}%
9071       \csxdef{\@msd@data@\@msd@c @stoplinenumberR}{\the\line@numR}%
9072       \csxdef{\@msd@data@\@msd@c @stopsublinenumberR}{\the\subline@numR}%
9073     }{}%
9074   \else%
9075     \ifcsundef{\@msd@data@\@msd@cR @stoplinenumberR}{%
9076       \csxdef{\@msd@data@\@msd@cR @stopabslinenumberR}{\the\absline@numR}%
9077       \csxdef{\@msd@data@\@msd@cR @stoplinenumberR}{\the\line@numR}%
9078       \csxdef{\@msd@data@\@msd@cR @stopsublinenumberR}{\the\subline@numR}%
9079     }%
9080   }{}%
9081 \fi%
9082 }%
9083 %

```

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 \@msd@list.

```

9084 \newcommand{\add@msdata}{%
9085   \bgroup%
9086   \normalfont%
9087   \unless\ifledRcol%
9088     \numgdef{\add@msd@c}{\add@msd@c+\@one}%
9089     \ifcsdef{\@msd@data@\add@msd@c @data}{%
9090       \letcs{\@data}{\@msd@data@\add@msd@c @data}%
9091     }%
9092     \edef\l@d@nums{%
9093       000| % Start page = we don't print it
9094       \csuse{\@msd@data@\add@msd@c @linenumberR}| % Start line number
9095       \csuse{\@msd@data@\add@msd@c @sublinenumberR}| % Start subline number
9096       000| % End page number, we don't print it
9097       \ifnumless{\csuse{\@msd@data@\add@msd@c @stopabslinenumberR}}{\csuse{
9098         \@lastabsline@forpage@\the\page@numR}}}%

```

```

9097         {\csuse{@msdata@add@msd@c @stoplinenumber}}}%End line number if
          in the same page
9098         {\csuse{@lastline@forpage@the\page@num}}}%Otherwise, last
          number of the page
9099         |%
9100         \ifnumless{\csuse{@msdata@add@msd@c @stopabslinenumber}}{\csuse{
@lastabsline@forpage@the\page@num}}}%
9101         {\csuse{@msdata@add@msd@c @stopsublinenumber}}}%End subline
          number if in the same page
9102         {\csuse{@lastsubline@forpage@the\page@num}}}%Otherwise, last
          subnumber of the page
9103         |%
9104         \edfont@info%Font
9105         }%
9106         \@msd@options@fullpagefalse%
9107         \if@firstlineofpage%Try if the data are for the full page. If yes
, will add options to the list.
9108         \unless\if@msdata@insertedfrompreviouspage%
9109         \ifnumless{\csuse{@lastabsline@forpage@the\page@num}}{\csuse
{@msdata@add@msd@c @stopabslinenumber}+\@one}%
9110         {%
9111         \numdef{\@tmp}{\add@msd@c+\@one}%
9112         \ifcsdef{@msdata@\@tmp @abslinenumber}%
9113         {\ifnumequal{\csuse{@msdata@\@tmp @abslinenumber}}{\csuse{
@lastabsline@forpage@the\page@num}}}%
9114         {}%
9115         {\@msd@options@fullpagetrue}%
9116         }%
9117         {\@msd@options@fullpagetrue}%
9118         }%
9119         {}%
9120         \fi%
9121         \fi%
9122         \listxadd{\@msdata@list}{%
9123         \@msd@options@iffullpage%
9124         \ifluatex%
9125         \csxdef{footnote@luatextextdir}{\the\textdir}%
9126         \csxdef{footnote@luatexpardir}{\the\pardir}%
9127         \fi%
9128         \csdef{@this@crossref@start}{\add@msd@c:start:msdata}%
9129         \csdef{@this@crossref@end}{\add@msd@c:end:msdata}%
9130         \unexpanded{%
9131         \def\annot@start{}%
9132         \def\annot@end{}%
9133         }%
9134         \noexpand\csuse{v\@msdata@series footnote}{\@msdata@series}{\{
expandonce\l@d@nums}{\ms@data@label}{\expandonce\data}}%
9135         \reset@msd@options@iffullpage%
9136         }%
9137         }%
9138         {}%
9139         \else%
9140         \numgdef{\add@msd@cR}{\add@msd@cR+\@one}%
9141         \ifcsdef{@msdata@add@msd@cR @dataR}{%

```

```

9142 \letcs{\@data}{\@msdata@add@msd@cR @dataR}%
9143 \edef\l@d@nums{%
9144 000| % Start page = we don't print it
9145 \csuse{\@msdata@add@msd@cR @linenumberR}| % Start line number
9146 \csuse{\@msdata@add@msd@cR @sublinenumberR}| % Start subline
number, for now, not used
9147 000| % End page number, we don't print it
9148 \ifnumless{\csuse{\@msdata@add@msd@cR @stopabslinenumberR}}{\
csuse{\@lastline@forpageR@the\page@numR}}}%
9149 {\csuse{\@msdata@add@msd@cR @stoplinenumberR}}}%End line number
if in the same page
9150 {\csuse{\@lastline@forpageR@the\page@numR}}}%Otherwise, last
number of the page
9151 |%
9152 \ifnumless{\csuse{\@msdata@add@msd@cR @stopabslinenumberR}}{\
csuse{\@lastline@forpageR@the\page@numR}}}%
9153 {\csuse{\@msdata@add@msd@cR @stopsublinenumberR}}}%End subline
number if in the same page
9154 {\csuse{\@lastsubline@forpageR@the\page@numR}}}%Otherwise, last
number of the page
9155 000| % End sub line number, for now, not used
9156 \edefont@info%Font
9157 }%
9158 \@msd@options@fullpagefalse%
9159 \if@firstlineofpageR%
9160 \unless\if@msdata@insertedfrompreviouspage%
9161 \ifnumless{\csuse{\@lastabsline@forpageR@the\page@numR}}{\
csuse{\@msdata@add@msd@c @stopabslinenumberR}+\@ne}%
9162 {%
9163 \numdef{\@tmp}{\add@msd@cR+\@ne}%
9164 \ifcsdef{\@msdata@\@tmp @abslinenumberR}%
9165 {\ifnumequal{\csuse{\@msdata@\@tmp @abslinenumberR}}{\csuse{\
@lastabsline@forpageR@the\page@numR}}}%
9166 {}%
9167 {\@msd@options@fullpagetrue}%
9168 }%
9169 {\@msd@options@fullpagetrue}%
9170 }%
9171 {}%
9172 \fi%
9173 \fi%
9174 \listxadd{\@msdata@list}{%
9175 \@msd@options@iffullpage%
9176 \ifluatex%
9177 \csxdef{footnote@luatextextdir}{\the\textdir}%
9178 \csxdef{footnote@luatexpardir}{\the\pardir}%
9179 \fi%
9180 \csdef{@this@crossref@start}{\add@msd@cR:start:msdata}%
9181 \csdef{@this@crossref@end}{\add@msd@cR:end:msdata}%
9182 \unexpanded{%
9183 \def\annot@start{}%
9184 \def\annot@end{}%
9185 }%

```

```

9186 \noexpand\csuse{v\@msdata@series footnote}{\@msdata@series}{\
expandonce\l@d@nums}{\ms@data@label}{\expandonce\@data}}%
9187 \reset@msd@options@if full page%
9188 }%
9189 }%
9190 {}%
9191 \fi%
9192 \egroup%
9193 }%
9194 %

```

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.

```

9195 \newif\if@msdata@insertedfrompreviouspage%
9196 %

```

@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.

```

9197 \newcommand{\add@msdata@firstlineofpage}{%
9198 \bgroup%
9199 \normalfont%
9200 \unless\ifledRcol{%
9201 \ifcsdef{@msdata@add@msd@c @data}{%
9202 \ifnumless{the\absline@num-\@ne}{\csuse{@msdata@\add@msd@c
@stopabslinenumber}}%
9203 {%
9204 \global\@msdata@insertedfrompreviouspagetrue%
9205 \letcs{@data}{@msdata@\add@msd@c @data}%
9206 \edef\l@d@nums{%
9207 000|% Start page = we don't print it
9208 \numexprthe\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
9209 000|% Start subline number, for now, not used
9210 000|% End page number, we don't print it
9211 \ifnumless{\csuse{@msdata@\add@msd@c @stopabslinenumber}}{\csuse{
@lastabsline@forpage@the\page@num}}%
9212 {\csuse{@msdata@\add@msd@c @stoplinenumber}}%End line number if
in the same page
9213 {\csuse{@lastline@forpage@the\page@num}}%Otherwise, last
number of the page
9214 |%
9215 000|% End sub line number, for now, not used
9216 \edefont@info{Font
9217 }%
9218 \@msd@options@fullpagefalse%
9219 \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
9220 {%
9221 \numdef{\@tmp}{\add@msd@c+\@ne}%
9222 \ifcsdef{@msdata@\@tmp @abslinenumber}%

```

```

9223      {\ifnumequal{\csuse{@msdata@\tmp @abslinenumber}}{\csuse{
@lastabsline@forpage@\the\page@num}}}%
9224      }%
9225      {\@msd@options@fullpagetrue}%
9226      }%
9227      {\@msd@options@fullpagetrue}%
9228      }%
9229      }%
9230      \listxadd{\@msdata@list}{%
9231      \@msd@options@iffullpage%
9232      \ifluatex%
9233      \csxdef{footnote@luatextextdir}{\the\textdir}%
9234      \csxdef{footnote@luatexpardir}{\the\pardir}%
9235      \fi%
9236      \csdef{@this@crossref@start}{\add@msd@c:start:msdata}%
9237      \csdef{@this@crossref@end}{\add@msd@c:end:msdata}%
9238      \unexpanded{%
9239      \def\annot@start{}%
9240      \def\annot@end{}%
9241      }%
9242      \noexpand\csuse{v\@msdata@series footnote}{\@msdata@series}{\@
expandonce\l@d@nums}{\ms@data@label}{\expandonce\@data}}%
9243      \reset@msd@options@iffullpage%
9244      }%
9245      }%
9246      {\global\@msdata@insertedfrompreviouspagefalse}%
9247      }%
9248      \else%
9249      \ifcsdef{@msdata@\add@msd@cR @dataR}{%
9250      \ifnumless{\the\absline@numR-\@one}{\csuse{@msdata@\add@msd@cR
@stopabslinenumberR}}%
9251      {%
9252      \global\@msdata@insertedfrompreviouspagetrue%
9253      \letcs{@data}{\@msdata@\add@msd@cR @dataR}%
9254      \edef\l@d@nums{%
9255      000| % Start page = we don't print it
9256      \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
9257      000| % Start subline number, for now, not used
9258      000| % End page number, we don't print it
9259      \ifnumless{\csuse{@msdata@\add@msd@cR @stopabslinenumberR}}{\@
csuse{@lastline@forpageR@\the\page@numR}}%
9260      {\csuse{@msdata@\add@msd@cR @stoplinenumberR}}%End line number
if in the same page
9261      {\csuse{@lastline@forpageR@\the\page@numR}}%Otherwise, last
number of the page
9262      |%
9263      000| % End sub line number, for now, not used
9264      \edefont@info%Font
9265      }%
9266      \@msd@options@fullpagefalse%
9267      \ifnumless{\csuse{@lastabsline@forpageR@\the\page@numR}}{\csuse{
@msdata@\add@msd@cR @stopabslinenumberR}+\@one}%

```



```

9268      {%
9269      \numdef{\@tmp}{\add@msd@cR+\@ne}%
9270      \ifcsdef{@msdata@\@tmp @abslinenumberR}%
9271      {\ifnumequal{\csuse{@msdata@\@tmp @abslinenumberR}}{\csuse{
@lastabsline@forpageR}\the\page@numR}}}%
9272      {}%
9273      {\@msd@options@fullpagetrue}%
9274      }%
9275      {\@msd@options@fullpagetrue}%
9276      }%
9277      {}%
9278      \listxadd{\@msdata@list}{%
9279      \@msd@options@iffullpage%
9280      \ifluatex%
9281      \csxdef{footnote@luatextextdir}{\the\textdir}%
9282      \csxdef{footnote@luatexpardir}{\the\pardir}%
9283      \fi%
9284      \csdef{@this@crossref@start}{\add@msd@cR:start:msdata}%
9285      \csdef{@this@crossref@end}{\add@msd@cR:end:msdata}%
9286      \unexpanded{%
9287      \def\annot@start{}%
9288      \def\annot@end{}%
9289      }%
9290      \noexpand\csuse{v\@msdata@series footnote}{\@msdata@series}{\{
expandonce\l@d@nums}{\ms@data@label}{\expandonce\@data}}}%
9291      \reset@msd@options@iffullpage%
9292      }%
9293      }%
9294      {\global\@msdata@insertedfrompreviouspagefalse}%
9295      }{}%
9296      \fi%
9297      \egroup%
9298      }%
9299      %

```

XXXI.6 Inserting footnote

Just before inserting standard insert (familiar and critical footnotes, sidenotes), we call `\insert@msdata` to insert manuscripts data's footnotes.

```

\insert@msdata \newcommand{\insert@msdata}{%
9301   \def\do##1{##1}%
9302   \dolistloop{\@msdata@list}%
9303   \global\let\@msdata@list\relax%
9304   }%
9305   %

```

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.

```

9306 \newif\if@msd@options@fullpage%
9307 \newcommand{\@msd@options@iffullpage}[0]{%
9308   \if@msd@options@fullpage%
9309     \noexpand\toggletrue{nonum@}%
9310     \ifdefined\ms@data@label}%
9311       {\noexpand\toggletrue{nosep@}}}%
9312   }%
9313 \fi%
9314 }%
9315 \newcommand{\reset@msd@options@iffullpage}[0]{%
9316   \noexpand\togglefalse{nonum@}%
9317   \noexpand\togglefalse{nosep@}%
9318 }%
9319 %

```

XXXII Arrays and tables

XXXII.1 Preamble: macro as environment

The following is borrowed, and renamed, from the `amsmath` package. See also the CTT thread “`eeq` and `amstex`”, 1995/08/31, started by Keith Reckdahl and ended definitively by David M. Jones.

Several of the `[math]` macros scan their body twice. This means we must collect all text in the body of an environment form before calling the macro.

`\@emptytoks` This is actually defined in the `amsgen` package.

```

9320 \newtoks\@emptytoks
9321
9322 %

```

The rest is from `amsmath`.

`\l@denvbody` A token register to contain the body.

```

9323 \newtoks\l@denvbody
9324
9325 %

```

`\addtol@denvbody` `\addtol@denvbody{arg}` adds `arg` to the token register `\l@denvbody`.

```

9326 \newcommand{\addtol@denvbody}[1]{%
9327   \global\l@denvbody\expandafter{\the\l@denvbody#1}}
9328 %

```

`\if@edtabular` A boolean set to true inside a `reledmac`’ tabular environment.

```

9329 \newif\if@edtabular%
9330 %

```

`\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`.

```

9331 \newcommand{\l@dcollect@body}[1]{%
9332   \ifvmode\else%
9333   \\\%Ensure that we do not have spurious indentation at the very first
line, caused by some text before.
9334   \fi%
9335   \l@denvbody{\expandafter#1\expandafter{\the\l@denvbody}}}%
9336   \edef\processl@denvbody{\the\l@denvbody\noexpand\end{\@currentenv}}%
9337   \l@denvbody\@emptytoks \def\l@dbegin@stack{b}%
9338   \begingroup
9339   \@edtabulartrue%
9340   \expandafter\let\csname\@currentenv\endcsname\l@dcollect@body
9341   \edef\processl@denvbody{\expandafter\noexpand\csname\@currentenv\
endcsname}%
9342   \processl@denvbody%
9343   }%
9344
9345 %

```

`\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.

```

9346 \def\l@dpush@begins#1\begin#2{%
9347   \ifx\end#2\else b\expandafter\l@dpush@begins\fi}
9348
9349 %

```

`\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.

```

9350 \def\l@dcollect@@body#1\end#2{%
9351   \edef\l@dbegin@stack{\l@dpush@begins#1\begin\end
9352     \expandafter\@gobble\l@dbegin@stack}%
9353   \ifx\@empty\l@dbegin@stack
9354     \endgroup
9355     \@checkend{#2}%
9356     \addtol@denvbody{#1}%
9357   \else
9358     \addtol@denvbody{#1\end{#2}}%
9359   \fi
9360   \processl@denvbody % A little tricky! Note the grouping
9361 }
9362
9363 %

```

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.

```

9364 \newcommand*{\l@dtabnoexpands}{%
9365   \let\rtab=0%
9366   \let\ctab=0%
9367   \let\ltab=0%
9368   \let\rtabtext=0%
9369   \let\ltabtext=0%
9370   \let\ctabtext=0%
9371   \let\edbeforetab=0%
9372   \let\edaftertab=0%
9373   \let\edatleft=0%
9374   \let\edatright=0%
9375   \let\edvertline=0%
9376   \let\edvertdots=0%
9377   \let\edrowfill=0%
9378 }
9379
9380 %

```

\disable@familiarnotes Macros to disable and restore familiar notes, to prevent them from printing multiple times in edtabularx and edarrayx environments.

\restore@familiarnotes

```

9381 \newcommand{\disable@familiarnotes}{%
9382   \unless\ifnofamiliar@%
9383     \def\do##1{%
9384       \csletcs{footnote@##1}{footnote##1}%
9385       \expandafter\renewcommand \csname footnote##1\endcsname[1]{%
9386         \protected@csxdef{@thefnmark##1}{\csuse{thefootnote##1}}%
9387         \csuse{@footnotemark##1}%
9388       }%
9389     }%
9390     \dolistloop{\@series}%
9391   \fi%
9392 }%
9393 \newcommand{\restore@familiarnotes}{%
9394   \unless\ifnofamiliar@%
9395     \def\do##1{%
9396       \csletcs{footnote##1}{footnote@##1}%
9397     }%
9398     \dolistloop{\@series}%
9399   \fi%
9400 }%
9401
9402 %

```

\disable@sidenotes The same for sidenotes.

\restore@sidenotes

```

9403 \newcommand{\disable@sidenotes}{%
9404   \let\@@ledrightnote\ledrightnote%
9405   \let\@@ledleftnote\ledleftnote%
9406   \let\@@ledsidenote\ledsidenote%
9407   \let\ledrightnote@gobble%
9408   \let\ledleftnote@gobble%
9409   \let\ledsidenote@gobble%
9410 }%

```

```

9411 \newcommand{\restore@sidenotes}{%
9412   \let\ledrightnote\@ledrightnote%
9413   \let\ledleftnote\@ledleftnote%
9414   \let\ledsidenote\@ledsidenote%
9415 }%
9416 %

```

\disable@endnotes The same for endnotes.

\restore@endnotes

```

9417 \newcommand{\disable@endnotes}{%
9418   \unless\ifnoend{%
9419     \def\do##1{%
9420       \csletcs{##1@endnote}{##1endnote}%
9421       \expandafter\renewcommand \csname ##1endnote\endcsname[1]{}%
9422     }%
9423     \dolistloop{\@series}%
9424   \fi%
9425 }%
9426 \newcommand{\restore@endnotes}{%
9427   \unless\ifnofamiliar{%
9428     \def\do##1{%
9429       \csletcs{##1endnote}{##1@endnote}%
9430     }%
9431     \dolistloop{\@series}%
9432   \fi%
9433 }%
9434 %

```

\disable@notes Disable/restore side, familiar and end notes.

\restore@notes

```

9435 \newcommand{\disable@notes}{%
9436   \disable@sidenotes%
9437   \disable@familiarnotes%
9438   \disable@endnotes%
9439 }%
9440 \newcommand{\restore@notes}{%
9441   \restore@sidenotes%
9442   \restore@familiarnotes%
9443   \restore@endnotes%
9444 }%
9445 %

```

\EDTEXT We need to be able to modify the `\edtext` macros and also restore their original definitions.

\xedtext

```

9446 \let\EDTEXT=\edtext
9447 \newcommand{\xedtext}[2]{\EDTEXT{#1}{#2}}
9448 %

```

\EDLABEL We need to be able to modify and restore the `\edlabel` macro.

\xedlabel

```

9449 \let\EDLABEL=\edlabel
9450 \newcommand*{\xedlabel}[1]{\EDLABEL{#1}}
9451 %

```

<code>\xedindex</code>	Macros supporting modification and restoration of <code>\edindex</code> .
<code>\nulledindex</code>	<pre> 9452 \AtBeginDocument{\let\xedindex\edindex}% 9453 \newcommand{\nulledindex}[2][\jobname]{\@bsphack\esphack} 9454 9455 % </pre>
<code>\@line@@num</code>	Macro supporting restoration of <code>\linenum</code> . <pre> 9456 \let\@line@@num=\linenum 9457 % </pre>
<code>\l@dgobbleoptarg</code>	<code>\l@dgobbleoptarg[⟨arg⟩]{⟨arg⟩}</code> replaces these two arguments (first is optional) by <code>\relax</code> . <pre> 9458 \newcommand*\l@dgobbleoptarg[2][]{\relax}% 9459 9460 % </pre>
<code>\l@secondmandarg</code>	<code>\l@secondoptarg[⟨arg⟩]{⟨arg⟩}</code> gobble the first (optional) argument, and expand to the second (mandatory) argument. <pre> 9461 \NewExpandableDocumentCommand{\l@secondmandarg}{om}{#2}% 9462 % </pre>
<code>\Relax</code>	<code>\let\Relax=\relax</code>
<code>\NEXT</code>	<code>\let\NEXT=\next</code> <pre> 9465 % 9466 % </pre>
<code>\l@dmodforedtext</code>	Modify and restore various macros for when <code>\edtext</code> is used.
<code>\l@drestoreforedtext</code>	<pre> 9467 \newcommand{\l@dmodforedtext}{% 9468 \let\edtext\relax 9469 \def\do##1{\global\csletcs{##1footnote}{\l@dgobbleoptarg}}% 9470 \dolistloop{\@series}% 9471 \let\edindex\nulledindex 9472 \let\linenum@gobble} 9473 \newcommand{\l@drestoreforedtext}{% 9474 \def\do##1{\global\csletcs{##1footnote}{##1@@footnote}} 9475 \dolistloop{\@series}% 9476 \let\edindex\xedindex} 9477 % </pre>
<code>\l@dnullfills</code>	Nullify and restore some column fillers, etc.
<code>\l@drestorefills</code>	<pre> 9478 \newcommand{\l@dnullfills}{% 9479 \def\edlabel##1{}% 9480 \def\edrowfill##1##2##3{}% 9481 } 9482 \newcommand{\l@drestorefills}{% 9483 \def\edrowfill##1##2##3{\@EDROWFILL@{##1}{##2}{##3}}% 9484 } 9485 9486 % </pre>

`\letsforverteilen` Gathers some lets and other code that is common to the `*verteilen*` macros.

```

9487 \newcommand{\letsforverteilen}{%
9488   \let\edtext\xedtext
9489   \let\edindex\xedindex
9490   \def\do##1{\global\csletcs{##1footnote}{##1@footnote}}
9491   \dolistloop{\@series}%
9492   \let\linenum\@line@num
9493   \hilfsskip=\l@dcolwidth%
9494   \advance\hilfsskip by -\wd\hilfsbox
9495   \def\edlabel##1{\xedlabel{##1}}
9496
9497 %

```

`\disablel@dtabfeet` Declarations for using or using `\edtext` inside tabulars. The default at this point is for `\edtext`.

`\enablel@dtabfeet`

```

9498 \newcommand\disablel@dtabfeet{\l@dmodforedtext}%
9499 \newcommand\enablel@dtabfeet{\l@drestoreforedtext}%
9500 %

```

XXXII.2.2 Counters, boxes and lengths

`\l@dampcount` `\l@dampcount` is a counter for the & column dividers and `\l@dcolcount` is a counter for the columns.

`\l@dcolcount`

```

9501 \newcount\l@dampcount
9502   \l@dampcount=1\relax
9503 \newcount\l@dcolcount
9504   \l@dcolcount=0\relax
9505
9506 %

```

`\hilfsbox` Some (temporary) helper items.

```

\hilfsskip
\Hilfsbox
\hilfscount
9507 \newbox\hilfsbox
9508 \newskip\hilfsskip
9509 \newbox\Hilfsbox
9510 \newcount\hilfscount
9511
9512 %

```

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).

```

9513 \newdimen\dcoli
9514 \newdimen\dcolii
9515 \newdimen\dcoliii
9516 \newdimen\dcoliv
9517 \newdimen\dcolv
9518 \newdimen\dcolvi
9519 \newdimen\dcolvii
9520 \newdimen\dcolviii
9521 \newdimen\dcolix

```

```

9522 \newdimen\dcoldx
9523 \newdimen\dcoldxi
9524 \newdimen\dcoldxii
9525 \newdimen\dcoldxiii
9526 \newdimen\dcoldxiv
9527 \newdimen\dcoldxv
9528 \newdimen\dcoldxvi
9529 \newdimen\dcoldxvii
9530 \newdimen\dcoldxviii
9531 \newdimen\dcoldxix
9532 \newdimen\dcoldxx
9533 \newdimen\dcoldxxi
9534 \newdimen\dcoldxxii
9535 \newdimen\dcoldxxiii
9536 \newdimen\dcoldxxiv
9537 \newdimen\dcoldxxv
9538 \newdimen\dcoldxxvi
9539 \newdimen\dcoldxxvii
9540 \newdimen\dcoldxxviii
9541 \newdimen\dcoldxxix
9542 \newdimen\dcoldxxx
9543 \newdimen\dcolderr % added for error handling
9544
9545 %

```

\l@dcoldwidth This is a cunning way of storing the columnwidths indexed by the column number **\l@dcoldcount**, like an array. (was *\Dimenzuordnung*)

```

9546 \newcommand{\l@dcoldwidth}{\ifcase \the\l@dcoldcount \dcoli %???
9547 \or \dcoli \or \dcolii \or \dcoliii
9548 \or \dcoliv \or \dcolv \or \dcolvi
9549 \or \dcolvii \or \dcolviii \or \dcolix \or \dcoldx
9550 \or \dcolxi \or \dcolxii \or \dcolxiii
9551 \or \dcolxiv \or \dcolxv \or \dcolxvi
9552 \or \dcolxvii \or \dcolxviii \or \dcolxix \or \dcoldxx
9553 \or \dcoldxxi \or \dcoldxxii \or \dcoldxxiii
9554 \or \dcoldxxiv \or \dcoldxxv \or \dcoldxxvi
9555 \or \dcoldxxvii \or \dcoldxxviii \or \dcoldxxix \or \dcoldxxx
9556 \else \dcolderr \fi}
9557
9558 %

```

\step1@dcoldcount This increments the column counter, and issues an error message if it is too large.

```

9559 \newcommand*{\step1@dcoldcount}{\advance\l@dcoldcount\@ne
9560 \ifnum\l@dcoldcount>30\relax
9561 \led@err@TooManyColumns
9562 \fi}
9563
9564 %

```

\l@dsetmaxcoldwidth Sets the column width to the maximum value seen so far.

```

9565 \newcommand{\l@dsetmaxcoldwidth}{%

```

```

9566 \ifdim\l@dcwidth < \wd\hilfsbox
9567 \l@dcwidth = \wd\hilfsbox
9568 \else \relax \fi}
9569
9570 %

```

\measuremcell Measure (recursively) the width required for a math cell.

```

9571 \def\measuremcell #1{%
9572   \ifx #1\ \ifnum\l@dcwidth=0\let\NEXT\relax%
9573     \else\l@dcwidth=0%
9574       \l@dcwidth=0%
9575       \let\NEXT\measuremcell%
9576     \fi%
9577   \else\setbox\hilfsbox=\hbox{$\displaystyle{#1}$}%
9578     \step\l@dcwidth%
9579     \l@dcwidth=0%
9580     \let\NEXT\measuremcell%
9581   \fi\NEXT}
9582
9583 %

```

\measuretcell Measure (recursively) the width required for a text cell.

```

9584 \def\measuretcell #1{%
9585   \ifx #1\ \ifnum\l@dcwidth=0\let\NEXT\relax%
9586     \else\l@dcwidth=0%
9587       \l@dcwidth=0%
9588       \let\NEXT\measuretcell%
9589     \fi%
9590   \else\setbox\hilfsbox=\hbox{#1}%
9591     \step\l@dcwidth%
9592     \l@dcwidth=0%
9593     \let\NEXT\measuretcell%
9594   \fi\NEXT}
9595
9596 %

```

\measuremrow Measure (recursively) the width required for a math row.

```

9597 \def\measuremrow #1{%
9598   \ifx #1\ \let\NEXT\relax%
9599   \else\measuremcell #1\&\&\&%
9600     \let\NEXT\measuremrow%
9601   \fi\NEXT}
9602
9603 %

```

\measuretrow Measure (recursively) the width required for a text row.

```

9603 \def\measuretrow #1{%
9604   \ifx #1\ \let\NEXT\relax%
9605   \else\measuretcell #1\&\&\&%
9606     \let\NEXT\measuretrow%
9607   \fi\NEXT}
9608
9609 %

```

\edtabcolsep The length \edtabcolsep controls the distance between columns.

```
9610 \newskip\edtabcolsep
9611 \global\edtabcolsep=10pt
9612
9613 %
```

```
\variab14 \newcommand{\variab}{\relax}
9615
9616 %
```

\l@dcheckcols Check that the number of columns is consistent.

```
9617 \newcommand*{\l@dcheckcols}{%
9618   \ifnum\l@dcolcount=1\relax
9619   \else
9620     \ifnum\l@dampcount=1\relax
9621     \else
9622       \ifnum\l@dcolcount=\l@dampcount\relax
9623       \else
9624         \l@derr@UnequalColumns
9625       \fi
9626     \fi
9627     \l@dampcount=\l@dcolcount
9628   \fi}
9629
9630 %
```

\edfilldimen A length.

```
9631 \newdimen\edfilldimen
9632 \edfilldimen=0pt
9633
9634 %
```

\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 E_T_X command, because some packages, like babel can override it in some specific cases (Greek, for example).

\theadcolcount

```
9635 \newcounter{addcolcount}
9636 \renewcommand{\theadcolcount}{\romannumeral \c@addcolcount}
9637 %
```

XXXII.2.3 Tabular typesetting

\setmcellright Typeset (recursively) cells of display math right justified.

```
9638 \def\setmcellright #1&{\def\edlabel##1}{%
9639   \let\edindex\nulledindex
9640   \ifx #1\\ \ifnum\l@dcolcount=0%\removelastskip
9641     \let\Next\relax%
9642   \else\l@dcolcount=0%
9643     \let\Next=\setmcellright%
```

```

9644         \fi%
9645     \else%
9646         \disablel@dtabfeet%
9647         \step1@dcolcount%
9648         \disable@notes%
9649         \setbox\hilfsbox=\hbox{$\displaystyle{#1}$}%
9650         \restore@notes%
9651         \letsforverteilen%
9652         \hskip\hilfsskip$\displaystyle{#1}$%
9653         \hskip\edtabcolsep%
9654         \let\Next=\setmcellright%
9655     \fi\Next}
9656
9657 %

```

\settcellright Typeset (recursively) cells of text right justified.

```

9658 \def\settcellright #1&{\def\edlabel##1{}}%
9659     \let\edindex\nulledindex
9660     \ifx #1\\ \ifnum\l@dcolcount=0%\removelastskip
9661         \let\Next\relax%
9662     \else\l@dcolcount=0%
9663         \let\Next=\settcellright%
9664     \fi%
9665 \else%
9666     \disablel@dtabfeet%
9667     \step1@dcolcount%
9668     \disable@notes%
9669     \setbox\hilfsbox=\hbox{#1}%
9670     \restore@notes%
9671     \letsforverteilen%
9672     \hskip\hilfsskip#1%
9673     \hskip\edtabcolsep%
9674     \let\Next=\settcellright%
9675 \fi\Next}
9676 %

```

\setmcellleft Typeset (recursively) cells of display math left justified.

```

9677 \def\setmcellleft #1&{\def\edlabel##1{}}%
9678     \let\edindex\nulledindex
9679     \ifx #1\\ \ifnum\l@dcolcount=0 \let\Next\relax%
9680         \else\l@dcolcount=0%
9681             \let\Next=\setmcellleft%
9682         \fi%
9683 \else \disablel@dtabfeet%
9684         \step1@dcolcount%
9685         \disable@notes%
9686         \setbox\hilfsbox=\hbox{$\displaystyle{#1}$}%
9687         \restore@notes%
9688         \letsforverteilen%
9689         $\displaystyle{#1}$\hskip\hilfsskip\hskip\edtabcolsep%
9690         \let\Next=\setmcellleft%
9691 \fi\Next}

```

```

9692 %
9693 %
\settcellleft Typeset (recursively) cells of text left justified.
9694 \def\settcellleft #1&{\def\edlabel##1{}}%
9695 \let\edindex\nulledindex
9696 \ifx #1\ \ifnum\l@dc@count=0 \let\Next\relax%
9697 \else\l@dc@count=0%
9698 \let\Next=\settcellleft%
9699 \fi%
9700 \else \disablel@dtabfeet%
9701 \step1@dc@count%
9702 \disable@notes%
9703 \setbox\hilsbox=\hbox{#1}%
9704 \restore@notes%
9705 \letsforverteilen%
9706 #1\hskip\hilsfsskip\hskip\edtabcolsep%
9707 \let\Next=\settcellleft%
9708 \fi\Next}
9709 %

```

\setmcellcenter Typeset (recursively) cells of display math centered.

```

9710 \def\setmcellcenter #1&{\def\edlabel##1{}}%
9711 \let\edindex\nulledindex
9712 \ifx #1\ \ifnum\l@dc@count=0\let\Next\relax%
9713 \else\l@dc@count=0%
9714 \let\Next=\setmcellcenter%
9715 \fi%
9716 \else \disablel@dtabfeet%
9717 \step1@dc@count%
9718 \disable@notes%
9719 \setbox\hilsbox=\hbox{$\displaystyle{#1}$}%
9720 \restore@notes%
9721 \letsforverteilen%
9722 \hskip 0.5\hilsfsskip$\displaystyle{#1}$\hskip0.5\hilsfsskip%
9723 \hskip\edtabcolsep%
9724 \let\Next=\setmcellcenter%
9725 \fi\Next}
9726 %
9727 %

```

\settcellcenter Typeset (recursively) cells of text centered.

```

9728 \def\settcellcenter #1&{\def\edlabel##1{}}%
9729 \let\edindex\nulledindex
9730 \ifx #1\ \ifnum\l@dc@count=0 \let\Next\relax%
9731 \else\l@dc@count=0%
9732 \let\Next=\settcellcenter%
9733 \fi%
9734 \else \disablel@dtabfeet%
9735 \step1@dc@count%
9736 \disable@notes%

```

```

9737         \setbox\hilfsbox=\hbox{#1}%
9738         \restore@notes%
9739         \letsforverteilen%
9740         \hskip 0.5\hilfsskip #1\hskip 0.5\hilfsskip%
9741         \hskip\edtabcolsep%
9742         \let\Next=\settcellcenter%
9743     \fi\Next}
9744
9745 %

```

\NEXT₄₆ \let\NEXT=\relax

```

9747
9748 %

```

\setmrowright Typeset (recursively) rows of right justified math.

```

9749 \def\setmrowright #1\{%
9750     \ifx #1& \let\NEXT\relax
9751     \else \centerline{\setmcellright #1&\&\&\&}
9752         \let\NEXT=\setmrowright
9753     \fi\NEXT}
9754 %

```

\settroright Typeset (recursively) rows of right justified text.

```

9755 \def\settroright #1\{%
9756     \ifx #1& \let\NEXT\relax
9757     \else \centerline{\settcellright #1&\&\&\&}
9758         \let\NEXT=\settroright
9759     \fi\NEXT}
9760
9761 %

```

\setmrowleft Typeset (recursively) rows of left justified math.

```

9762 \def\setmrowleft #1\{%
9763     \ifx #1& \let\NEXT\relax
9764     \else \centerline{\setmcellleft #1&\&\&\&}
9765         \let\NEXT=\setmrowleft
9766     \fi\NEXT}
9767 %

```

\settrorleft Typeset (recursively) rows of left justified text.

```

9768 \def\settrorleft #1\{%
9769     \ifx #1& \let\NEXT\relax
9770     \else \centerline{\settcellleft #1&\&\&\&}
9771         \let\NEXT=\settrorleft
9772     \fi\NEXT}
9773
9774 %

```

\setmrowcenter Typeset (recursively) rows of centered math.

```

9775 \def\setmrowcenter #1\\{%
9776   \ifx #1& \let\NEXT\relax%
9777   \else \centerline{\setmcellcenter #1&\\&\\&}
9778     \let\NEXT=\setmrowcenter
9779   \fi\NEXT}
9780 %

```

\settrowcenter Typeset (recursively) rows of centered text.

```

9781 \def\settrowcenter #1\\{%
9782   \ifx #1& \let\NEXT\relax
9783   \else \centerline{\settcclcenter #1&\\&\\&}
9784     \let\NEXT=\settrowcenter
9785   \fi\NEXT}
9786 %
9787 %

```

```

\newcommand{\nullsetzen}88 {\newcommand{\nullsetzen}{%
9789   \step1@dcolcount%
9790   \l@dcolwidth=0pt%
9791   \ifnum\l@dcolcount=30\let\NEXT\relax%
9792     \l@dcolcount=0\relax
9793   \else\let\NEXT\nullsetzen%
9794   \fi\NEXT}
9795 %
9796 %

```

\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.

```

9797 \newcommand{\edatleft}[3][\@empty]{%
9798   \ifx#1\@empty
9799     \vbox to 10pt{\vss\hbox{$\left#2\vrule width0pt height #3
9800       depth 0pt \right. $\hss}\vfil}
9801   \else
9802     \vbox to 4pt{\vss\hbox{$#1\left#2\vrule width0pt height #3
9803       depth 0pt \right. $\}\vfil}
9804   \fi}
9805 %

```

\edatright $\edatright[\langle math \rangle]{\langle symbol \rangle}{\langle len \rangle}$. Right $\langle symbol \rangle$, $2\langle len \rangle$ high with appended $\langle math \rangle$ vertically centered.

```

9806 \newcommand{\edatright}[3][\@empty]{%
9807   \ifx#1\@empty
9808     \vbox to 10pt{\vss\hbox{$\left.\vrule width0pt height #3
9809       depth 0pt \right#2 $\hss}\vfil}
9810   \else
9811     \vbox to 4pt{\vss\hbox{$\left.\vrule width0pt height #3
9812       depth 0pt \right#2 #1 $\}\vfil}
9813   \fi}
9814 %
9815 %

```


`\edvertline` `\edvertline{⟨len⟩}` vertical line *⟨len⟩* high.

```
9816 \newcommand{\edvertline}[1]{\vbox to 8pt{\vss\hbox{\vrule height #1}\vfil}}
9817
9818 %
```

`\edvertdots` `\edvertdots{⟨len⟩}` vertical dotted line *⟨len⟩* high.

```
9819 \newcommand{\edvertdots}[1]{\vbox to 1pt{\vss\vbox to #1%
9820     {\cleaders\hbox{$\m@th\hbox{.}\vbox to 0.5em{ }}\vfil}}}
9821
9822 %
```

`\l@dtabaddcols` `\l@dtabaddcols{⟨startcol⟩}{⟨endcol⟩}` adds the widths of the columns *⟨startcol⟩* through *⟨endcol⟩* to `\edfilldimen`. It is a \LaTeX style reimplementation of the original `\@add@`.

```
9823 \newcommand{\l@dtabaddcols}[2]{%
9824     \l@dcheckstartend{#1}{#2}%
9825     \ifl@dstartendok
9826     \setcounter{addcolcount}{#1}%
9827     \@whilenum \value{addcolcount}<#2\relax \do
9828     {\advance\edfilldimen by \the \csname dcol\theaddcolcount\endcsname
9829     \advance\edfilldimen by \edtabcolsep
9830     \stepcounter{addcolcount}}%
9831     \advance\edfilldimen by \the \csname dcol\theaddcolcount\endcsname
9832     \fi
9833 }
9834
9835 %
```

`\ifl@dstartendok` `\l@dcheckstartend{⟨startcol⟩}{⟨endcol⟩}` checks that the values of *⟨startcol⟩* and *⟨endcol⟩* are sensible. If they are then `\ifl@dstartendok` is set TRUE, otherwise it is set FALSE.

```
9836 \newif\ifl@dstartendok
9837 \newcommand{\l@dcheckstartend}[2]{%
9838     \l@dstartendoktrue
9839     \ifnum #1<\@ne
9840         \l@dstartendokfalse
9841         \led@err@LowStartColumn
9842     \fi
9843     \ifnum #2>30\relax
9844         \l@dstartendokfalse
9845         \led@err@HighEndColumn
9846     \fi
9847     \ifnum #1>#2\relax
9848         \l@dstartendokfalse
9849         \led@err@ReverseColumns
9850     \fi
9851 }
9852
9853 %
```

`\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 `\wapunktet` macros.

```

9854 \newcommand*{\edrowfill}[3]{%
9855   \l@dtabaddcols{#1}{#2}%
9856   \hb@xt@ \the\l@dcolwidth{\hb@xt@ \the\edfilldimen{#3}\hss}}
9857 \let\@edrowfill=\edrowfill
9858 \def\@EDROWFILL@#1#2#3{\@edrowfill@{#1}{#2}{#3}}
9859 %
9860 %

```

`\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`.

```

9861 \newcommand{\leftltab}[1]{%
9862   \hb@xt@ \z@{\vbox{\edtabindent%
9863     \moveleft\Hilfsskip\hbox{\ #1}}\hss}}
9864 %
9865 %

```

`\leftrtab` `\leftrtab{<text>}{<math>}` for `\edbeforetab` in `\rtab`.

```

9866 \newcommand{\leftrtab}[2]{%
9867   #2\hb@xt@ \z@{\vbox{\edtabindent%
9868     \advance\Hilfsskip by\dcoli%
9869     \moveleft\Hilfsskip\hbox{\ #1}}\hss}}
9870 %
9871 %

```

`\leftctab` `\leftctab{<text>}{<math>}` for `\edbeforetab` in `\ctab`.

```

9872 \newcommand{\leftctab}[2]{%
9873   \hb@xt@ \z@{\vbox{\edtabindent\l@dcolcount=\l@dampcount%
9874     \advance\Hilfsskip by 0.5\dcoli%
9875     \setbox\hilfsbox=\hbox{\def\edlabel##1}%
9876     \disablel@dtabfeet$\displaystyle{#2}$}%
9877     \advance\Hilfsskip by -0.5\wd\hilfsbox%
9878     \moveleft\Hilfsskip\hbox{\ #1}}\hss}%
9879   #2}
9880 %
9881 %

```

`\rightctab` `\rightctab{<math>}{<text>}` for `\edaftertab` in `\ctab`.

```

9882 \newcommand{\rightctab}[2]{%
9883   \setbox\hilfsbox=\hbox{\def\edlabel##1}%
9884   \disablel@dtabfeet#2\l@dampcount=\l@dcolcount%
9885   #1\hb@xt@ \z@{\vbox{\edtabindent\l@dcolcount=\l@dampcount%

```

```

9886 \advance\Hilfsskip by 0.5\l@dcwidth%
9887 \advance\Hilfsskip by -\wd\hilfsbox%
9888 \setbox\hilfsbox=\hbox{\def\edlabel##1{%
9889 \disablel@dtabfeet$\displaystyle{#1}$}%
9890 \advance\Hilfsskip by -0.5\wd\hilfsbox%
9891 \advance\Hilfsskip by \edtabcolsep%
9892 \moveright\Hilfsskip\hbox{ #2}}\hss}%
9893 }
9894
9895 %

```

\rightltab `\rightltab{<math><math>}{<text>}` for `\edaftertab` in `\ltab`.

```

9896 \newcommand{\rightltab}[2]{%
9897 \setbox\hilfsbox=\hbox{\def\edlabel##1{%
9898 \disablel@dtabfeet#2}\l@dampcount=\l@dcwidth%
9899 #1\hb@xt@{z@{\vbox{\edtabindent\l@dcwidth=\l@dampcount%
9900 \advance\Hilfsskip by\l@dcwidth%
9901 \advance\Hilfsskip by-\wd\hilfsbox%
9902 \setbox\hilfsbox=\hbox{\def\edlabel##1{%
9903 \disablel@dtabfeet$\displaystyle{#1}$}%
9904 \advance\Hilfsskip by-\wd\hilfsbox%
9905 \advance\Hilfsskip by\edtabcolsep%
9906 \moveright\Hilfsskip\hbox{ #2}}\hss}%
9907 }
9908
9909 %

```

\rightrtab `\rightrtab{<math><math>}{<text>}` for `\edaftertab` in `\rtab`.

```

9910 \newcommand{\rightrtab}[2]{%
9911 \setbox\hilfsbox=\hbox{\def\edlabel##1{%
9912 \disablel@dtabfeet#2}%
9913 #1\hb@xt@{z@{\vbox{\edtabindent%
9914 \advance\Hilfsskip by-\wd\hilfsbox%
9915 \advance\Hilfsskip by\edtabcolsep%
9916 \moveright\Hilfsskip\hbox{ #2}}\hss}%
9917 }
9918
9919 %

```

\rtab `\rtab{<body>}` typesets `<body>` as an array with the entries right justified.

\edbeforetab The process is first to measure the `<body>` to get the column widths, and then in a
\edaftertab second pass to typeset the body.

```

9920 \newcommand{\rtab}[1]{%
9921 \l@dnnullfills
9922 \def\edbeforetab##1##2{\lefttab{##1}{##2}}%
9923 \def\edaftertab##1##2{\righttab{##1}{##2}}%
9924 \measurebody{#1}%
9925 \l@drestorefills
9926 \variab
9927 \setmrowright #1\\&\\%
9928 \enablel@dtabfeet}

```

```

9929
9930 %

\measurebody \measurebody{⟨body⟩} measures the array ⟨body⟩.

9931 \newcommand{\measurebody}[1]{%
9932   \disablel@dtabfeet%
9933   \l@dc@colcount=0%
9934   \nullsetzen%
9935   \l@dc@colcount=0
9936   \measuremrow #1\\&\\%
9937   \global\l@dampcount=1}
9938
9939 %

\rtabtext \rtabtext{⟨body⟩} typesets ⟨body⟩ as a tabular with the entries right justified.

9940 \newcommand{\rtabtext}[1]{%
9941   \l@dnullfills
9942   \measuretbody{#1}%
9943   \l@drestorefills
9944   \variab
9945   \settroright #1\\&\\%
9946   \enablel@dtabfeet}
9947
9948 %

\measuretbody \measuretbody{⟨body⟩} measures the tabular ⟨body⟩.

9949 \newcommand{\measuretbody}[1]{%
9950   \disable@notes%
9951   \disablel@dtabfeet%
9952   \l@dc@colcount=0%
9953   \nullsetzen%
9954   \l@dc@colcount=0
9955   \measuretbody #1\\&\\%
9956   \restore@notes%
9957   \global\l@dampcount=1}
9958
9959 %

\ltab Array with entries left justified.
\edbeforetab \newcommand{\ltab}[1]{%
\edaftertab 9960 \l@dnullfills
9961 \def\edbeforetab##1##2{\leftltab{##1}{##2}}%
9962 \def\edaftertab##1##2{\rightltab{##1}{##2}}%
9963 \measuretbody{#1}%
9964 \l@drestorefills
9965 \variab
9966 \setmrowleft #1\\&\\%
9967 \enablel@dtabfeet}
9968
9969 %
9970 %

```

\ltabtext Tabular with entries left justified.

```

9971 \newcommand{\ltabtext}[1]{%
9972   \l@dnnullfills
9973   \measuretbody{#1}%
9974   \l@drestorefills
9975   \variab
9976   \settrrowleft #1\&\&\&%
9977   \enablel@dtabfeet}
9978
9979 %

```

\ctab Array with centered entries.

```

\edbeforetab
\edaftertab
9980 \newcommand{\ctab}[1]{%
9981   \l@dnnullfills
9982   \def\edbeforetab##1##2{\leftctab{##1}{##2}}%
9983   \def\edaftertab##1##2{\rightctab{##1}{##2}}%
9984   \measuretbody{#1}%
9985   \l@drestorefills
9986   \variab
9987   \setmrowcenter #1\&\&\&%
9988   \enablel@dtabfeet}
9989
9990 %

```

\ctabtext Tabular with entries centered.

```

9991 \newcommand{\ctabtext}[1]{%
9992   \l@dnnullfills
9993   \measuretbody{#1}%
9994   \l@drestorefills
9995   \variab
9996   \settrrowcenter #1\&\&\&%
9997   \enablel@dtabfeet}
9998
9999 %

```

```

\spreadtext
10000 \newcommand{\spreadtext}[1]{%\l@dcolcount=\l@dampcount%
10001   \hb@xt@ \the\l@dcolwidth{\hbox{#1}\hss}}
10002 %

```

```

\spreadmath
10003 \newcommand{\spreadmath}[1]{%
10004   \hb@xt@ \the\l@dcolwidth{\hbox{$\displaystyle{#1}$}\hss}}
10005
10006 %

```

\HILFSskip More helpers.

```

\Hilfsskip
10007 \newskip\HILFSskip
10008 \newskip\Hilfsskip
10009
10010 %

```

```

\EDTABINDENT \newcommand{\EDTABINDENT}{%
10011
10012   \ifnum\l@dcolcount=30\let\NEXT\relax\l@dcolcount=0%
10013   \else\step1@dcolcount%
10014     \advance\Hilfsskip by\l@dcolwidth%
10015     \ifdim\l@dcolwidth=0pt\advance\hilfscout\@ne
10016     \else\advance\Hilfsskip by \the\hilfscout\edtabcolsep%
10017     \hilfscout=1\fi%
10018     \let\NEXT=\EDTABINDENT%
10019   \fi\NEXT}%
10020 %

```

\edtabindent (was \tabindent)

```

10021 \newcommand{\edtabindent}{%
10022   \l@dcolcount=0\relax
10023   \Hilfsskip=0pt%
10024   \hilfscout=1\relax
10025   \EDTABINDENT%
10026   \hilfsskip=\hsize%
10027   \advance\hilfsskip -\Hilfsskip%
10028   \Hilfsskip=0.5\hilfsskip%
10029   }%
10030 %
10031 %

```

\EDTAB (was \TAB)

```

10032 \def\EDTAB #1|#2|{%
10033   \setbox\tabhilfbox=\hbox{$\displaystyle{#1}$}%
10034   \setbox\tabHilfbox=\hbox{$\displaystyle{#2}$}%
10035   \advance\tabelskip -\wd\tabhilfbox%
10036   \advance\tabelskip -\wd\tabHilfbox%
10037   \unhbox\tabhilfbox\hskip\tabelskip%
10038   \unhbox\tabHilfbox}%
10039 %
10040 %

```

\EDTABtext (was \TABtext)

```

10041 \def\EDTABtext #1|#2|{%
10042   \setbox\tabhilfbox=\hbox{#1}%
10043   \setbox\tabHilfbox=\hbox{#2}%
10044   \advance\tabelskip -\wd\tabhilfbox%
10045   \advance\tabelskip -\wd\tabHilfbox%
10046   \unhbox\tabhilfbox\hskip\tabelskip%
10047   \unhbox\tabHilfbox}%
10048 %

```

\tabhilfbox Further helpers.

```

\-tabHilfbox
10049 \newbox\tabhilfbox
10050 \newbox\tabHilfbox
10051
10052 %

```

XXXII.2.4 Environments

edarrayl edarrayc edarrayr The environment forms for \ltab, \ctab and \rtab.

```
10053 \newenvironment{edarrayl}{\l@dcollect@body\ltab}{}
10054 \newenvironment{edarrayc}{\l@dcollect@body\ctab}{}
10055 \newenvironment{edarrayr}{\l@dcollect@body\rtab}{}
10056
10057 %
```

edtabularl edtabularc edtabularr The environment forms for \ltabtext, \ctabtext and \rtabtext.

```
10058 \newenvironment{edtabularl}{\l@dcollect@body\ltabtext}{}
10059 \newenvironment{edtabularc}{\l@dcollect@body\ctabtext}{}
10060 \newenvironment{edtabularr}{\l@dcollect@body\rtabtext}{}
10061
10062 %
```

XXXIII Quotation's commands

`\initnumbering@quote` This macro, called at the beginning of any numbered section, locally redefines the quotation and quote environments, in order to allow their use inside of numbered sections.

```
\quotation \initnumbering@quote defines quotation environment.
\endquotation
\quote      10063 \newcommand{\initnumbering@quote}{
10064 \ifnoquotation@else
\endquote   10065 \renewcommand{\quotation}{\par\leavevmode%
10066 \parindent=1.5em%
10067 \skipnumbering%
10068 \ifautopar%
10069 \vskip-\parskip%
10070 \else%
10071 \vskip\topsep%
10072 \fi%
10073 \global\leftskip=\leftmargin%
10074 \global\rightskip=\leftmargin%
10075 }
10076 \renewcommand{\endquotation}{\par%
10077 \global\leftskip=0pt%
10078 \global\rightskip=0pt%
10079 \leavevmode%
10080 \skipnumbering%
10081 \ifautopar%
10082 \vskip-\parskip%
10083 \else%
10084 \vskip\topsep%
10085 \fi%
10086 }
10087 \renewcommand{\quote}{\par\leavevmode%
10088 \parindent=0pt%
10089 \skipnumbering%
10090 \ifautopar%
```

```

10091         \vskip-\parskip%
10092     \else%
10093         \vskip\topsep%
10094     \fi%
10095     \global\leftskip=\leftmargin%
10096     \global\rightskip=\leftmargin%
10097 }
10098 \renewcommand{\endquote}{\par%
10099     \global\leftskip=0pt%
10100     \global\rightskip=0pt%
10101     \leavevmode%
10102     \skipnumbering%
10103     \ifautopar%
10104         \vskip-\parskip%
10105     \else%
10106         \vskip\topsep%
10107     \fi%
10108 }
10109 \fi
10110 }
10111 %

```

XXXIV Section's title commands

XXXIV.1 Commands to disable some feature

`\ledsectnotoc` The `\ledsectnotoc` only disables the `\addcontentsline` macro.

```

10112 \newcommand{\ledsectnotoc}{\let\addcontentsline@gobblethree}
10113 %

```

`\ledsectnomark` The `\ledsectnomark` only disables the `\chaptermark`, `\sectionmark` and `\subsectionmark` macros.

```

10114 \newcommand{\ledsectnomark}{%
10115     \let\chaptermark@gobble%
10116     \let\sectionmark@gobble%
10117     \let\subsectionmark@gobble%
10118 }
10119 %

```

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).

```
10120 \catcode`\#=12
10121 \notbool{@noeled@sec}{%
10122 %
```

`\beforeeledchapter` For technical reasons, not yet solved, page-breaking before chapters can't be made automatically by `eledmac`. Users have to use `\beforeeledchapter`.

```
10123 \ifl@dmemoir
10124   \newcommand\beforeeledchapter{%
10125     \clearforchapter%
10126   }
10127 \else
10128   \newcommand\beforeeledchapter{%
10129     \if@openright%
10130       \cleardoublepage%
10131     \else%
10132       \clearpage%
10133     \fi%
10134   }
10135 \fi
10136 %
```

XXXIV.4 Auxiliary commands

`\leftmargin@eledsection` `\print@leftmargin@eledsection` and `\print@rightmargin@eledsection` are added by `reledmac` inside the code of sectioning command, in order to affix lines numbers. They include tests for RTL languages.

```

10137 \def\print@rightmargin@eledsection{%
10138   \if@eled@sectioning%
10139     \begingroup%
10140     \if@RTL%
10141       \let\llap\rlap%
10142       \let\leftlinenum\rightlinenum%
10143       \let\leftlinenumR\rightlinenumR%
10144       \let\l@drd@ta\l@dld@ta%
10145       \let\l@drsn@te\l@dlsn@te%
10146     \fi%
10147     \hfill\l@drd@ta \csuse{LR}{\l@drsn@te}%
10148   \endgroup%
10149 \fi%
10150 }%
10151
10152 \def\print@leftmargin@eledsection{%
10153   \if@eled@sectioning%
10154     \leavevmode%
10155     \begingroup%
10156     \if@RTL%
10157       \let\rlap\llap%
10158       \let\rightlinenum\leftlinenum%
10159       \let\rightlinenumR\leftlinenumR%
10160       \let\l@dld@ta\l@drd@ta%
10161       \let\l@dlsn@te\l@drsn@te%
10162     \fi%
10163     \l@dld@ta\csuse{LR}{\l@dlsn@te}%
10164   \endgroup%
10165 \fi%
10166 }%
10167
10168 %

```

XXXIV.5 Patching standard commands

`\M@ssect`
`\@mem@old@ssect`
`\@makechapterhead`
`\@makechapterhead`
`\@makeschapterhead`
`\@ssect`
`\@ssect`

We have to patch \LaTeX , book and memoir sectioning commands in order to:

- Disable `\edtext` inside.
- Disable page breaking (for `\chapter`).
- Add line numbers and sidenotes.

Unfortunately, Maïeul Rouquette was not able to try if memoir is loaded. That is why `eledmac` tries to define for both standard class and memoir class.

```

10169 \AtBeginDocument{%
10170
10171
10172 \pretocmd{\M@ssect}
10173   {\let\old@edtext=\edtext%
10174    \let\edtext=\dummy@edtext@showlemma%
10175   }
10176 {}
10177 {}

```

```

10178
10179 \apptocmd{\M@sect}
10180 {\let\edtext=\old@edtext}
10181 {}
10182 {}
10183
10184 \patchcmd{\M@sect}
10185 { #9}
10186 { #9%
10187 \print@rightmargin@eledsection%
10188 }
10189 {}
10190 {}
10191
10192 \patchcmd{\M@sect}
10193 {\hskip #3\relax}
10194 {\hskip #3\relax%
10195 \print@leftmargin@eledsection%
10196 }
10197 {}
10198 {}
10199
10200 \patchcmd{\@mem@old@ssect}
10201 {#5}
10202 {#5%
10203 \print@leftmargin@eledsection%
10204 }
10205 {}
10206 {}
10207
10208 \patchcmd{\@mem@old@ssect}
10209 {\hskip #1}
10210 {\hskip #1%
10211 \print@rightmargin@eledsection%
10212 }
10213 {}
10214 {}
10215
10216
10217
10218 \patchcmd{\scr@startchapter}{\if@openright\cleardoublepage\else\clearpage\fi}{%
10219 \if@eled@sectioning\else%
10220 \ifl@dprintingpages\else%
10221 \if@openright\cleardoublepage\else\clearpage\fi}%No clearpage inside a
10222 \fi%
10223 \fi%
10224 }
10225 {}
10226 {}
10227
10228 \patchcmd{\@makechapterhead}

```

```

10229 {#1}
10230 {\print@leftmargin@eledsection%
10231   #1%
10232   \print@rightmargin@eledsection%
10233 }
10234 {}
10235 {}
10236
10237 \patchcmd{\@makechapterhead}% For BIDI
10238   {\if@RTL\raggedleft\else\raggedright\fi}%
10239   {\if@eled@sectioning\else%
10240     \if@RTL\raggedleft\else\raggedright\fi%
10241     \fi}%
10242   }%
10243   {}%
10244   {}%
10245
10246 \patchcmd{\@makeschapterhead}
10247   {#1}
10248   {\print@leftmargin@eledsection%
10249     #1%
10250     \print@rightmargin@eledsection%
10251   }
10252   {}
10253   {}
10254
10255 \pretocmd{\@sect}
10256   {\let\old@edtext=\edtext
10257     \let\edtext=\dummy@edtext@showlemma%
10258   }
10259   {}
10260   {}
10261
10262 \apptocmd{\@sect}
10263   {\let\edtext=\old@edtext}
10264   {}
10265   {}
10266
10267 \pretocmd{\@ssect}
10268   {\let\old@edtext=\edtext%
10269     \let\edtext=\dummy@edtext@showlemma%
10270   }
10271   {}
10272   {}
10273
10274 \apptocmd{\@ssect}
10275   {\let\edtext=\old@edtext}
10276   {}
10277   {}
10278
10279 %

```

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 com-

mands.

```

10280 \ifdef{\H@old@sect}{%
10281
10282   \patchcmd{\H@old@sect}%
10283     {#8}
10284     {#8%
10285       \print@rightmargin@eledsection%
10286     }
10287     {}
10288     {}
10289
10290   \patchcmd{\H@old@sect}%
10291     {\hskip #3\relax}
10292     {\hskip #3\relax%
10293       \print@leftmargin@eledsection%
10294     }
10295     {}
10296     {}
10297
10298   \patchcmd{\H@old@@sect}%
10299     {#5}
10300     {#5%
10301       \print@rightmargin@eledsection%
10302     }
10303     {}
10304     {}
10305
10306   \patchcmd{\H@old@@@sect}%
10307     {\hskip #1}
10308     {\hskip #1%
10309       \print@leftmargin@eledsection%
10310     }
10311     {}
10312     {}
10313 }%
10314 {
10315   \patchcmd{\@sect}
10316     {#8}
10317     {#8%
10318       \print@rightmargin@eledsection%
10319     }
10320     {}
10321     {}
10322
10323   \patchcmd{\@sect}
10324     {\hskip #3\relax}
10325     {\hskip #3\relax%
10326       \print@leftmargin@eledsection%
10327     }
10328     {}
10329     {}
10330
10331   \patchcmd{\@ssect}
10332     {#5}

```

```

10333     {#5%
10334     \print@rightmargin@eledsection%
10335     }
10336     {}
10337     {}
10338
10339     \patchcmd{\@ssect}
10340     {\hskip #1}
10341     {\hskip #1%
10342     \print@leftmargin@eledsection%
10343     }
10344     {}
10345     {}
10346     }%
10347 }%
10348 %

```

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 `#`.

```

10349 {}}%
10350 \protect\catcode`\#=6 %Space NEEDS by \catcode
10351 %

```

\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.

```

10352 \AtBeginDocument{%
10353 \patchcmd{\chapter}{\clearforchapter}{%
10354 \if@eled@sectioning\else%
10355 \ifl@dprintingpages\else%
10356 \clearforchapter%
10357 \fi%
10358 \fi%
10359 }%
10360 {}%
10361 {}%
10362
10363 \patchcmd{\chapter}{\if@openright\cleardoublepage\else\clearpage\fi}{%
10364 \if@eled@sectioning\else%
10365 \ifl@dprintingpages%
10366 \endgraf%
10367 \else%
10368 \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
10369 \fi%
10370 \fi%
10371 }%
10372 {}%
10373 {}%
10374 }%
10375 %

```

`\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.

```
10376 \newif\if@eled@sectioning%
10377 %
```

We reopen a new `\notbool{@noeled@sec}` statement, as we will define the `\elesection` commands.

```
10378 \notbool{@noeled@sec}{%
10379 %
```

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.

```
10380 \newwrite\eled@sectioning@out
10381 %
```

`\eledchapter` And now, the user sectioning commands, which write to the file, and also add content
`\eledsection` as a “normal” line.

```

\eledsubsection
\eledsubsubsection
\eledchapter*
\eledsection*
\eledsubsection*
\eledsubsubsection*
10382 \newcommand{\eledchapter}[2] [] {%
10383   \disable@familiarnotes%
10384   #2%
10385   \restore@familiarnotes%
10386   \ifledRcol%
10387     \immediate\write\eled@sectioningR@out{%
10388       \string\eled@chapter{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR}{-}{R}
10389     }%
10390   \else%
10391     \immediate\write\eled@sectioning@out{%
10392       \string\eled@chapter{#1}{\unexpanded{#2}}{\the\pstarts@read@L}{-}{-}%
10393     }%
10394   \fi%
10395 }
10396
10397 \newcommand{\eledsection}[2] [] {%
10398   \disable@familiarnotes%
10399   #2%
10400   \restore@familiarnotes%
10401   \ifledRcol%
10402     \immediate\write\eled@sectioningR@out{%
10403       \string\eled@section{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR}{-}{R}
10404     }%
10405   \else%
10406     \immediate\write\eled@sectioning@out{%
10407       \string\eled@section{#1}{\unexpanded{#2}}{\the\pstarts@read@L}{-}{-}%
10408     }%
10409   \fi%
10410 }
10411
```

```

10412 \newcommand{\eledsubsection}[2] [] {%
10413   \disable@familiarnotes%
10414   #2%
10415   \restore@familiarnotes%
10416   \ifledRcol%
10417     \immediate\write\eled@sectioningR@out{%
10418       \string\eled@subsection{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR}{R}
10419     }%
10420   \else%
10421     \immediate\write\eled@sectioning@out{%
10422       \string\eled@subsection{#1}{\unexpanded{#2}}{\the\pstarts@read@L}{R}
10423     }%
10424   \fi%
10425 }
10426 \newcommand{\eledsubsubsection}[2] [] {%
10427   \disable@familiarnotes%
10428   #2%
10429   \restore@familiarnotes%
10430   \ifledRcol%
10431     \immediate\write\eled@sectioningR@out{%
10432       \string\eled@subsubsection{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR
10433     }{R}
10434     }%
10435   \else%
10436     \immediate\write\eled@sectioning@out{%
10437       \string\eled@subsubsection{#1}{\unexpanded{#2}}{\the\pstarts@read@L
10438     }{R}}%
10439     }%
10440   \fi%
10441 }
10442 \WithSuffix\newcommand\eledchapter*[2] [] {%
10443   \disable@familiarnotes%
10444   #2%
10445   \restore@familiarnotes%
10446   \ifledRcol%
10447     \immediate\write\eled@sectioningR@out{%
10448       \string\eled@chapter{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR}{*}{R}
10449     }%
10450   \else%
10451     \immediate\write\eled@sectioning@out{%
10452       \string\eled@chapter{#1}{\unexpanded{#2}}{\the\pstarts@read@L}{*}{R}
10453     }%
10454   \fi%
10455 }
10456 \WithSuffix\newcommand\eledsection*[2] [] {%
10457   \disable@familiarnotes%
10458   #2%
10459   \restore@familiarnotes%
10460   \ifledRcol%
10461

```



```

10462 \immediate\write\eled@sectioningR@out{%
10463 \string\eled@section{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR}{*}{R}
10464 }%
10465 \else%
10466 \immediate\write\eled@sectioning@out{%
10467 \string\eled@section{#1}{\unexpanded{#2}}{\the\pstarts@read@L}{*}{}%
10468 }%
10469 \fi%
10470 }
10471
10472 \WithSuffix\newcommand\eledsubsection*[2] [] {%
10473 \disable@familiarnotes%
10474 #2%
10475 \restore@familiarnotes%
10476 \ifledRcol%
10477 \immediate\write\eled@sectioningR@out{%
10478 \string\eled@subsection{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR}{*}{R}
10479 }%
10480 \else%
10481 \immediate\write\eled@sectioning@out{%
10482 \string\eled@subsection{#1}{\unexpanded{#2}}{\the\pstarts@read@L
10483 }{*}{}%
10484 }%
10485 \fi%
10486 }
10487
10488 \WithSuffix\newcommand\eledsubsubsection*[2] [] {%
10489 \disable@familiarnotes%
10490 #2%
10491 \restore@familiarnotes%
10492 \ifledRcol%
10493 \immediate\write\eled@sectioningR@out{%
10494 \string\eled@subsubsection{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR
10495 }{*}{R}
10496 }%
10497 \else%
10498 \immediate\write\eled@sectioning@out{%
10499 \string\eled@subsubsection{#1}{\unexpanded{#2}}{\the\pstarts@read@L
10500 }{*}{}%
10501 }%
10502 \fi%
10503 }
10504 %

```

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.

```

10502 \def\eled@chapter#1#2#3#4#5{%
10503     \ifstrempy{#4}%
10504     {%
10505         \ifstrempy{#1}%
10506         {%
10507             \csgdef{eled@sectioning@#3#5}{\let\edtext=\dummy@edtext@showlemma\
chapter{#2}}}%
10508             \csgdef{eled@sectmark@#3#5}{\let\edtext=\dummy@edtext{}\chaptermark
{#2}}}%
10509             }%Need for \pairs, because of using parbox.
10510             {%
10511                 \csgdef{eled@sectioning@#3#5}{\let\edtext=\dummy@edtext@showlemma\
chapter[#1]{#2}}}%
10512                 \csgdef{eled@sectmark@#3#5}{\let\edtext=\dummy@edtext{}\chaptermark
{#2}}}%Need for \pairs, because of using parbox.
10513                 }%
10514             }%
10515             {%
10516                 \ifstrempy{#1}%
10517                 {\csgdef{eled@sectioning@#3#5}{\let\edtext=\dummy@edtext@showlemma\
chapter*{#2}}}%
10518                 {\csgdef{eled@sectioning@#3#5}{\let\edtext=\dummy@edtext@showlemma\
chapter*{#1}{#2}}}%Bug in LaTeX!
10519                 }%
10520                 \listcsgadd{eled@sections#5@#}{#3}%
10521                 }
10522 \def\eled@section#1#2#3#4#5{%
10523     \ifstrempy{#4}%
10524     {\ifstrempy{#1}%
10525         {%
10526             \csgdef{eled@sectioning@#3#5}{\section{#2}}}%
10527             \csgdef{eled@sectmark@#3#5}{\let\edtext=\dummy@edtext{}\sectionmark
{#2}}}%Need for \pairs, because of using parbox.
10528             }%
10529             {%
10530                 \csgdef{eled@sectioning@#3#5}{\section[#1]{#2}}}%
10531                 \csgdef{eled@sectmark@#3#5}{\let\edtext=\dummy@edtext{}\sectionmark
{#1}}}%Need for \pairs, because of using parbox.
10532                 }%
10533             }%
10534             {\ifstrempy{#1}%
10535                 {\csgdef{eled@sectioning@#3#5}{\section*{#2}}}%
10536                 {\csgdef{eled@sectioning@#3#5}{\section*{#1}{#2}}}%Bug in LaTeX!
10537                 }
10538                 \listcsgadd{eled@sections#5@#}{#3}%
10539                 }
10540 \def\eled@subsection#1#2#3#4#5{%
10541     \ifstrempy{#4}%
10542     {\ifstrempy{#1}%
10543         {%

```

```

10544 \csgdef{eled@sectioning@#3#5}{\subsection{#2}}%
10545 \csgdef{eled@sectmark@#3#5}{\let\edtext=\dummy@edtext{}\csuse{
subsectionmark}{#2}}%Need for \pairs, because of using parbox. \csuse in
case of \subsectionmark is not defined (book)
10546 }%
10547 {%
10548 \csgdef{eled@sectioning@#3#5}{\subsection[#1]{#2}}%
10549 \csgdef{eled@sectmark@#3#5}{\let\edtext=\dummy@edtext{}\csuse{
subsectionmark}{#1}}%Need for \pairs, because of using parbox. \csuse in
case of \subsectionmark is not defined (book)
10550 }%
10551 }%
10552 {\ifstrempy{#1}%
10553 {\csgdef{eled@sectioning@#3#5}{\subsection*{#2}}}%
10554 {\csgdef{eled@sectioning@#3#5}{\subsection*{#1}{#2}}}%Bug in LaTeX!
10555 }
10556 \listcsgadd{eled@sections#5@@}{#3}%
10557 }
10558 \def\eled@subsubsection#1#2#3#4#5{%
10559 \ifstrempy{#4}%
10560 {\ifstrempy{#1}%
10561 {\csgdef{eled@sectioning@#3#5}{\subsubsection{#2}}}%
10562 {\csgdef{eled@sectioning@#3#5}{\subsubsection[#1]{#2}}}%
10563 }%
10564 {\ifstrempy{#1}%
10565 {\csgdef{eled@sectioning@#3#5}{\subsubsection*{#2}}}%
10566 {\csgdef{eled@sectioning@#3#5}{\subsubsection*{#1}{#2}}}%Bug in
LaTeX!
10567 }
10568 \listcsgadd{eled@sections#5@@}{#3}%
10569 }
10570
10571 %

End of the conditional test about noeledsec option.

10572 }{}
10573 %

```

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.

```

10574 \def\normal@page@break{}
10575 %

```

`\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.

```
10576 \def\l@prev@pb{}
10577 \def\l@prev@nopb{}
10578 %
```

`\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.

```
10579 \newcommand{\ledpb}{\write\linenum@out{\string\led@pb}}
10580 \newcommand{\ledpbnum}[1]{\write\linenum@out{\string\led@pbnum{#1}}}
10581 \newcommand{\lednopb}{\write\linenum@out{\string\led@nopb}}
10582 \newcommand{\lednopbnum}[1]{\write\linenum@out{\string\led@nopbnum{#1}}}
10583 %
```

`\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.

```
10584 \newcommand{\led@pb}{\listxadd{\l@prev@pb}{\the\absline@num}}
10585 \newcommand{\led@pbnum}[1]{\listxadd{\l@prev@pb}{#1}}
10586 \newcommand{\led@nopb}{\listxadd{\l@prev@nopb}{\the\absline@num}}
10587 \newcommand{\led@nopbnum}[1]{\listxadd{\l@prev@nopb}{#1}}
10588 %
```

`\ledpbsetting` The `\ledpbsetting` macro only changes the value of `\led@pb@macro`, for which the default value is before.

```
10589 \def\led@pb@setting{before}
10590 \newcommand{\ledpbsetting}[1]{\gdef\led@pb@setting{#1}}
10591 %
```

`\led@check@pb` The `\led@check@pb` and `\led@check@nopb` are called before or after each line. They check if a page break must occur, depending on the current line and on the content of `\l@pb`.

```
10592 \newcommand{\led@check@pb}{\xifinlist{\the\absline@num}{\l@prev@pb}{\pagebreak[4]}}
10593 \newcommand{\led@check@nopb}{%
10594   \IfStrEq{\led@pb@setting}{before}{%
10595     \xifinlist{\the\absline@num}{\l@prev@nopb}%
10596     {\numdef{\abs@prevline}{\the\absline@num-1}%
10597     \xifinlist{\abs@prevline}{\normal@page@break}%
10598     {\nopagebreak[4]\enlargethispage{\baselineskip}}%
10599     {}}%
10600   }%
10601 }%
10602 }%
10603 \IfStrEq{\led@pb@setting}{after}{%
10604   \xifinlist{\the\absline@num}{\l@prev@nopb}{%

```

```

10605 \xifinlist{\the\absline@num}{\normal@page@break}%
10606 {\nopagebreak[4]\enlargethispage{\baselineskip}}}%
10607 {}%
10608 }%
10609 {}%
10610 {}%
10611 {}%
10612 }
10613 %

```

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.

```

10614 \newcommand{\check@pb@in@verse}{%
10615 \ifinstanza\iflednopbinverse\ifinserthangingsymbol% Using stanzas and
enabling page breaks in verse control, while on a hanging verse.
10616 \ifnum\page@num=\last@page@num\else%If we have change page
10617 \IfStrEq{\led@pb@setting}{before}{%
10618 \numgdef{\abs@line@verse}{\the\absline@num-1}%
10619 \ledpbnum{\abs@line@verse}%
10620 }{}%
10621 \IfStrEq{\led@pb@setting}{after}{%
10622 \numgdef{\abs@line@verse}{\the\absline@num-1}%
10623 \lednopbnum{\abs@line@verse}%
10624 }{}%
10625 \fi%
10626 \fi\fi\fi%
10627 }
10628 %

```

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.³⁸

³⁸<http://tex.stackexchange.com/a/17138/7712>.

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.³⁹

```

10629 \def\Hy@raisedlink@left#1{%
10630   \ifvmode
10631     #1%
10632   \else
10633     \Hy@SaveSpaceFactor
10634     \llap{\smash{%
10635       \begin{group}
10636         \let\HyperRaiseLinkLength\@tempdima
10637         \setlength\HyperRaiseLinkLength\HyperRaiseLinkDefault
10638         \HyperRaiseLinkHook
10639       \expandafter\endgroup
10640       \expandafter\raise\the\HyperRaiseLinkLength\hbox{%
10641         \Hy@RestoreSpaceFactor
10642         #1%
10643         \Hy@SaveSpaceFactor
10644       }%
10645     }}%
10646     \Hy@RestoreSpaceFactor
10647     \penalty\@M\hskip\z@\relax
10648   \fi
10649 }
10650 %

```

XXXVIII Compatibility with eledmac

Here, we define some commands for the `eledmac-compat` option.

```

10651 \ifeledmaccompat@%
10652
10653   \newcommand{\footnormalX}[1]{\arrangementX[#1]{normal}}%
10654   \newcommand{\footparagraphX}[1]{\arrangementX[#1]{paragraph}}%
10655   \newcommand{\foottwocolX}[1]{\arrangementX[#1]{twocol}}%
10656   \newcommand{\footthreecolX}[1]{\arrangementX[#1]{threecol}}%
10657
10658   \unless\ifnocritical@
10659     \newcommand{\footnormal}[1]{\Xarrangement[#1]{normal}}%
10660     \newcommand{\footparagraph}[1]{\Xarrangement[#1]{paragraph}}%
10661     \newcommand{\foottwocol}[1]{\Xarrangement[#1]{twocol}}%
10662     \newcommand{\footthreecol}[1]{\Xarrangement[#1]{threecol}}%
10663     \let\hsizetwocol\Xhsizetwocol
10664     \let\hsizethreecol\Xhsizethreecol
10665     \let\bhookXnote\Xbhooknote
10666     \let\boxsymlinenum\Xboxsymlinenum
10667     \let\symlinenum\Xsymlinenum
10668     \let\beforenumberinfootnote\Xbeforenumber
10669     \let\afternumberinfootnote\Xafternumber

```

³⁹The 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>.

```

10670 \let\beforeXsymlinenum\Xbeforesymlinenum
10671 \let\afterXsymlinenum\Xaftersymlinenum
10672 \let\inplaceofnumber\Xinplaceofnumber
10673 \let\Xlemmaseparator\lemmaseparator
10674 \let\afterlemmaseparator\Xafterlemmaseparator
10675 \let\beforelemmaseparator\Xbeforelemmaseparator
10676 \let\inplaceoflemmaseparator\Xinplaceoflemmaseparator
10677 \let\txbeforeXnotes\Xtxbeforenotes
10678 \let\afterXrule\Xafterrule
10679 \let\numberonlyfirstinline\Xnumberonlyfirstinline
10680 \let\numberonlyfirstintwlines\Xnumberonlyfirstintwlines
10681 \let\nonumberinfootnote\Xnonumberinfootnote
10682 \let\pstartinfootnote\Xpstart
10683 \let\pstartinfootnoteeverytime\Xpstarteverytime
10684 \let\onlyXpstart\Xonlypstart
10685 \let\Xnonumberinfootnote\Xnonumber
10686 \let\nonbreakableafternumber\Xnonbreakableafternumber
10687 \let\maxhXnotes\Xmaxhnotes
10688 \let\beforeXnotes\Xbeforenotes
10689 \let\boxlinenum\Xboxlinenum
10690 \let\boxlinenumalign\Xboxlinenumalign
10691 \let\boxstartlinenum\Xboxstartlinenum
10692 \let\boxendlinenum\Xboxendlinenum
10693 \let\twlines\Xtwlines
10694 \let\morethantwlines\Xmorethantwlines
10695 \let\twlinesbutnotmore\Xtwlinesbutnotmore
10696 \let\twlinesonlyinsamepage\Xtwlinesonlyinsamepage
10697 \fi
10698
10699 \unless\ifnofamiliar@
10700 \let\notesXwidthliketwocolumns\noteswidthliketwocolumnsX
10701 \fi
10702 \newcommandx{\parafootsep}[2][1,usedefault]{%
10703 \Xparafootsep[#1]{#2}%
10704 \parafootsepX[#1]{#2}
10705 }%
10706
10707 \newcommandx{\afternote}[2][1,usedefault]{%
10708 \Xafternote[#1]{#2}%
10709 \afternoteX[#1]{#2}%
10710 }%
10711
10712 \unless\ifnoend@
10713 \let\XendXtwlines\Xendtwlines
10714 \let\XendXmorethantwlines\Xendmorethantwlines
10715 \let\XhookXendnote\Xendhooknote
10716 \let\boxXendlinenum\Xendboxlinenum%
10717 \let\boxXendlinenumalign\Xendboxlinenumalign%
10718 \let\boxXendstartlinenum\Xendboxstartlinenum%
10719 \let\boxXendendlinenum\Xendboxendlinenum%
10720 \let\XendXlemmaseparator\Xendlemmaseparator
10721 \let\XendXbeforelemmaseparator\Xendbeforelemmaseparator
10722 \let\XendXafterlemmaseparator\Xendafterlemmaseparator
10723 \let\XendXinplaceoflemmaseparator\Xendinplaceoflemmaseparator

```

```
10724 \fi
10725
10726 \AtBeginDocument{%
10727   \ifdef\lineref{}\let\lineref\edlineref}%
10728 }%
10729
10730
10731 \fi%
10732 %
```

</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 changed⁴⁰ to `\critext`; use the `\edtext` macro instead. However, if you do use `\critext` (the new name for `\text`), the following is a reminder.

`\critext` Within numbered paragraphs, footnotes and endnotes are generated by forms of the `\critext` macro:

```
\critext{⟨lemma⟩}{⟨commands⟩/}
```

The `⟨lemma⟩` argument is the lemma in the main text: `\critext` both prints this as part of the text, and makes it available to the `⟨commands⟩` you specify to generate notes. The `/` at the end terminates the command; it is part of the macro's definition so that spaces after the macro will be treated as significant.

For example:

<pre>I saw my friend \critext{Smith} \Afootnote{Jones C, D.}/ on Tuesday.</pre>	<pre>1 I saw my friend 2 Smith on Tuesday. 2 Smith] Jones C, D.</pre>
---	---

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

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

<pre>\critext{I saw my friend \critext{Smith}{\Afootnote{Jones C, D.}/ on Tuesday.} \Bfootnote{The date was July 16, 1954.} /}</pre>	<pre>1 I saw my friend 2 Smith on Tuesday. 2 Smith] Jones C, D. 1–2 I saw my friend Smith on Tuesday.] The date was July 16, 1954.</pre>
--	--

However, `\critext` cannot handle overlapping but unnested notes—for example, one note covering lines 10–15, and another covering 12–18; a `\critext` that starts in the `⟨lemma⟩` argument of another `\critext` must end there, too. (The `\lemma` and `\linenum` commands may be used to generate overlapping notes if necessary.)

The second argument of the `\critext` macro, `⟨commands⟩`, is the same as the second argument to the `\edtext` macro.

It is possible to define aliases for `\critext`, which can be easier to type. You can make a single character substitute for `\critext` by saying this:

```
\catcode'\<=\active
\let<=\critext
```

⁴⁰A name like `\text` is likely to be defined by other \LaTeX packages (it certainly is by the AMS packages) and it seems sensible to try and avoid clashes with other definitions.

Then you might say `<{Smith}\variant{Jones}/`. This of course destroys the ability to use `<` in any new macro definitions, so long as it remains in effect; hence it should be used with care.

Changing the character at the end of the command requires more work:

```
\catcode'\<=\active
\def\xtext#1#2>{\critext{#1}{#2}/}
\let<=\xtext
```

This allows you to say `<{Smith}\Afootnote{Jones}>`.

Aliases for `\critext` of the first kind shown here also can't be nested—that is, you can't use the alias in the text that forms the first argument to `\critext`. (See VI p. 130 to find out why.) Aliases of the second kind may be nested without any problem.

If you really have to use `\critext` in any of the tabular or array environments, then `\edtext` must not be used in the same environment. If you use `\critext` in one of these environments then you have to issue the declaration `\usingcritext` beforehand. The declaration `\usingedtext` must be issued to revert to the default assumption that `\edtext` will be used.

A.2 Migration from ledmac to eledmac

In `eledmac`, some changes were made in the code to allow easy customization. This may cause problems for people who have already made their own. The next sections explain how to handle this.

If you have created your own series using `\addfootins` and `\addfootinsX`, you must use instead the `\newseries` command (see 6.7.1 p. 39), and remove any `\Xfootnote` command.

If you have customized the `\XXXXXXfmt` command, please check whether you can achieve the same by the commands documented for display options (7 p. 40) or `\Xfootnote` options (6.2.2 p. 28). Otherwise please add a new ticket on Github to request a new function for doing this.⁴¹

If for some reason you do not want to make the modifications to use the new functions of `eledmac`, you can continue using your own `\XXXXXXfmt` command, but you must replace:

```
\renewcommand*{XXXXfmt}[3]
```

with

```
\renewcommandx*{XXXXfmt}[4][4=Z]
```

If you do not make that, you will get a spurious `[X]`, where `X` is series letter.

If you used a `\protect` command inside a `\footnote` command inside a numbered section, you must change the `\protect` to `\noexpand`. Otherwise the command after the `\protect` will be discarded.

⁴¹<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 introduce a better way to handle sectional divisions inside numbered text. Please read 16.2 p. 71.

A.5 Migration to *eledmac* 17.1

This version changes the default setting of `\Xpstart`. Henceforth, `pstart` numbers will be printed in footnotes within the section of text where you have called `\numberpstarttrue`.

We do not see any reason to print them in the other sections. However, if you want to print the `\pstart` numbers in every footnote, whatever the section, without having to use `\numberpstarttrue`, you can use `\Xpstarteverytime`.

A.6 Migration to eledmac 1.21.0

A.6.1 `\Xledsetnormalparstuff` and `\ledsetnormalparstuffX`

The `\ledsetnormalparstuff` has been split into two different commands:

- `\Xledsetnormalparstuff` for critical notes;
- `\ledsetnormalparstuffX` for familiar notes.

Both commands can take an optional argument which is the series letter. If you have redefined `\ledsetnormalparstuff` or any of the commands which call them, you must change them accordingly.

A.6.2 Endnotes

In any case, delete the `.end` file before the next run.

The previous version of Eledmac had a bug: there were two spaces between the starting page number and the starting line number, but only one space between the ending page number and the ending line number.

As a matter of fact, a spurious space was added after the first `\printnpnum`. This spurious space has been deleted. However, if you want to keep the previous spurious space, you may load the package with the `oldprintnpnumspace` option.

If you have redefined `\endprint`, you must:

- Contact us and ask for the feature that required your hack, in order to avoid such a hack in the future.
- Use the new fifth argument.
- Add `\xdef\@currentseries{#4}` at the beginning of your own command.

A.7 Migration to eledmac 1.22.0

The `\ledinnote` command now takes a first optional argument, which is the label for the hyperreference. If you have redefined it, change your redefinition, and check whether you can avoid this redefinition by only redefining `\ledinnotemark`.

A.8 Migration to eledmac 1.23.0

You must delete the numbered auxiliary files before compiling with the new version of eledmac.

A.9 Migration from eledmac to reledmac

There are many changes in reledmac which require the user to make modifications.

A.9.1 Risk of ‘no room for a new’

The risk to obtain a ‘no room for a new something’ error is greater in reledmac than it is in eledmac. See 19.1.3 p. 74 in order to know how to limit it.

A.9.2 Multiple indices with memoir

Eledmac and ledmac used the specific indexing tools of the memoir class designed to produce multiple indices. However, eledmac could also use imakeidx or indextools tools independently of the memoir class. This system forced to maintain redundant code. Since reledmac, we use only the imakeidx or indextools tools.

Consequently: Users of memoir are invited to use indextool or imakeidx to produce multiple indices.

A.9.3 Deprecated commands and options

The table of deprecated commands and their alternatives follows. Note that the way some commands must be used may have changed. Please read the handbook.

<i>Deprecated command</i>	<i>Replaced with</i>
<code>\addfootins</code>	<code>\newseries</code>
<code>\addfootinsX</code>	<code>\newseries</code>
<code>\critext</code>	<code>\edtext</code>
<code>\falseverse</code>	<code>\newverse</code>
<code>\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>noendnotes</code>
<code>\pageparbreak</code>	<code>\ledpb</code>

The `ledsecnolinenum` option has been removed, because it was related to deprecated commands.

The `oldprintnpspace` 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>\&</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>\sidenotesep</code>	<code>\setsidenotesep</code>	
<code>\startstanzahook</code>	Optional argument of <code>\stanza</code>	
<code>\symplinenum</code>	<code>\Xsymplinenum</code>	

A.9.5 Commands the names of which have been changed

In order to help the migration from eledmac to reledmac, you may load reledmac with eledmac-compat option. However, it is advised not to, and to change the command names themselves instead. In many cases, you use only a few of them, except the `\footparagraph` command.

<i>Old command</i>	<i>New command</i>
<code>\footparagraph</code>	<code>\Xarrangement</code>
<code>\footnormal</code>	<code>\Xarrangement</code>
<code>\foottwocol</code>	<code>\Xarrangement</code>
<code>\footthreecol</code>	<code>\Xarrangement</code>
<code>\footparagraphX</code>	<code>\arrangementX</code>
<code>\footnormalX</code>	<code>\arrangementX</code>
<code>\foottwocolX</code>	<code>\arrangementX</code>
<code>\footthreecolX</code>	<code>\arrangementX</code>
<code>\afterlemmaseparator</code>	<code>\Xafterlemmaseparator</code>
<code>\afternote</code>	<code>\Xafternote</code> and <code>\afternoteX</code>
<code>\afternumberinfootnote</code>	<code>\Xafternumber</code>
<code>\afterXrule</code>	<code>\Xafterrule</code>
<code>\afterXsymplinenum</code>	<code>\Xaftersymplinenum</code>
<code>\beforelemmaseparator</code>	<code>\Xbeforelemmaseparator</code>
<code>\beforenumberinfootnote</code>	<code>\Xbeforenumber</code>
<code>\beforeXnotes</code>	<code>\Xbeforenotes</code>
<code>\beforeXsymplinenum</code>	<code>\Xbeforesymplinenum</code>
<code>\bhookXnote</code>	<code>\Xbhookendnote</code>
<code>\bhookXnote</code>	<code>\Xbhooknote</code>
<code>\boxendlinenum</code>	<code>\Xboxendlinenum</code>
<code>\boxlinenum</code>	<code>\Xboxlinenum</code>
<code>\boxlinenumalign</code>	<code>\Xboxlinenumalign</code>
<code>\boxstartlinenum</code>	<code>\Xboxstartlinenum</code>
<code>\boxsymplinenum</code>	<code>\Xboxsymplinenum</code>
<code>\boxXendlinenum</code>	<code>\Xendboxlinenum</code>
<code>\boxXendlinenumalign</code>	<code>\Xendboxlinenumalign</code>
<code>\boxXendstartlinenum</code>	<code>\boxXendstartlinenum</code>
<code>\letboxXendendlinenum</code>	<code>\Xendletboxendlinenum</code>
<code>\hsizetwocol</code>	<code>\Xhsizetwocol</code>
<code>\hsizethreecol</code>	<code>\Xhsizethreecol</code>
<code>\inplaceoflemmaseparator</code>	<code>\Xinplaceoflemmaseparator</code>
<code>\inplaceofnumber</code>	<code>\Xinplaceofnumber</code>
<code>\lemmaseparator</code>	<code>\Xlemmaseparator</code>
<code>\maxhXnotes</code>	<code>\Xmaxhnotes</code>

<i>Old command</i>	<i>New command</i>
<code>\morethantwolines</code>	<code>\Xmorethantwolines</code>
<code>\nonumberinfootnote</code>	<code>\Xnonumber</code>
<code>\notesXwidthliketwocolumns</code>	<code>\noteswidthliketwocolumnsX</code>
<code>\noXlemmaseparator</code>	<code>\Xnolemmaseparator</code>
<code>\numberonlyfirstinline</code>	<code>\Xnumberonlyfirstinline</code>
<code>\numberonlyfirstintwolines</code>	<code>\Xnumberonlyfirstintwolines</code>
<code>\nonbreakableafternumber</code>	<code>\Xnonbreakableafternumber</code>
<code>\onlyXpstart</code>	<code>\Xonlypstart</code>
<code>\parafootsep</code>	<code>\Xparafootsep</code> and <code>\parafootsepX</code>
<code>\pstartinfootnote</code>	<code>\Xpstart</code>
<code>\pstartinfootnoteeverytime</code>	<code>\Xpstarteverytime</code>
<code>\symlinenum</code>	<code>\Xsymlinenum</code>
<code>\twolines</code>	<code>\Xtwolines</code>
<code>\twolinesbutnotmore</code>	<code>\Xtwolinesbutnotmore</code>
<code>\twolinesonlyinsamepage</code>	<code>\Xtwolinesonlyinsamepage</code>
<code>\txtbeforeXnotes</code>	<code>\Xtxtbeforenotes</code>
<code>\XendXafterlemmaseparator</code>	<code>\Xendafterlemmaseparator</code>
<code>\XendXbeforelemmaseparator</code>	<code>\Xendbeforelemmaseparator</code>
<code>\XendXinplaceoflemmaseparator</code>	<code>\Xendinplaceoflemmaseparator</code>
<code>\XendXlemmaseparator</code>	<code>\Xendlemmaseparator</code>
<code>\XendXmorethantwolines</code>	<code>\Xendmorethantwolines</code>
<code>\XendXtwolines</code>	<code>\Xendtwolines</code>
<code>\Xnonumberinfootnote</code>	<code>\Xnonumber</code>
<code>\lineref</code>	<code>\edlineref</code>

A.9.6 Endnotes

With reledmac, there is now one auxiliary file for each endnotes set (.Aend, .Bend, .Cend etc.). If you have overridden `\doendnotes` (which you should not have done) you must adapt your code.

A.9.7 Z Series

The ‘Z’ series of notes has been removed. Only five series are provided now by default: A, B, C, D, E.

A.9.8 Internal commands

Users who have overridden internal commands, which is wrong, must adapt according to the following. Or better, they should not override any of such commands and use reledmac options instead.

- If you have modified `\Xfootfmt`, note that the fourth argument is now mandatory.
- `\unvxh` has been replaced with `\Xunvxh` and `\unvxhX` with two mandatory arguments.

A.10 Migration to reledmac 2.1.0

Reledmac 2.1.0 fix some bugs when using `\Xhooknote` and `\hooknoteX` not in order to execute code at the beginning of each notes, but to insert content of at the beginning

of each notes.

People who use these commands to do it, which is not the original idea, must change the following:

1. Horizontal space is no longer automatically added after the content of the `\Xbhooknote/\bhooknoteX` argument. You must include it manually. So instead of `\Xbhooknote{content}`, use `\Xbhooknote{content }.`
2. Indent is no longer automatically added before the content of the `\Xbhooknote/\bhooknoteX` argument. If you want to keep it, add `\indent` to the argument of `\Xbhooknote/\bhooknoteX`.

A.11 Migration to reledmac 2.1.3

Reledmac 2.1.3 fix an historical bug, (style in ledmac 0.7!) which doubled the space before the rules of paragraphed familiar footnotes. Consequently, if you use paragraphed familiar footnotes, you should maybe adapt it, playing with `\beforenotesX`.

A.12 Migration to reledmac 2.3.0

Before reledmac 2.3.0, for typesetting verse, any empty line was considered a paragraph inside verses. Counting empty lines this created breaking verse, hanging verses, and also added spurious vertical spaces. Version 2.3.0 disables paragraph in stanza. If you want vertical space, use the optional argument of `\stanza` or `\endverse`.

A.13 Migration to reledmac 2.4.0

It is not mandatory, but strongly recommended, to change any `\renewcommand{\endashchar}{\langle...}\rangle` to the use of `\Xlinerrangeseparator` or `/` and `\Xendlinerrangeseparator` (7.4.4 p. 43).

A.14 Migration to reledmac 2.5.0

It is strongly recommended to stop redefining `\printnpnum` and to use the hooks documented in 7.4.3 p. 42.

`\xlineref` does not print anymore the side flag (R for right side), because it is incompatible with numerical test. Use `\xflagref` to obtain it.

The `\printlines` and `\printendlines` commands take now an eighth argument, which is the side flag. It is strongly recommended to NEVER redefine these two commands and to use the setting commands instead (or to ask for new setting commands if the actual does not answer to your needs). However, if you have done it, just change your redefinition to have a new argument.

It is strongly recommended to stop redefining `\fullstop` and to use `\Xsublinesesep` 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 `\Serefwthpage`. 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^a was introduced after the `pstart` number (only for normal typesetting, not while typesetting in parallel). The new version of the package has deleted this spurious space. If you consider that it was NOT a spurious space, you should add it manually in your definition of `\thepstart`.

A.22 Migration to reledmac 2.26.0

You must delete your `.aux` file after having upgraded to this new version of reledmac.

A.23 Migration to reledmac 2.27.1

This release fixes spurious space in `\hidenumbering`. If you considered this spurious space as normal, you must insert it manually using.

```
xx\hidenumbering\ xx
```

To get the space after `xx`.

A.24 Migration to reledmac 2.30.0

If you have multiple annotations for the same line, these are now separated with a comma in the margin. You can use `\setlinenumannotationsep` to change the separator.

A.25 Migration to reledmac 2.31.1

If you use `\Xgroupbyline`, the indentation of the notes has been deleted, in order to have the same behavior as for normal critical notes.

Use `\Xparindent` to restore indentation.

A.26 Migration to reledmac 2.32.7

For paragraphed familiar footnotes, the separator between notes was `\Xparafootsep` instead of `\parafootsepX`. This release fixes this bug. You may adapt your settings.

A.27 Migration to reledmac 2.33.0

You should not redefine anymore `\ledinnotemark`, but use instead `\innotemarkX` and `\Xinnotemark`.

If you have personal script which parses for `\ledinnote` or if you redefine `\ledinnote` (but normally, you should not), this macro have new arguments. Look at the code for more details.

Appendix B Auxiliary softwares

This appendix lists some software and tools related to reledmac that may be useful.

B.1 *samewords*

The *samewords* program (Michael Stenskjær Christensen) automatically adds `\sameword` commands (6.3 p. 31) into a `.tex` file.

<https://samewords.readthedocs.io/en/latest/>

B.2 critical-keys for Emacs

The Emacs editor has a package to help inserting `reledmac`'s commands (Juan Manuel Macías Chaín).

<https://gitlab.com/maciaschain/critical-keys>

B.3 critical-marks for Emacs

A set of functions for Emacs that run in the export process from Org Mode to LaTeX and replace a series of simple textual marks with the basic `reledmac`'s commands (Juan Manuel Macías Chaín) <https://gitlab.com/maciaschain/critical-marks>

B.4 Import from TEI

B.5 Import from TEI

There are multiple tools to convert from XML-TEI to *reledmac*. Here is a non exhaustive-list:

- <http://ciham-digital.huma-num.fr/teicat/>
- https://github.com/fizzbucket/tei_transformer
- <https://github.com/TEIC/Stylesheets>
- <http://lombardpress.org/print/>
- <https://github.com/Jean-Baptiste-Camps/TEItoLaTeX>

Please, tell us if you know of other tools.

References

- [Bre96] Herbert Breger. `tabmac`. October 1996. (Available from CTAN in `macros/plain/contrib/tabmac`)
- [Bur01] John Burt. “Typesetting critical editions of poetry”. *TUGboat*, **22**, 4, pp. 353–361, December 2001. (Code available from CTAN in `macros/latex/contrib/poemscol`)
- [Eck03] Matthias Eckermann. *The Parallel-Package*. April 2003. (Available from CTAN in `macros/latex/contrib/parallel`)
- [Fai03] Robin Fairbairns. *footmisc—a portmanteau package for customising footnotes in L^AT_EX*. 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>\&</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>\@docclearpage</code>	1
<code>\@doreinfeetX</code>	1
<code>\@edindex@fornote@</code>	1
<code>\@edindex@hyperref</code>	1
<code>\@edrowfill@</code>	1
<code>\@edtext@level</code>	1
<code>\@emptytoks</code>	1

\@fnpos	1
\@footnotemark	1
\@footnotetext	1
\@getfirstseries	1
\@gobblefour	1
\@gobbleseven	1
\@gobblethree	1
\@h	1
\@hangingsymbol	1
\@iiminipage	1
\@insertstanzanumber	1
\@insidethisline	1
\@k	1
\@l@dttempcnta	1
\@l@dttempcntb	1
\@lab	1
\@led@testifnofoot	1
\@lemma	1
\@line@#num	1
\@linenumannotationsep	1
\@lock	1
\@lopL	1
\@lopR	1
\@makechapterhead	1
\@makeschapterhead	1
\@mem@extranofeet	1
\@mem@old@ssect	1
\@mpfnpos	1
\@msd	1
\@msd@c	1
\@msd@options@iffullpage	1
\@msdata@list	1
\@nl	1
\@nl@reg	1
\@opXfeet	1
\@outputpage	1
\@page	1
\@pend	1
\@pendR	1
\@ref	1
\@ref@reg	1
\@ref@reg@parse	1
\@resetannot	1
\@sect	1
\@series	1
\@set	1
\@sidenotesep	1
\@ssect	1
\@startstanza	1
\@stopmsd	1
\@stopstanza	1
\@sw	1
\@tag	1
\@wredindex	1

\@xloop	1
\@xympar	1
CLASSarticle	74
CLASSbook	74, 362
CLASSmemoir	214, 276–279, 313, 362, 381, 442, 445
CLASSscrbook	445, 446
COMMAND*footnote	75
COMMAND\...\@footnotemark...	217
COMMAND\...d@ta	157
COMMAND\<hook	
@<series	263
COMMAND\<hookname	
<pseudoserries	265, 266
COMMAND\<type	
footfmt	200
COMMAND\@@@line	190
COMMAND\@MM	178, 442
COMMAND\@Rlineflag	315, 443
COMMAND\@SErefprefix	293
COMMAND\@SErefprefixmore	293
COMMAND\@add@	353
COMMAND\@adv	114
COMMAND\@annot	163
COMMAND\@annot@end@print	204
COMMAND\@annot@start@print	204, 208, 245
COMMAND\@apprefprefixmore	293
COMMAND\@apprefprefixsingle	293
COMMAND\@beforeinsertofthisedtext	137
COMMAND\@bsphack	281
COMMAND\@doclearpage	277, 278, 436, 445
COMMAND\@doreinfeetX	445
COMMAND\@dprintingcolumns	442
COMMAND\@edindex@hyperref	315, 316
COMMAND\@edtext@level	133
COMMAND\@esphack	281
COMMAND\@firstofone	35
COMMAND\@fnpos	234, 274
COMMAND\@footnotemark	214, 436, 445
COMMAND\@footnotetext	214, 215, 436
COMMAND\@gobble	26, 35, 131, 132, 164, 258
COMMAND\@gobblefive	444
COMMAND\@gobblefour	442
COMMAND\@gobbleseven	260
COMMAND\@gobblethree	435
COMMAND\@h	193
COMMAND\@hangingsymbol	321
COMMAND\@iiiminipage	305, 307, 435, 445
COMMAND\@iiiminipage	305
COMMAND\@l	440
COMMAND\@l@tempcnta	160, 165, 172
COMMAND\@l@tempcntb	165
COMMAND\@l@reg	440
COMMAND\@lab	110, 281, 284, 288, 435
COMMAND\@ldunboxmpfoot	307

COMMAND\@led@extranofeet	381
COMMAND\@ledinnote@command	311, 312
COMMAND\@lemma	137, 139
COMMAND\@lock	104, 321
COMMAND\@lopL	436
COMMAND\@lopR	436
COMMAND\@makecol	273, 274, 276, 277, 445
COMMAND\@mempnum	279
COMMAND\@mpfnpos	234
COMMAND\@msd	331
COMMAND\@msd@c	331
COMMAND\@msd@options@iffullpage	337
COMMAND\@msdata@list	331, 332
COMMAND\@nl	110–113, 115, 124, 164, 284, 435, 436
COMMAND\@nl@reg	111, 379, 436, 440
COMMAND\@opXfeet	436
COMMAND\@opfeetX	445
COMMAND\@opxtrafeeti	445
COMMAND\@outputpage	279, 280
COMMAND\@page	111, 112, 284
COMMAND\@pend	436
COMMAND\@pendR	436
COMMAND\@ref	110, 120–122, 125, 126, 131
COMMAND\@ref@later	121, 127
COMMAND\@ref@reg	120, 164, 436
COMMAND\@ref@reg@parsearg	121
COMMAND\@reinserts	273, 275, 276, 445
COMMAND\@resetannot	164
COMMAND\@secondoftwo	76
COMMAND\@sect	364
COMMAND\@series	262
COMMAND\@set	115
COMMAND\@sidenotesep	303
COMMAND\@stopmsd	332
COMMAND\@sw	121, 122, 141, 144, 145
COMMAND\@tag	133, 134, 139
COMMAND\@tempcnta	89
COMMAND\@tempcntb	89
COMMAND\@this@crossref@start	213
COMMAND\@toksa	96
COMMAND\@toksb	96
COMMAND\@xloop	174
COMMAND\@xympar	297, 445
COMMAND\Aendnote	16, 29
COMMAND\Afootfmt	178
COMMAND\Afootgroup	178
COMMAND\Afootnote	8, 16, 27, 28, 31, 135, 186, 213, 235, 253, 444
COMMAND\Afootstart	178
COMMAND\AtBeginDocument	277
COMMAND\AtEndEveryPend	18, 451
COMMAND\AtEveryPend	18, 57, 151, 442, 443, 445, 451
COMMAND\AtEveryPend*	18
COMMAND\AtEveryPstart	18, 57, 442, 443, 445, 448, 451
COMMAND\AtEveryPstart*	18

COMMAND\AtEveryStanza	57, 448, 451
COMMAND\AtEveryStopStanza	57, 448, 451
COMMAND\AtStartEveryPstart	18, 451
COMMAND\AtStartEveryStanza	57, 451
COMMAND\BeforeEveryStopStanza	57, 451
COMMAND\Bendnote	16, 27
COMMAND\Bfootnote	8, 16, 213, 235, 253
COMMAND\Centering	50
COMMAND\Cfootnote	213
COMMAND\Columns	90, 147, 183
COMMAND\Dfootnote	213
COMMAND\Efootnote	213
COMMAND\Gls	68
COMMAND\H@old	364
COMMAND\Hy@raisedlink	373
COMMAND\Hy@raisedlink@left	374
COMMAND\LTR	50
COMMAND\Pages	90, 147, 274, 275, 278
COMMAND\ProcessOptionsX	79
COMMAND\RL	48
COMMAND\RaggedLeft	50
COMMAND\RaggedRight	50
COMMAND\RenewExpandableDocumentCommand	36, 143
COMMAND\SEonlypage	290, 447
COMMAND\SEref	61–64, 290, 294, 448, 450, 453
COMMAND\SErefonlypage	61–63, 384, 447
COMMAND\SErefwithpage	61, 63, 290, 294, 384, 447, 449, 454
COMMAND\Stanza	440
COMMAND\Waklam	354
COMMAND\X@doreinfeet	275, 445
COMMAND\XR@prefix	297
COMMAND\XR@test	296, 297
COMMAND\XR@test@mac	297
COMMAND\XR@test@mac@test	297
COMMAND\XXXXXXfmt	378
COMMAND\XXXXXXfmt	378
COMMAND\Xafternumber	45
COMMAND\Xafterlemmaseparator	47, 382
COMMAND\Xafternote	51, 381, 382
COMMAND\Xafternumber	45, 382
COMMAND\Xafterrule	52, 235, 382, 441, 443
COMMAND\Xaftersymlinenum	45, 382
COMMAND\Xarrangement	40, 51, 75, 178, 179, 264, 382
COMMAND\Xarrangement@footparagraph	184
COMMAND\Xarrangement@normal	179
COMMAND\Xarrangement@paragraph	184
COMMAND\Xbeforeinserting	50
COMMAND\Xbeforelemmaseparator	47, 382
COMMAND\Xbeforenotes	52, 235, 382, 440, 443
COMMAND\Xbeforenumber	45, 382
COMMAND\Xbeforepagenumber	42, 455
COMMAND\Xbeforesymlinenum	45, 382
COMMAND\Xbhookendnote	382
COMMAND\Xbhookgroup	52, 447, 448

COMMAND\Xbhooknote	50, 382–384, 446
COMMAND\Xboxendlinenum	46, 47, 382, 444
COMMAND\Xboxlinenum	46, 47, 382
COMMAND\Xboxlinenumalign	46, 47, 382, 444
COMMAND\Xboxstartlinenum	46, 47, 382, 444
COMMAND\Xboxsymlinenum	46, 382
COMMAND\Xcolalign	50, 444
COMMAND\Xdo@feet	445, 450
COMMAND\Xend	260
COMMAND\XendXafterlemmaseparator	383
COMMAND\XendXbeforelemmaseparator	383
COMMAND\XendXinplaceoflemmaseparator	383
COMMAND\XendXlemmaseparator	383
COMMAND\XendXmorethantwolines	383
COMMAND\XendXtwolines	383
COMMAND\Xendafterenumber	45, 446
COMMAND\Xendafterlemmaseparator	47, 383
COMMAND\Xendafternote	53, 448
COMMAND\Xendafternumber	49
COMMAND\Xendafterpagenumbe	449
COMMAND\Xendafterpagenumber	42, 63
COMMAND\XendafterpagenumberSereonlypage	384
COMMAND\Xendaftersymlinenum	45, 49, 446
COMMAND\Xendahookinplaceofnumber	49, 446
COMMAND\Xendahooklinenum	49, 446
COMMAND\Xendbeforelemmaseparator	47, 383
COMMAND\Xendbeforelinenum	49
COMMAND\Xendbeforenumber	45, 446
COMMAND\Xendbeforepagenumber	42, 62, 63
COMMAND\XendbeforepagenumberSereonlypage	62
COMMAND\Xendbeforesymlinenum	45, 49, 446
COMMAND\Xendbhookinplaceofnumber	49, 446
COMMAND\Xendbhooklinenum	49, 446
COMMAND\Xendbhooknote	50
COMMAND\Xendboxendlinenum	47, 444
COMMAND\Xendboxlinenum	47, 382, 443
COMMAND\Xendboxlinenumalign	47, 382, 444
COMMAND\Xendboxstartlinenum	47, 444
COMMAND\Xendboxsymlinenum	46, 446
COMMAND\Xendfootnote	133
COMMAND\Xendhangindent	49, 446, 448
COMMAND\Xendinplaceoflemmaseparator	29, 47, 383
COMMAND\Xendinplaceofnumber	46, 445
COMMAND\Xendinplaceofpagenumber	42, 450
COMMAND\Xendinsertsep@	241
COMMAND\Xendlemmadisablefontselection	48
COMMAND\Xendlemmafont	48, 447
COMMAND\Xendlemmaseparator	29, 47, 383
COMMAND\Xendletboxendlinenum	382
COMMAND\Xendlineflag	63
COMMAND\Xendlinenumannotationonlyfirst	45, 245, 453
COMMAND\Xendlinenumannotationonlyfirstintwo	45, 245, 453
COMMAND\Xendlinenumannotationposition	25, 63
COMMAND\Xendlineprefixmore	42, 63

COMMAND\Xendlineprefixsingle	42, 63
COMMAND\Xendlinerangeseparator	43, 63, 198, 384, 446
COMMAND\Xendmorethantwolines	29, 43, 63, 383, 444
COMMAND\Xendnoidenticallinenumannotation	26, 63, 245, 453
COMMAND\Xendnolinenum	455
COMMAND\Xendnolinenumifannotation	45, 455
COMMAND\Xendnonumber	43, 445
COMMAND\Xendnotboxingsubline	47, 455
COMMAND\Xendnote	238, 259, 260, 444
COMMAND\Xendnotefontsize	48, 382
COMMAND\Xendnotenumfont	48, 49, 381
COMMAND\Xendnotes	241
COMMAND\Xendnumberonlyfirstinline	41, 446
COMMAND\Xendnumberonlyfirstintwolines	41, 446
COMMAND\Xendpagenumberonlyfirst	42, 450
COMMAND\Xendpagenumberonlyfirsttifsingl	42, 450
COMMAND\Xendpagenumberonlyfirstintwo	42, 450
COMMAND\Xendparagraph	53, 440
COMMAND\Xendsep	53, 453
COMMAND\Xendstorelineinfo	239
COMMAND\Xendsublinesep	45, 63, 198
COMMAND\Xendsymlinenum	41, 446, 453
COMMAND\Xendsymlinenumannotation	45, 453
COMMAND\Xendsympagenum	42, 450
COMMAND\Xendtwolines	29, 43, 63, 383, 444
COMMAND\Xendtwolinesbutnotmore	43, 63, 444
COMMAND\Xendtwolinesonlyinsamepage	43, 63, 444
COMMAND\Xendtxtbeforenotes	51, 452
COMMAND\Xendwrapcontent	49, 448
COMMAND\Xendwraplemma	48, 448
COMMAND\Xendwraplinenumannotation	26, 63
COMMAND\Xfootfmt	383
COMMAND\Xfootgroup	183
COMMAND\Xfootins	182
COMMAND\Xfootnote	60, 66, 133, 378, 438, 441, 443, 444, 447, 449
COMMAND\Xfootstarts	183
COMMAND\Xgroupbyline	51, 171, 212, 271, 386, 452–455
COMMAND\Xgroupbylines	450
COMMAND\Xgroupbylineseparetwolines	51
COMMAND\Xhangindent	49, 446, 453
COMMAND\Xhsize	385, 447, 448
COMMAND\Xhsizethreecol	50, 53, 382
COMMAND\Xhsizetwocol	50, 53, 265, 382
COMMAND\Xinnotemark	66, 386, 454
COMMAND\Xinplaceoflemmaseparator	28, 47, 382
COMMAND\Xinplaceofnumber	28, 46, 382, 443, 444
COMMAND\Xinsertparafootsep	188, 190
COMMAND\Xledsetnormalparstuff	380, 381, 443
COMMAND\Xlemmadisablefontselection	48
COMMAND\Xlemmafont	48, 447
COMMAND\Xlemmaseparator	47, 198, 267, 269, 273, 382
COMMAND\Xlineflag	62, 385, 449
COMMAND\Xlinenumannotationonlyfirst	44, 45, 208, 453
COMMAND\Xlinenumannotationonlyfirstintwo	44, 45, 208, 453

COMMAND\Xlinenumannotationposition	25, 63, 163
COMMAND\Xlinenumannotationposition@side	163
COMMAND\Xlinenumannotationpositionside	25
COMMAND\Xlinangeseparator	43, 62, 198, 384, 446
COMMAND\Xmaxhnotes	52, 74, 75, 381, 382, 440, 443
COMMAND\Xmorethantwolines	28, 43, 62, 383, 443
COMMAND\Xnoidenticallinenumannotation	26, 63, 208, 453
COMMAND\Xnoindent	446
COMMAND\Xnolemmaseparator	47, 273, 383
COMMAND\Xnolinenumber	204, 455
COMMAND\Xnolinenumberifannotation	45, 204, 455
COMMAND\Xnonbreakableafternumber	45, 383, 439
COMMAND\Xnonote	41, 455
COMMAND\Xnonumber	43, 383
COMMAND\Xnonumberinfootnote	383
COMMAND\Xnopagenumberifcurrent	43, 210, 455
COMMAND\Xnotboxingsubline	47, 455
COMMAND\Xnotefontsize	48, 382
COMMAND\Xnotefontsize@(<s>)	188, 192, 193
COMMAND\Xnotenumfont	48, 381
COMMAND\Xnoteswidthliketwocolumns	53, 441
COMMAND\Xnumberonlyfirstinline	41, 42, 51, 107, 200, 239, 266, 267, 269, 383, 438, 442, 451
COMMAND\Xnumberonlyfirstintwolines	41, 51, 383, 438, 451
COMMAND\Xonlypstart	44, 383, 438, 442
COMMAND\Xpagelinesep	45, 450
COMMAND\Xparafootsep	51, 107, 381, 383, 386, 449, 451, 452
COMMAND\Xparafootsep@series	188
COMMAND\Xparindent	49, 385, 386, 444, 446, 448, 453
COMMAND\Xprenotes	52, 235, 449
COMMAND\Xprenotes@	182, 235, 438
COMMAND\Xprintlines	106
COMMAND\Xpstart	44, 379, 383, 438, 442
COMMAND\Xstarteverytime	44, 379, 383, 442
COMMAND\Xpstartonlyfirst	44, 454
COMMAND\Xpstartseparator	44, 454
COMMAND\Xragged	51
COMMAND\Xrigidbalance	190, 385, 447
COMMAND\Xsethangindent	197
COMMAND\Xstanza	44, 57
COMMAND\Xstanzaonlyfirst	44, 454, 455
COMMAND\Xstanzaseparator	44
COMMAND\Xstorelineinfo	200
COMMAND\Xsublinesep	22, 45, 63, 198, 384
COMMAND\Xsublinesepside	22, 45
COMMAND\Xsymlinenum	41, 51, 382, 383, 444, 451
COMMAND\Xsymlinenumannotation	44, 45, 453
COMMAND\Xtextbeforenotes	174
COMMAND\Xtoendnotes	30, 260
COMMAND\Xtwolines	28, 43, 63, 207, 209, 265, 383, 443
COMMAND\Xtwolinesappref	265
COMMAND\Xtwolinesbutnotmore	43, 63, 383, 444
COMMAND\Xtwolinesbutnotmoreappref	266
COMMAND\Xtwolinesonlyinsamepage	43, 63, 383, 444
COMMAND\Xtxtbeforenotes	51, 52, 383, 449–451

COMMAND\Xtxtbeforenotesonlyonce	51, 451
COMMAND\Xtxtbeforenumber	42
COMMAND\Xunvxh	186, 383
COMMAND\Xwidth	53, 385, 448
COMMAND\Xwrapcontent	49, 448
COMMAND\Xwraplemma	48, 50, 448
COMMAND\Xwraplinenumannotation	25, 26, 63, 163
COMMAND\Xwraplinenumannotation@ref	163
COMMAND\Xwraplinenumannotation@side	163
COMMAND\Xwraplinenumannotationref	26
COMMAND\Xwraplinenumnotationside	25
COMMAND\&	381
COMMAND\absline@num	104, 159
COMMAND\accent	132
COMMAND\actionlines@list	105, 161
COMMAND\actions@list	105
COMMAND\add@Xgroupbyline	171
COMMAND\add@hooktoggle@specific@to@cs	271
COMMAND\add@inserts	105, 170, 171
COMMAND\add@inserts@next	171
COMMAND\add@msd@	331
COMMAND\add@msdata	331, 332
COMMAND\add@msdata@firstlineofpage	335
COMMAND\add@msddata	331
COMMAND\add@penalties	159, 172
COMMAND\addcontentsline	360
COMMAND\addfootins	378, 381
COMMAND\addfootinsX	378, 381
COMMAND\addtoendnotes	260
COMMAND\advancelabel@refs	283
COMMAND\advanceline	22, 106, 114, 128, 445
COMMAND\affixline@num	165, 168, 169, 303, 436
COMMAND\affixpstart@num	169
COMMAND\afterXrule	382
COMMAND\afterXsymlinenum	382
COMMAND\afterantilabe	58
COMMAND\aftergroup	131, 136
COMMAND\afterlemmaseparator	382
COMMAND\afternote	382
COMMAND\afternoteX	51, 381, 382
COMMAND\afternumberinfootnote	382
COMMAND\afterruleX	52, 441, 443, 444
COMMAND\annot@end@print	208, 245
COMMAND\annot@list	106, 164
COMMAND\annotationref	25, 26, 59
COMMAND\antilabe	15, 58, 454
COMMAND\applabel	62, 136, 284, 286, 443, 449, 455
COMMAND\appref	60, 62–64, 290, 294, 447, 448, 453
COMMAND\apprefprefixmore	62, 381
COMMAND\apprefprefixsingle	62, 381
COMMAND\apprefwithpage	62, 63, 290, 294, 444, 447, 454
COMMAND\arrangementX	40, 75, 217, 264, 382
COMMAND\arrangementX@normal	222
COMMAND\article	15

COMMAND\at@every@pend	151
COMMAND\autopar	17, 18, 148, 152, 153, 232, 437, 439, 440, 443, 452
COMMAND\ballast	75
COMMAND\ballast@count	159, 173
COMMAND\baselineskip	41, 185, 188
COMMAND\beforeXnotes	382
COMMAND\beforeXsymlinenum	382
COMMAND\beforeantilabe	58
COMMAND\beforeedchapter	10, 72, 361
COMMAND\beforeinsertingX	50
COMMAND\beforelemmaseparator	382
COMMAND\beforenotesX	52, 384, 439, 440, 443, 444
COMMAND\beforenumberinfootnote	382
COMMAND\begin	339
COMMAND\beginnumbering	16, 17, 19, 90, 91, 94, 95, 103, 108, 123, 152, 237, 330, 437, 440, 444, 445, 450
COMMAND\beginnumberingR	90
COMMAND\bf	437
COMMAND\bfseries	48, 437
COMMAND\bhookXnote	382
COMMAND\bhookgroupX	52, 447
COMMAND\bhooknoteX	50, 383, 384, 446
COMMAND\body	322
COMMAND\bodyfootmarkA	38
COMMAND\book	15
COMMAND\boxXendlinenum	382
COMMAND\boxXendlinenumalign	382
COMMAND\boxXendstartlinenum	382
COMMAND\boxendlinenum	382
COMMAND\boxlinefootnote	202
COMMAND\boxlinenum	382
COMMAND\boxlinenumalign	382
COMMAND\boxstartlinenum	382
COMMAND\boxsymlinenum	382
COMMAND\break	41, 186
COMMAND\brokenpenalty	172
COMMAND\c@page	124
COMMAND\c@page@page	124
COMMAND\centering	50
COMMAND\ch@ck@l@ck	436
COMMAND\ch@cksub@l@ck	168, 436
COMMAND\chapter	71, 362, 366, 441, 443, 445, 449
COMMAND\chaptermark	360
COMMAND\check@pb@in@verse	373
COMMAND\colalignX	50, 444
COMMAND\collect@body	340
COMMAND\color	448
COMMAND\colorbox	75
COMMAND\columns	53
COMMAND\columnwidth	185, 441
COMMAND\command names	265, 266
COMMAND\continuousnumberingwithcolumns@sync@linenum@singletext	162
COMMAND\copyright	132
COMMAND\correct@Xfootins@box	443

COMMAND\correct@footinsX@box	443
COMMAND\count	192
COMMAND\critex	437
COMMAND\critext	139, 377, 378, 381
COMMAND\csname	80
COMMAND\csquotes	257
COMMAND\ctab	354, 359
COMMAND\ctabtext	359
COMMAND\current@annot	164
COMMAND\dcoll	348
COMMAND\def	77, 93
COMMAND\dimen	192
COMMAND\dimexpr	53
COMMAND\discretionary	186
COMMAND\displaywidowpenalty	172
COMMAND\do@Xfeet	274, 436, 445, 450
COMMAND\do@actions	159, 161, 436
COMMAND\do@actions@fixedcode	436
COMMAND\do@actions@next	159, 161
COMMAND\do@ballast	159, 173
COMMAND\do@feet@custom@order	274
COMMAND\do@insidelinehook	439
COMMAND\do@line	105, 130, 150, 153, 157, 170, 172, 321, 436, 439, 440
COMMAND\do@linehook	436
COMMAND\do@lockoff	106
COMMAND\do@lockon	106
COMMAND\dodoreintrafeet	435
COMMAND\doendnotes	29, 241, 242, 383, 444, 451
COMMAND\doendnotesbysection	29, 242, 260, 444, 451
COMMAND\doennotes	450
COMMAND\doinsidelinehook	26, 441
COMMAND\doinsidethislinehook	27, 157, 453
COMMAND\dolinehook	26, 441
COMMAND\doreintrafeeti	445
COMMAND\doreintrafeetii	445
COMMAND\doxtrafeet	435
COMMAND\doxtrafeeti	445
COMMAND\doxtrafeetii	445
COMMAND\dummy@ref	131
COMMAND\edaftertab	70, 354, 355
COMMAND\edannotationref	288
COMMAND\edatleft	70, 352
COMMAND\edatright	70, 71, 352
COMMAND\edbforetab	70, 354
COMMAND\edfilldimen	353
COMMAND\edfont@info	139
COMMAND\edgls	68, 310, 455
COMMAND\edgls...	448
COMMAND\edglsadd	450
COMMAND\edindex	36, 66–68, 310, 311, 314, 316, 344, 439, 441, 442, 445, 446, 449–451, 454
COMMAND\edindexlab	67
COMMAND\edlabel	59–62, 64, 132, 281, 283, 284, 287, 288, 296, 310, 343, 374, 385, 435, 437, 438, 440, 442, 447, 452
COMMAND\edlabelE	61, 286

COMMAND\edlabelS	61, 286
COMMAND\edlabelSE	61
COMMAND\edlineref	59, 281, 383, 442, 444, 447, 450
COMMAND\edmakelabel	61, 296
COMMAND\edpageref	59, 281, 287, 296
COMMAND\edrowfill	354
COMMAND\edsublineref	59
COMMAND\edtabcolsep	348
COMMAND\edtext	7, 24, 27, 28, 30–33, 35, 54, 59, 60, 62, 65, 66, 68, 75, 94, 105, 106, 120, 122, 125, 126, 130–139, 141, 142, 144, 146, 162, 164, 284–286, 289, 343, 345, 362, 374, 377, 378, 381, 435, 437, 438, 441–445, 449–452, 454, 455
COMMAND\edtext@level	444
COMMAND\edtext@notes	133, 134
COMMAND\edtext@notes@start	133, 134
COMMAND\edtextlater	122
COMMAND\edvertdots	71, 353
COMMAND\edvertline	71, 353
COMMAND\elechapter	72
COMMAND\eled@sectioning@out	367
COMMAND\eledchapter	71, 381, 441, 446
COMMAND\eledchapter*	72
COMMAND\eledmac@error	435
COMMAND\eledsection	7, 16, 72, 131, 156, 361, 381, 443, 451, 452, 454, 455
COMMAND\eledsection*	72
COMMAND\eledsubsection	72, 381
COMMAND\eledsubsection*	72
COMMAND\eledsubsubsection	72, 381
COMMAND\eledsubsubsection*	72
COMMAND\eledxxx	10, 72, 367, 440, 441
COMMAND\eledxxxx	360
COMMAND\elesection	367
COMMAND\else	309, 361
COMMAND\emph	35
COMMAND\empty	89, 166, 281
COMMAND\end	339
COMMAND\end@lemmas	131
COMMAND\endashchar	198
COMMAND\endgraf	150, 188, 232
COMMAND\endlock	22, 106, 129, 326
COMMAND\endminipage	305, 307, 435, 445
COMMAND\endmsdata	37
COMMAND\endnotes	443, 447
COMMAND\endnumbering	16, 17, 19, 91, 93, 94, 123, 436, 444, 450
COMMAND\endnumberingR	90
COMMAND\endprint	237, 240, 260, 380
COMMAND\endstanzaextra	381
COMMAND\endsub	21, 106, 127
COMMAND\endverse	384
COMMAND\everypar	152
COMMAND\extensionchars	73, 91
COMMAND\externaldocument	64, 296
COMMAND\f@x@l@cks	436
COMMAND>falseverse	381, 439, 440
COMMAND\fi	361

COMMAND\firstlinenum	20, 165, 437
COMMAND\firstsublinenum	20, 437
COMMAND\flag@end	125, 126, 138, 441
COMMAND\flag@end@RTL	126
COMMAND\flag@end@later	127
COMMAND\flag@start	125, 126, 138, 441
COMMAND\flag@start@RTL	126
COMMAND\flag@start@later	127
COMMAND\flagstanza	58
COMMAND\floatingpenalty	178, 442
COMMAND\flush@notes	173, 174
COMMAND\fnpos	39, 234, 440, 450
COMMAND\footfmt	177, 180
COMMAND\footfmt...	218
COMMAND\footfootmarkA	38
COMMAND\footfudgefactor	186
COMMAND\footfudgefiddle	75, 184, 185, 435
COMMAND\footgroup	178
COMMAND\footins	182
COMMAND\footnormal	265, 382, 435
COMMAND\footnormalX	382
COMMAND\footnote	37, 74, 213–215, 378, 436
COMMAND\footnote@lang	198
COMMAND\footnoteA	16, 37
COMMAND\footnoteB	16
COMMAND\footnoteC	27
COMMAND\footnoteE	37
COMMAND\footnoteX	8, 38, 256, 258, 449
COMMAND\footnoteX@reading	258
COMMAND\footnoteXmark	258, 451, 454
COMMAND\footnoteXmk	273
COMMAND\footnoteXtext	258, 451, 454
COMMAND\footnote⟨X⟩	132
COMMAND\footnote⟨X⟩mark	38
COMMAND\footnote⟨X⟩mk	38
COMMAND\footnote⟨X⟩nomk	38
COMMAND\footnote⟨X⟩text	38
COMMAND\footnotelang@lua	176
COMMAND\footnotelang@poly	177
COMMAND\footnotemark	38, 258
COMMAND\footnoteoption@	176, 446
COMMAND\footnoterule	191
COMMAND\footnotesize	48
COMMAND\footnotetext	38, 258
COMMAND\footparagraph	185, 265, 382, 440
COMMAND\footparagraphX	227, 382, 440
COMMAND\footplitskips	436, 442
COMMAND\footstart	178, 182, 191
COMMAND\footstrut	188
COMMAND\footthreecol	382
COMMAND\footthreecolX	382, 444
COMMAND\foottwocol	382
COMMAND\foottwocolX	382, 444
COMMAND\foreignlanguage	49

COMMAND\fullstop	384
COMMAND\get@edindex@hyperref	315
COMMAND\get@edindex@ledinnote@command	311
COMMAND\get@fnmark	215
COMMAND\get@index@command	439
COMMAND\get@linelistfile	436
COMMAND\get@thisfootnote	221
COMMAND\getline@num	158, 160
COMMAND\gl@p	97
COMMAND\global	110
COMMAND\globaldefs	110
COMMAND\gls	68, 318
COMMAND\hangindentX	49, 444, 446, 453
COMMAND\hangingsymbol	381, 437
COMMAND\hbox	186
COMMAND\hfill	439
COMMAND\hidenumering	23, 119, 386, 443, 453, 454
COMMAND\hidenumeringonleftpage	23, 119, 449
COMMAND\hidenumeringonrightpage	23, 119, 449
COMMAND\hline	68
COMMAND\hooktoggle@specific	271
COMMAND\hrulefill	354
COMMAND\hspace	41, 182, 185, 186, 192, 195, 233, 436, 441
COMMAND\hspaceX	385, 447, 448
COMMAND\hsizethreecol	382
COMMAND\hsizethreecolX	50, 53
COMMAND\hsizetwocol	382
COMMAND\hsizetwocolX	50, 53
COMMAND\hyperlinkR	315
COMMAND\hyperlinkformat	315
COMMAND\hyperlinkformatR	315
COMMAND\if@RTL	81, 135
COMMAND\if@edtext@	442, 444
COMMAND\if@eled@sectioning	367
COMMAND\if@firstlineofpage	81
COMMAND\if@firstlineofpageR	81
COMMAND\if@msd@options@fullpage	337
COMMAND\if@msdata@insertedfrompreviouspage	335
COMMAND\if@nobreak	151
COMMAND\if@noneed@notes	133
COMMAND\ifXnote@	90
COMMAND\ifafterendnumberingR	90
COMMAND\ifautopar	152
COMMAND\ifby@autopar	152
COMMAND\ifbypage@	97
COMMAND\ifbypage@R	97
COMMAND\ifbypstart@	97
COMMAND\ifbypstart@R	97
COMMAND\iffirst@linenum@out@	123
COMMAND\ifindtl@innote	90
COMMAND\ifindtl@notenumber	90
COMMAND\ifinserthangingsymbol	321
COMMAND\ifinstanza	321
COMMAND\ifistwofollowinglines	209

COMMAND\ifl@d@Xlinenumber	204
COMMAND\ifl@d@Xmorethantwolines	204, 443
COMMAND\ifl@d@Xtwolines	204
COMMAND\ifl@d@dash	204
COMMAND\ifl@d@elin	204
COMMAND\ifl@d@esl	204
COMMAND\ifl@d@pnum	203
COMMAND\ifl@d@ssub	204
COMMAND\ifl@dend@X	259
COMMAND\ifl@dmemoir	435
COMMAND\ifl@dpaging	441
COMMAND\ifl@dpairing	90, 436
COMMAND\ifl@dprintingpages	442
COMMAND\ifl@dskipnumber	165
COMMAND\ifl@dstartendok	353
COMMAND\ifl@imakeidx	81
COMMAND\ifl@noexpands@edgl	320
COMMAND\ifledRcol	90, 437
COMMAND\ifledRcol@	90, 440
COMMAND\iflemmacommand@	442
COMMAND\iflinenumannotationotherside	163
COMMAND\ifnoend@	242
COMMAND\ifnoledgroup@	309
COMMAND\ifnoteschanged@	107
COMMAND\ifnumberedpar@	147
COMMAND\ifnumbering	91, 94
COMMAND\ifnumberingR	90, 437
COMMAND\ifnumberline	138, 165
COMMAND\ifpst@rted	436
COMMAND\ifpst@rtedL	92
COMMAND\ifresumenumbering@start	95
COMMAND\ifseriesbefore	263
COMMAND\ifstopmsdata@inserted@	330
COMMAND\ifsublines@	104, 116
COMMAND\ifsublines@R	104
COMMAND\iftrue	444
COMMAND\ifvmode	283
COMMAND\ifxxx	361
COMMAND\ignorespaces	135
COMMAND\imki@wrindexentry	81
COMMAND\immediate	123, 237
COMMAND\indent	18, 152, 384
COMMAND\index	317, 450
COMMAND\indtl@wrindexentry	81
COMMAND\initnumbering@quote	359, 445
COMMAND\initnumbering@reg	436
COMMAND\initnumbering@sectcmd	445
COMMAND\innotemarkX	66, 386, 454
COMMAND\inplaceoflemmaseparator	382
COMMAND\inplaceofnumber	382
COMMAND\insert	137, 170, 177, 180, 193, 212, 218
COMMAND\insert@Xtxtbeforenotes	174, 193
COMMAND\insert@count	120, 125, 135
COMMAND\insert@countR	135

COMMAND\insert@msdata	331, 337
COMMAND\insert@txtbeforenotesX	174
COMMAND\insertthangingsymbol	439
COMMAND\insertlines@list	105, 120
COMMAND\insertparafootsepX	231
COMMAND\inserts@list	130, 148, 170, 171, 186
COMMAND\interAfootnotelinepenalty	437
COMMAND\interfootnotelinepenalty	437
COMMAND\interlinepenalty	178
COMMAND\interparanoteglue	381
COMMAND\justifying	50
COMMAND\keep@this@crossref@forinserting	213
COMMAND\l@advance@parledegroupp@beforenormalnotes	445
COMMAND\l@d@@wrindexhyp	442
COMMAND\l@d@add	140
COMMAND\l@d@end	237, 259
COMMAND\l@d@nums	134, 138–140, 203, 204
COMMAND\l@d@section	237
COMMAND\l@d@set	115, 129
COMMAND\l@dampcount	345
COMMAND\l@dbfnote	215, 436
COMMAND\l@dcheckstartend	353
COMMAND\l@dchset@num	115
COMMAND\l@dcolcount	345, 346
COMMAND\l@dcollect@@body	339
COMMAND\l@dcollect@body	339
COMMAND\l@dcsnote	440
COMMAND\l@dcsnotetext	157, 301
COMMAND\l@dcsnotetext@l	157, 301
COMMAND\l@dcsnotetext@r	157, 301
COMMAND\l@ddodorextrafeet	275, 435
COMMAND\l@ddoxtrafeet	273, 275, 435
COMMAND\l@emptyd@ta	436
COMMAND\l@dend@close	237
COMMAND\l@dend@open	237
COMMAND\l@dend@stuff	237
COMMAND\l@denvbody	339
COMMAND\l@dfeetbeginmini	435
COMMAND\l@dfeetendmini	435
COMMAND\l@dgetline@margin	436
COMMAND\l@dgetlock@disp	436
COMMAND\l@dgetref@num	288, 289
COMMAND\l@dgetsidenote@margin	297, 436
COMMAND\l@dgobbleoptarg	441
COMMAND\l@dgobblearg	441
COMMAND\l@dgobbleoptarg	344
COMMAND\l@dlabel@parse	288, 289
COMMAND\l@dld@ta	165, 167
COMMAND\l@dlp@rbox	303
COMMAND\l@dlsn@te	436
COMMAND\l@dlsnote	440
COMMAND\l@dmake@labels	283, 297
COMMAND\l@dmake@labelsR	297
COMMAND\l@dnumpstartsL	92, 436

COMMAND\l@dp@rsefootspec	204
COMMAND\l@dp@rsefootspec	204
COMMAND\l@dpush@begins	339
COMMAND\l@d@rd@ta	165, 167
COMMAND\l@dref@undefined	288
COMMAND\l@drsn@te	436
COMMAND\l@drsnote	440
COMMAND\l@dtabaddcols	353
COMMAND\l@dtabnoexpands	435
COMMAND\l@dumboxmpfoot	445
COMMAND\l@dunboxmpfoot	436
COMMAND\l@dunhbox@line	153
COMMAND\l@dzeropenalties	436, 441
COMMAND\l@noexpands@edgl	320
COMMAND\l@pb	372
COMMAND\l@prev@nopb	372
COMMAND\l@prev@pb	372
COMMAND\l@reg	379
COMMAND\l@secondoptarg	344
COMMAND\l@wrap@ifnotemptybox	89
COMMAND\label	19, 61, 64, 67, 281, 282, 289
COMMAND\label@refs	281
COMMAND\labelstarttrue	19, 385, 438, 452
COMMAND\labelref@list	281, 284
COMMAND\language	186
COMMAND\lastbox	152
COMMAND\lastskip	127
COMMAND\latex@makecol	276
COMMAND\leavevmode	18, 152
COMMAND\led@check@nopb	372
COMMAND\led@check@pb	372
COMMAND\led@nopb	372, 373
COMMAND\led@nopbnum	372
COMMAND\led@pb	372, 373
COMMAND\led@pb@macro	372
COMMAND\led@pbnum	372
COMMAND\led@reinit@index@fornote	317
COMMAND\led@set@index@fornote	317
COMMAND\ledRflag	315
COMMAND\ledchapter	381, 438
COMMAND\ledfootinsdim	381
COMMAND\ledinnernote	64, 299, 440, 441, 449
COMMAND\ledinnote	312, 380, 386, 444
COMMAND\ledinnotemark	380, 386, 443, 454
COMMAND\ledleftnote	64, 299
COMMAND\ledlinenum	102, 436
COMMAND\ledllfill	158
COMMAND\ledlsnotefontsetup	448
COMMAND\ledlsnotesep	65
COMMAND\ledlsnotewidth	65
COMMAND\lednopb	73, 372
COMMAND\lednopbinverse	373
COMMAND\lednopbinversetrue	56, 73
COMMAND\lednopbnum	372

COMMAND\ledouternote	64, 299, 440, 441, 449
COMMAND\ledpb	73, 372, 381
COMMAND\ledpbnum	372
COMMAND\ledpbsetting	73, 372, 446
COMMAND\ledrightnote	64, 299
COMMAND\ledrsnotefontsetup	448
COMMAND\ledrsnotesep	65
COMMAND\ledrsnotewidth	65
COMMAND\ledsection	381
COMMAND\ledsectnomark	360
COMMAND\ledsectnotoc	360
COMMAND\ledsetnormalparstuff	380, 381, 443
COMMAND\ledsetnormalparstuff@common	232
COMMAND\ledsetnormalparstuffX	380, 381, 443
COMMAND\ledsidenote	64, 299, 301
COMMAND\ledsidenote@	299
COMMAND\ledsubsection	381
COMMAND\ledsubsubsection	381
COMMAND\ledxxx	440
COMMAND\left	70
COMMAND\leftctab	354
COMMAND\leftheadline	102
COMMAND\leftlinenum	21, 102, 435, 436
COMMAND\leftlinenumannotation	103
COMMAND\leftltab	354
COMMAND\leftnoteupfalse	65
COMMAND\leftstartnum	169
COMMAND\leftrtab	354
COMMAND\leftsidenote	301
COMMAND\leftskip	182, 185, 186
COMMAND\lemma	3, 28, 30–33, 35, 130, 134–136, 139, 142, 377, 437, 438, 444, 445, 447, 455
COMMAND\lemmaseparator	382
COMMAND\let	31, 54, 132, 137, 258, 276, 326, 435
COMMAND\letboxXendendlinenum	382
COMMAND\line	190, 194
COMMAND\line@list	105, 106, 121, 138
COMMAND\line@list@stuff	91, 108, 123, 435, 437
COMMAND\line@list@version	110
COMMAND\line@margin	98, 167, 297
COMMAND\line@num	104, 106, 165, 435
COMMAND\line@set	140
COMMAND\lineannot	25, 60, 140
COMMAND\lineannot@set	140
COMMAND\lineation	21, 97, 452, 454
COMMAND\linebreak	41
COMMAND\lineneation	448
COMMAND\linenum	25, 28, 30, 31, 60, 62, 130, 139, 140, 287, 289, 296, 377
COMMAND\linenum@out	122, 123, 281, 284
COMMAND\linenumannotation	24, 162, 452, 453
COMMAND\linenumannotationothersidefalse	25
COMMAND\linenumannotationothersidettrue	25, 455
COMMAND\linenumberlist	20, 21, 89, 166, 435
COMMAND\linenumberstyle	22, 101, 435, 450
COMMAND\linenumincrement	20, 437

COMMAND\linenummargin	21, 98, 297
COMMAND\linenumr@p	101, 435, 437
COMMAND\linenumrep	101, 437
COMMAND\linenumsep	21, 65, 102, 298
COMMAND\linerangesep@	273
COMMAND\lineref	281, 287, 296, 383, 442
COMMAND\list@clear	96
COMMAND\list@clearing@reg	437
COMMAND\list@create	96
COMMAND\lock@disp	100
COMMAND\lock@off	117
COMMAND\lock@on	117
COMMAND\lockdisp	22, 100
COMMAND\loop	174, 322
COMMAND\ltab	354, 355, 359
COMMAND\ltabtext	359
COMMAND\m@mmf@prepare	214
COMMAND\mainmatter	452, 453
COMMAND\makeatletter	157
COMMAND\makehboxoffhboxes	187, 188
COMMAND\makeindex	66, 314
COMMAND\makelabel	296
COMMAND\managestanza@modulo	323
COMMAND\marginpar	64, 74, 297, 436
COMMAND\marginparwidth	65, 298
COMMAND\markboth	26, 27, 157
COMMAND\mathchardef	322
COMMAND\maxhXnotes	382
COMMAND\maxhnotesX	52, 75, 381, 439, 440, 443, 444
COMMAND\maxlinesinpar@list	108
COMMAND\measurebody	356
COMMAND\measretbody	356
COMMAND\memorybreak	20
COMMAND\metaXXXvfootnote	212
COMMAND\microtype	153
COMMAND\morenoexpands	75, 76, 130, 132
COMMAND\morethantwolines	383
COMMAND\mpfnpos	39, 234, 440, 450
COMMAND\mpnormalfootgroup	435
COMMAND\mpnormalvfootnote	435
COMMAND\msdata	36, 37, 329–331, 450, 451, 453
COMMAND\msdataposition	37
COMMAND\multfootsep	38, 214
COMMAND\multiplefootnotemarker	214
COMMAND\musixtex	440
COMMAND\n@num	437, 443
COMMAND\n@num@ref	443
COMMAND\new@line	124, 436
COMMAND\new@series	132
COMMAND\newcommand	31, 77, 93, 214, 283
COMMAND\newcommandx	31
COMMAND\newhookarg@specific	271
COMMAND\newhookcommand@series	265, 266, 443, 454
COMMAND\newhookcommand@series@reload	266

COMMAND\newhookcommand@toggle@reload	266, 441
COMMAND\newhooktoggle@series	266, 443
COMMAND\newhooktoggle@specific	271
COMMAND\newif	443
COMMAND\newline	41
COMMAND\newlinechar	259
COMMAND\newseries	39, 378, 381
COMMAND\newseries@	250, 263
COMMAND\newverse	57, 58, 381, 440, 451
COMMAND\next	322
COMMAND\next@action	109
COMMAND\next@actionline	109
COMMAND\next@insert	171
COMMAND\next@line@list@stuff	123, 452
COMMAND\nl@regR	111
COMMAND\no@expands	75, 132, 139, 258, 320, 435
COMMAND\noXlemmaseparator	383
COMMAND\nobreak	203
COMMAND\nocritical	251
COMMAND\noeledsec	381
COMMAND\noendnotes	381
COMMAND\noexpand	378
COMMAND\nofamiliar	269
COMMAND\noindent	18, 56, 57, 152, 446
COMMAND\noindentX	446
COMMAND\nomk@	273
COMMAND\nonbreakableafternumber	383
COMMAND\nonoteX	41, 455
COMMAND\nonumberinfootnote	383
COMMAND\norelax	54
COMMAND\normal@footnotemarkX	217
COMMAND\normal@page@break	371
COMMAND\normal@pars	232
COMMAND\normalbfnoteX	437
COMMAND\normalbodyfootmarkX	217
COMMAND\normalfont	450
COMMAND\normalfootfmt	54, 182, 188, 198, 237, 448
COMMAND\normalfootfmtX	218, 219
COMMAND\normalfootfootmarkX	219
COMMAND\normalfootgroup	183
COMMAND\normalfootgroupX	219
COMMAND\normalfootnoterule	178
COMMAND\normalfootstart	182, 185
COMMAND\normalfootstartX	219
COMMAND\normalvfootnote	180, 181
COMMAND\normalvfootnote@inserted	180, 181
COMMAND\normalvfootnoteX	218
COMMAND\notbool	361
COMMAND\notefontsetup	382
COMMAND\notefontsizeX	48, 382
COMMAND\notenumfont	381
COMMAND\notenumfontX	48, 381
COMMAND\notesXwidthliketwocolumns	383
COMMAND\noteswidthliketwocolumnsX	53, 383, 441, 444

COMMAND\num@lines	147, 172
COMMAND\numberlinefalse	20, 453
COMMAND\numberlinetrue	20, 453
COMMAND\numberonlyfirstinline	263, 383
COMMAND\numberonlyfirstintwolines	383
COMMAND\numberstartfalse	19
COMMAND\numberstarttrue	19, 44, 379, 437, 445, 450, 452
COMMAND\numberstanza	44
COMMAND\numberstanzafalse	57
COMMAND\numberstanzatrue	57
COMMAND\numlabfont	21, 53, 102
COMMAND\one@line	147
COMMAND\onehalfspacing	446
COMMAND\onlyXpstart	383
COMMAND\onlysideX	257
COMMAND\page@action	106, 115
COMMAND\page@start	106, 437
COMMAND\pagecontents	106
COMMAND\pagenumbering	279
COMMAND\pageparbreak	381
COMMAND\pageref	61, 287
COMMAND\par	30, 41, 152, 232
COMMAND\par@line	147, 172
COMMAND\par@patch@pagenumbering	279
COMMAND\par@patch@thepage	124, 279
COMMAND\para@footgroup	185
COMMAND\para@footgroupX	230
COMMAND\para@footsetup	185, 435
COMMAND\para@footsetupX	228, 435, 441
COMMAND\para@vfootnoteX	228
COMMAND\parafootfmt	187, 188, 448
COMMAND\parafootfmtX	229
COMMAND\parafootftm	190
COMMAND\parafootftmX	231
COMMAND\parafootftmsep	381
COMMAND\parafootsep	383, 439, 444
COMMAND\parafootsepX	51, 107, 381, 383, 386, 449, 454, 455
COMMAND\parafootstart	185
COMMAND\parafootstartX	228
COMMAND\paravfootnote	186, 188
COMMAND\parfillskip	188
COMMAND\parindent	446
COMMAND\parindentX	49, 385, 446, 448, 453
COMMAND\parse@annot	164
COMMAND\parshape	74
COMMAND\parskip	18, 57, 152
COMMAND\pause@numbering	94
COMMAND\pausenumbering	19, 94, 108, 110, 153, 440, 441, 448
COMMAND\pausenumbering@page@num	94
COMMAND\penalty	188
COMMAND\pend	2, 7, 17, 18, 20–22, 72, 128, 130, 133, 141, 147, 148, 150–153, 169, 170, 328, 379, 439–441, 449, 451, 452
COMMAND\preXnotes	443, 449
COMMAND\prenotesX	52, 236, 443

COMMAND\prepare@Xgroupbyline	212, 213
COMMAND\prepare@Xprenotes	235
COMMAND\prev@nopb	372
COMMAND\prev@pb	372
COMMAND\prevlineX	107
COMMAND\prevpageX@num	107
COMMAND\print@Xfootnoterule	443
COMMAND\print@Xnotes	274, 275
COMMAND\print@Xnotes@forpages	443
COMMAND\print@eledsection	156
COMMAND\print@footnoterule	443
COMMAND\print@leftmargin@eledsection	361
COMMAND\print@lemma	199
COMMAND\print@line	154
COMMAND\print@notesX@forpages	443
COMMAND\print@rightmargin@eledsection	361
COMMAND\printendlines	242, 250, 294, 384, 435, 437
COMMAND\printlinefootnote	200, 201, 442
COMMAND\printlinefootnotearea	201, 203, 442
COMMAND\printlinefootnotenumbers	200
COMMAND\printlines	182, 198, 203, 205, 242, 294, 384, 435, 437, 443, 447
COMMAND\printnpnum	380, 384
COMMAND\printpstart	199
COMMAND\protect	132, 378
COMMAND\providecommand	214, 435
COMMAND\pstarL	148
COMMAND\pstart	2, 7, 17–22, 71, 72, 115, 128, 129, 133, 137, 141, 147, 148, 150–152, 156, 170, 328, 366, 379, 437–441, 443–446, 448–452, 454
COMMAND\pstartinfootnote	383
COMMAND\pstartinfootnoteeverytime	383
COMMAND\pstartnum	169
COMMAND\pstartref	59, 281, 288, 440, 451
COMMAND\pstarts	437
COMMAND\pstarts@readL	147
COMMAND\pstarts@typeset@L	147
COMMAND\raggedX	51
COMMAND\raggedleft	50
COMMAND\raggedright	50
COMMAND\raw@text	147, 148
COMMAND\rbracket	47
COMMAND\read@linelist	108–110
COMMAND\ref	61, 64, 67
COMMAND\reformatted@	294
COMMAND\reformattedwithpage	294
COMMAND\relax	18, 115, 137, 159, 171, 326, 344, 379
COMMAND\renewcommand	75, 381, 382, 384
COMMAND\reset@current@annot	164
COMMAND\reset@msd@options@iffullpage	337
COMMAND\resetlinenumannotation	24, 164
COMMAND\resetprevline@	107
COMMAND\resetprevpage@	107
COMMAND\resumenumbering	19, 90, 91, 94, 95, 108, 110, 153, 437, 440, 441, 448
COMMAND\right	70
COMMAND\rightctab	354

COMMAND\rightlinenum	21, 102, 435, 436
COMMAND\rightlinenumannotation	103
COMMAND\rightlftab	355
COMMAND\rightnoteupfalse	65
COMMAND\righttrtab	355
COMMAND\rightsidenote	301
COMMAND\rightskip	182, 185, 186, 188
COMMAND\rightstartnum	169
COMMAND\rigidbalance	190, 192–194, 385, 447
COMMAND\rigidbalanceX	190, 385, 447
COMMAND\robustify	41, 45
COMMAND\roman	348, 447
COMMAND\rtab	354, 355, 359
COMMAND\rtabtext	356, 359
COMMAND\sameparallelpagenum	124
COMMAND\sameword	15, 31–33, 35, 122, 141–143, 146, 386, 442, 444–446, 449, 451, 452, 454
COMMAND\sameword@inedtext	142
COMMAND\saweword	142
COMMAND\scriptsize	102
COMMAND\section	71, 437, 449
COMMAND\section@num	91
COMMAND\sectionmark	360
COMMAND\select@lemmafont	53, 54, 176
COMMAND\series	250
COMMAND\series@	250
COMMAND\seriesatbegin	39, 262, 443
COMMAND\seriesatend	39, 263, 443
COMMAND\set@Xtxtbeforenotes	174
COMMAND\set@continuousnumberingforl	95
COMMAND\set@line	138
COMMAND\set@line@action	106, 116
COMMAND\set@txtbeforenotesX	174
COMMAND\setSErefonlypageprefixmore	62, 293, 447, 451
COMMAND\setSErefonlypageprefixsingle	62, 293, 447, 451
COMMAND\setSErefprefixmore	62
COMMAND\setSErefprefixsingle	62
COMMAND\setapprefprefixmore	62, 381
COMMAND\setapprefprefixsingle	62, 64, 381, 447
COMMAND\setcommand@series	264
COMMAND\sethangindentX	197
COMMAND\sethangingsymbol	56, 321, 381, 446
COMMAND\sethanginsymbol	54
COMMAND\setistwofollowinglines	209
COMMAND\setl@dlprbox	303
COMMAND\setline	22, 106, 111, 115, 128, 132, 446
COMMAND\setlinenum	22, 110, 115, 129, 151, 435, 452
COMMAND\setlinenumannotationsep	26, 386, 453
COMMAND\setmsdatalabel	37
COMMAND\setmsdataposition	450
COMMAND\setmsdataseries	37
COMMAND\setprintendlines	242, 246, 437
COMMAND\setprintlines	205, 209, 242, 437
COMMAND\setsidenotesep	65
COMMAND\setsidenotsep	382

COMMAND\setstanzaindent	323
COMMAND\setstanzaindents	55, 322, 379, 454
COMMAND\setstanzapenalties	322
COMMAND\setstanzavalues	322
COMMAND\settoggle@series	264, 438, 441
COMMAND\showlemma	131, 435
COMMAND\showwordrank	35, 142
COMMAND\sidenote@margin	436
COMMAND\sidenotemargin	64, 436, 440
COMMAND\sidenotesep	382
COMMAND\sidepstartnumtrue	19
COMMAND\skip	182
COMMAND\skipnumbering	23, 118, 129, 437, 443–445
COMMAND\skipnumbering@reg	443
COMMAND\small	48
COMMAND\somemacro	35
COMMAND\special	13
COMMAND\splitmaxdepth	178, 192
COMMAND\splitoff	190
COMMAND\splittopskip	178, 192, 193
COMMAND\stanza	22, 23, 56–58, 90, 326, 382, 384, 446, 451
COMMAND\stanza@hang	325
COMMAND\stanza@line	325
COMMAND\stanzaindent	55, 323, 442
COMMAND\stanzaindent*	55
COMMAND\stanzaindentbase	322
COMMAND\stanzanumwrapper	57
COMMAND\startlock	20, 22, 106, 129, 326
COMMAND\startstanzahook	382
COMMAND\startsub	21, 106, 127
COMMAND\stopmsd	332
COMMAND\stopmsdata	36, 37, 330
COMMAND\store@annot@to@absline	164
COMMAND\strip@pt	185
COMMAND\strutbox	192
COMMAND\sub@action	106, 116
COMMAND\sub@lock	104
COMMAND\sub@off	113, 284
COMMAND\sub@on	113, 284
COMMAND\subline@num	104, 106
COMMAND\sublinenum@rep	435
COMMAND\sublinenumberstyle	23, 101, 435
COMMAND\sublinenumincrement	20
COMMAND\sublinenumr@p	101, 435, 437
COMMAND\sublinenumrep	101, 437
COMMAND\sublineref	59, 281, 287
COMMAND\subsectionmark	360
COMMAND\sw@inthisedtext	134
COMMAND\sw@list@inedtext	139, 146
COMMAND\sw@txt	143
COMMAND\swnoexpands	35, 452
COMMAND\symlinenum	383
COMMAND\symplinenum	382
COMMAND\sza@penalty	325

COMMAND\tag	442
COMMAND\text	377
COMMAND\text(language)	49
COMMAND\textcolor	76
COMMAND\textheight	75
COMMAND\textsuperscript	25, 89, 163
COMMAND\the	124, 435
COMMAND\thefootnoteA	38
COMMAND\thefootnoteX	439
COMMAND\thelabidx	316
COMMAND\thepage	111, 124, 279
COMMAND\thepstart	19, 385
COMMAND\thepstartL	437
COMMAND\thepstartR	437
COMMAND\thestanza	57
COMMAND\this@crossref@end	213
COMMAND\this@line@list@version	123
COMMAND\this@section@next@page@num	94
COMMAND\this@section@page@num	107
COMMAND\thisfootnote	221
COMMAND\threecol@begin@insert	193
COMMAND\threecolfootfmt	193, 448
COMMAND\threecolfootfmtX	226
COMMAND\threecolfootgroup	192
COMMAND\threecolfootgroupX	226
COMMAND\threecolfootsetup	192
COMMAND\threecolfootsetupX	225
COMMAND\threecolvfootnote	192
COMMAND\threecolvfootnote@inserted	192
COMMAND\threecolvfootnoteX	226
COMMAND\toendnotes	29, 241, 449
COMMAND\twocolfootfmt	448
COMMAND\twocolfootfmtX	224
COMMAND\twocolfootgroupX	224
COMMAND\twocolfootsetupX	223
COMMAND\twocolvfootnoteX	224
COMMAND\twolines	263, 383
COMMAND\twolines@A	263
COMMAND\twolines@B	263
COMMAND\twolines@C	263
COMMAND\twolinesbutnotmore	383
COMMAND\twolinesonlyinsamepage	383
COMMAND\txbeforeXnotes	383
COMMAND\txbeforenotesX	51, 174, 451
COMMAND\txbeforenotesonlyonceX	51, 451
COMMAND\uline	38
COMMAND\unhbox	153, 186
COMMAND\unpenalty	187, 188
COMMAND\unskip	188
COMMAND\unvxh	188, 383
COMMAND\unvxhX	383
COMMAND\upbracefill	354
COMMAND\usingcritext	378, 381
COMMAND\usingedtext	378, 381

COMMAND\VAfootnote	178
COMMAND\vadjust	126
COMMAND\variant	31
COMMAND\vbox	150, 152, 186, 190, 234
COMMAND\vfootnote	177, 182, 186, 192
COMMAND\vl@dbfnote	215, 436
COMMAND\vl@disnote	301
COMMAND\vl@dlsnote	301
COMMAND\vl@dosnote	301
COMMAND\vl@drsnote	301
COMMAND\vnumfootnoteX	437
COMMAND\vsizer	52, 75
COMMAND\vsplit	172
COMMAND\waklam	354
COMMAND\waklamec	354
COMMAND\wapunktel	354
COMMAND\wastricht	354
COMMAND\widthX	53, 385, 448
COMMAND\wrap@edcrossref	286, 441
COMMAND\wrapcontentX	49, 448
COMMAND\wrapped@bodyfootmarkX	232
COMMAND\wrapped@footfootmarkX	231
COMMAND\x...	60
COMMAND\xabslineref	287
COMMAND\xannotationref	60
COMMAND\xdef	96, 326
COMMAND\xflagref	60, 288, 384, 447
COMMAND\xleft@appenditem	96, 131
COMMAND\xlineref	60, 384, 447
COMMAND\xpageref	60
COMMAND\xparse	36
COMMAND\xpstartref	60, 440, 452
COMMAND\xr	64
COMMAND\xright@appenditem	96
COMMAND\xsublineref	60
COMMAND\xxref	60, 289, 296, 440, 443, 449
COMMAND\xxxmatter	279
ENVIRONMENTastanza	91, 447
ENVIRONMENTedarrayc	359
ENVIRONMENTedarrayl	359
ENVIRONMENTedarrayr	359
ENVIRONMENTedtabular	454
ENVIRONMENTedtabularc	359
ENVIRONMENTedtabularl	359
ENVIRONMENTedtabularr	359
ENVIRONMENTledgroup	81, 308, 385, 447
ENVIRONMENTledgroupsized	308
PACKAGE(r)(e)ledmac	39
PACKAGEEDMAC	1, 6, 10, 12, 13, 77, 203, 213, 281, 322, 377, 388, 435
PACKAGEEledmac	106, 313, 380, 381, 442, 443
PACKAGEEledpar	443
PACKAGEEtoolbox	80
PACKAGEParallel	388
PACKAGEReledmac	58, 383–385

PACKAGEamsgen	338
PACKAGEamsmath	338
PACKAGEbabel	49, 76, 348, 447
PACKAGEbiblatex	74
PACKAGEbidi	48, 50, 81, 446, 454
PACKAGEcalc	455
PACKAGEcaption	89
PACKAGEcolor	75
PACKAGEcsquotes	451
PACKAGEedstanza	1, 13, 321
PACKAGEeledmac	1, 10, 14, 15, 67, 213, 309, 313, 341, 362, 374, 378, 380–382, 438, 440–442
PACKAGEeledpar	178, 388, 437, 440–443
PACKAGEetex	446
PACKAGEetoolbox	96, 141, 250, 263, 273, 301, 361, 371, 372
PACKAGEfancyhdr	276, 277, 449
PACKAGEfloatrow	74, 385
PACKAGEfootmisc	38, 76, 81, 213, 214, 388, 450
PACKAGEgeometry	15
PACKAGEglossaries	68, 318, 447, 455
PACKAGEglossaries-extra	68, 455
PACKAGEhandout	442
PACKAGEhyperlink	258
PACKAGEhyperref	60, 134, 136, 231, 232, 282, 286, 315, 364, 373, 439–442, 448, 451, 453, 455
PACKAGEifluatex	80
PACKAGEifxetex	80
PACKAGEimakeidx	66, 74, 81, 309, 313, 381, 439–442
PACKAGEindextols	317
PACKAGEindextool	381
PACKAGEindextools	66, 74, 81, 90, 309, 313, 317, 381, 442, 447, 450
PACKAGEledarab	76
PACKAGEledmac	1, 10, 13, 14, 76, 96, 313, 377, 378, 381, 384
PACKAGEledpar	76
PACKAGEMemoir	81, 313, 381, 388, 441, 445
PACKAGEMorewrites	74
PACKAGEMusixtex	440
PACKAGEperpage	447
PACKAGEpolyglossia	47, 76, 177, 198, 447, 454, 455
PACKAGERagged2e	50, 80
PACKAGEReledmac	1, 6, 10–12, 14–17, 19, 20, 23, 26, 27, 29–33, 36, 38–40, 42–44, 46, 49, 50, 52, 54, 56, 58, 60–62, 64–68, 72–78, 97, 99, 105, 106, 109, 110, 113, 122–124, 132, 135, 137, 140, 141, 170, 179, 182, 186, 198, 213, 237, 251, 255, 256, 263, 273, 278, 286, 287, 290, 296, 313, 335, 338, 341, 360, 361, 373, 374, 380–387, 445, 448, 450
PACKAGEReledpar	1, 4, 6, 8, 16, 18, 20, 38, 53, 60, 62, 63, 73, 74, 76, 78, 90, 97, 108, 113, 123, 124, 135, 137, 147, 148, 179, 183, 233, 250, 257, 273–275, 278, 309, 321, 360, 446, 447, 449–452, 455
PACKAGESuffix	80
PACKAGETabmac	1, 13, 388
PACKAGETHALIE	58
PACKAGEulem	38
PACKAGEuninormalize	32
PACKAGExargs	31, 80
PACKAGExkeyval	78, 273
PACKAGExparse	80, 452
PACKAGEXr	5, 64, 296, 297, 448
PACKAGEXref	296

PACKAGEXstring 80, 315

A

\absline@num 1
 Abu Kamil Shuja' b. Aslam 13
 \actionlines@list 1
 \actions@list 1
 \add@hookarg@specific@to@cs 1
 \add@hooktoggle@specific@to@cs 1
 \add@inserts 1
 \add@inserts@next 1
 \add@msd@ 1
 \add@msdata 1
 \add@msdata@firstlineofpage 1
 \add@penalties 1
 \add@Xgroupbyline 1
 \addtol@denbody 1
 Adelard II 13
 \advancelabel@refs 1
 \advanceline 1, 22
 \Aendnote 28
 \affixline@num 1
 \affixpstart@num 1
 \affixside@note 1
 \Afootnote 28
 \afterantilabe 58
 \afternoteX 51
 \afterruleX 52
 \ampersand 1, 58
 \annot@list 1
 \annotationref 1, 59
 \antilabe 1, 58
 \append@notesX 1
 \append@Xnotes 1
 \applabel 1, 62
 \appref 1, 62
 \apprefwithpage 1, 62
 \arrangementX 1, 40
 \arrangementX@normal 1
 \arrangementX@paragraph 1
 \arrangementX@threecol 1
 \arrangementX@twocol 1
 \at@every@pend 1
 \AtEndEveryPend 1, 18
 \AtEveryPend 1, 18
 \AtEveryPstart 1, 18
 \AtEveryStanza 1, 57
 \AtEveryStopStanza 1, 57
 \AtStartEveryPstart 1, 18
 \AtStartEveryStanza 1, 57
 \autopar 1, 17

B

\ballast 74

\ballast@count	1
Beeton, Barbara Ann Neuhaus Friend	18
\beforeantilabe	58
\beforeeiledchapter	1
\BeforeEveryStopStanza	1, 57
\beforeinsertingX	50
\beforeinsertion@X	1
\beforenotesX	52
\beginnumbering	1, 16
\Bendnote	28
\Bfootnote	28
\bhookgroupX	52
\bhooknoteX	50
\bodyfootmarkA	38
\boxfootnotenumbers	1
Bredon, Simon	13
Breger, Herbert	13, 341
Brey, Gerhard	13
Busard, Hubert L. L.	13
\bypage@false	1
\bypage@true	1
\bypstart@false	1
\bypstart@true	1

C

\c@addcolcount	1
\c@ballast	1
\c@firstlinenum	1
\c@firstsublinenum	1
\c@labidx	1
\c@linenumincrement	1
\c@sublinenumincrement	1
\Cendnote	28
\Cfootnote	28
\ch@ck@l@ck	1
\ch@cksub@l@ck	1
\chapter	1
\check@pb@in@verse	1
Chester, Robert of	13
Claassens, Geert H. M.	13
\colalignX	50
\continuousnumberingwithcolumns@sync@linenumber@singletext	1
Copernicus, Nicolaus	13
\critext	377
\ctab	1
\ctabtext	1
\current@annot	1

D

Dekker, Dirk-Jan	75
\Dendnote	28
\Dfootnote	28
\disable@endnotes	1
\disable@familiarnotes	1

<code>\disable@notes</code>	1
<code>\disable@sidenotes</code>	1
<code>\disablel@dtabfeet</code>	1
<code>\do@actions</code>	1
<code>\do@actions@fixedcode</code>	1
<code>\do@actions@next</code>	1
<code>\do@ballast</code>	1
<code>\do@feet@custom@order</code>	1
<code>\do@feetX</code>	1
<code>\do@insidelinehook</code>	1
<code>\do@line</code>	1
<code>\do@linehook</code>	1
<code>\do@lockoff</code>	1
<code>\do@lockoffL</code>	1
<code>\do@lockon</code>	1
<code>\do@lockonL</code>	1
<code>\do@Xfeet</code>	1
<code>\doedindexlabel</code>	1
<code>\doendnotes</code>	1, 29
<code>\doendnotesbysection</code>	1, 29
<code>\doinsidelinehook</code>	1, 26
<code>\doinsidethislinehook</code>	1, 27
<code>\dolinehook</code>	1, 26
<code>\dosplits</code>	1
Downes, Michael	75, 186, 188
<code>\dummy@edindex</code>	1
<code>\dummy@edtext</code>	1
<code>\dummy@edtext@showlemma</code>	1
<code>\dummy@ref</code>	1

E

<code>\edaftertab</code>	1, 70, 354
<code>edarrayc</code> (environment)	68
<code>edarrayl</code> (environment)	68
<code>edarrayr</code> (environment)	68
<code>\edatleft</code>	1, 70
<code>\edatright</code>	1, 70
<code>\edbeforetab</code>	1, 70, 354
<code>\edfilldimen</code>	1
<code>\edfont@info</code>	1
<code>\edindex</code>	1, 65
<code>\edindexlab</code>	1, 67
<code>\EDLABEL</code>	1
<code>\edlabel</code>	1, 59
<code>\edlabelE</code>	1, 61
<code>\edlabelS</code>	1, 61
<code>\edlabelSE</code>	1, 61
<code>\edlineref</code>	1, 59
<code>\edmakelabel</code>	1, 61
<code>\edpageref</code>	1, 59
<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</code> (environment)	68
<code>edtabularl</code> (environment)	68
<code>edtabularr</code> (environment)	68
<code>\EDTEXT</code>	1
<code>\edtext</code>	1, 27
<code>\edtext@notes</code>	1
<code>\edtext@notes@start</code>	1
<code>\edvertdots</code>	1, 71
<code>\edvertline</code>	1, 71
<code>\Eendnote</code>	28
<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, 21
<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
<code>minipage</code>	59
<code>Euclid</code>	13
<code>\extensionchars</code>	1, 73

F

<code>\f@x@l@cks</code>	1
-------------------------	---

Fairbairns, Robin	38
\first@linenum@out@false	1
\first@linenum@out@true	1
\firsthalf@antilabe@wd	1
\firstlinenum	1, 20
\firstseriesX@	1
\firstsublinenum	1, 20
\firstXseries@	1
\flag@end	1
\flag@end@later	1
\flag@end@RTL	1
\flag@start	1
\flag@start@later	1
\flag@start@RTL	1
\flagstanza	1, 58
\flush@notes	1
\fnpos	1, 39
Folkerts, Menso	13
\footfootmarkA	38
\footfudgefiddle	1, 75
\footnote	1
\footnoteA	37
\footnoteB	37
\footnoteC	37
\footnoteD	37
\footnoteE	37
\footnotelang@lua	1
\footnotelang@poly	1
\footnoteoptions@	1
\footnoteXmark	38
\footnoteXtext	38
\footplitskips	1
\fullstop	1

G

Gädeke, Nora	13
\get@edindex@hyperref	1
\get@edindex@ledinnote@command	1
\get@firsthalf@antilabe@wd	1
\get@fnmark	1
\get@fnmarkX	1
\get@index@command	1
\get@linelistfile	1
\get@pstartcounter	1
\get@stanzacounter	1
\get@sw@txt	1
\get@thisfootnote	1
\get@thisfootnoteX	1
\getline@num	1
\gl@p	1

H

\h@num	1
\hangindentX	49

\hidenumbering	1, 23
\hidenumberingonleftpage	1, 23
\hidenumberingonrightpage	1
\Hilfsbox	1
\hilfsbox	1
\hilfscount	1
\HILFSskip	1
\Hilfsskip	1
\hilfsskip	1
\hookarg@specific	1
\hooktoggle@specific	1
\hsizethreecolX	50
\hsizetwocolX	50
\Hy@raisedlink@left	1
\hyperlinkformat	1
\hyperlinkformatR	1
\hyperlinkR	1

I

\if@addsw	1
\if@edtabular	1
\if@edtext@secondarg@	1
\if@eled@sectioning	1
\if@firstlineofpage	1
\if@led@nofoot	1
\if@ledgroup	1
\if@lemmacommand@	1
\if@msdata@insertedfrompreviouspage	1
\if@noeled@sec	1
\if@noneed@notes	1
\if@RTL	1
\ifafterendnumberingR	1
\ifantilabe@	1
\ifat@every@pend@star@	1
\ifat@every@pstart@star@	1
\ifautopar@pause	1
\ifbypage@	1
\ifbypage@R	1
\ifbypstart@	1
\ifbypstart@R	1
\ifeledmacompat@	1
\iffirst@linenum@out@	1
\ifinastanzaL	1
\ifinastanzaR	1
\ifindtl@innote	1
\ifindtl@notenumber	1
\ifinserthangingsymbol	1
\ifinstanza	1
\ifinstanzaL	1
\ifinstanzaR	1
\ifl@d@dash	1
\ifl@d@elin	1
\ifl@d@esl	1
\ifl@d@pnun	1

<code>\ifl@d@ssub</code>	1
<code>\ifl@d@Xendlinenumber</code>	1
<code>\ifl@d@Xlinenumber</code>	1
<code>\ifl@d@Xmorethantwolines</code>	1
<code>\ifl@d@Xtwolines</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
<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>\ifparapparat@</code>	1
<code>\ifparledgroup</code>	1
<code>\ifprevpgnotnumbered</code>	1
<code>\ifpst@rtedL</code>	1
<code>\ifresumenumbering@start</code>	1
<code>\ifsameparallelpagenumber</code>	1
<code>\ifseriesbefore</code>	1
<code>\ifsidepstartnum</code>	1
<code>\ifstopmsdata@inserted@</code>	1
<code>\ifsublines@</code>	1
<code>\ifsublines@R</code>	1
<code>\ifwidthliketwocolumns</code>	1
<code>\ifXendinsertsep@</code>	1
<code>\ifxindy@</code>	1
<code>\ifxindyhyperref@</code>	1
<code>\initnumbering@quote</code>	1
<code>\initnumbering@reg</code>	1

\innotemarkX	66
\insert@count	0, 1
\insert@msdata	1
\insert@txtbeforenotesX	1
\insert@Xtxtbeforenotes	1
\inserthangingymbol	1
\insertlines@list	1
\insertparafootsepX	1
\inserts@list	1

J

Jayaditya	13
-----------------	----

K

Kabelschacht, Alois	174
\keep@this@crossref@forinserting	1

L

\l@advance@parledgroup@beforenormalnotes	1
\l@d@add	1
\l@d@nums	1
\l@d@section	1
\l@d@set	1
\l@d@Xend	1
\l@dampcount	1
\l@dbfnote	1
\l@dcheckcols	1
\l@dcheckstartend	1
\l@dchset@num	1
\l@dcolcount	1
\l@dcollect@@body	1
\l@dcollect@body	1
\l@dcolwidth	1
\l@dcsnote	1
\l@dcsnotetext	1
\l@dcsnotetext@l	1
\l@dcsnotetext@r	1
\l@ddodoreinxtrafeet	1
\l@ddoxtrafeet	1
\l@dedbeginmini	1
\l@dedendmini	1
\l@emptyd@ta	1
\l@dend@close	1
\l@dend@open	1
\l@dend@stuff	1
\l@dend@Xfalse	1
\l@dend@Xtrue	1
\l@denvbody	1
\l@desnote	1
\l@dfambeginmini	1
\l@dfamendmini	1
\l@dfeetbeginmini	1
\l@dfeetendmini	1
\l@dgetline@margin	1

<code>\l@getlock@disp</code>	1
<code>\l@getref@num</code>	1
<code>\l@getsidenote@margin</code>	1
<code>\l@gobbleoptarg</code>	1
<code>\l@disnote</code>	1
<code>\l@dlabel@parse</code>	1
<code>\l@dld@ta</code>	1
<code>\l@dlp@rbox</code>	1
<code>\l@dlsn@te</code>	1
<code>\l@dlsnote</code>	1
<code>\l@dmake@labels</code>	1
<code>\l@dmodforedtext</code>	1
<code>\l@dnnullfills</code>	1
<code>\l@dnumpstartsL</code>	1
<code>\l@doldold@footnotetext</code>	1
<code>\l@dp@rsefootspec</code>	1
<code>\l@dparsedendline</code>	1
<code>\l@dparsedendpage</code>	1
<code>\l@dparsedendsub</code>	1
<code>\l@dparsedstartline</code>	1
<code>\l@dparsedstartpage</code>	1
<code>\l@dparsedstartsub</code>	1
<code>\l@dparsefootspec</code>	1
<code>\l@dpush@begins</code>	1
<code>\l@drd@ta</code>	1
<code>\l@dref@undefined</code>	1
<code>\l@drestorefills</code>	1
<code>\l@dstoreforedtext</code>	1
<code>\l@drp@rbox</code>	1
<code>\l@drsn@te</code>	1
<code>\l@drsnote</code>	1
<code>\l@dsetmaxcolwidth</code>	1
<code>\l@dskipnumberfalse</code>	1
<code>\l@dskipnumbertrue</code>	1
<code>\l@dtabaddcols</code>	1
<code>\l@dtabnoexpands</code>	1
<code>\l@dunboxmpfoot</code>	1
<code>\l@dunhbox@line</code>	1
<code>\l@dzeropenalties</code>	1
<code>\l@noexpands@edgl</code>	1
<code>\l@secondmandarg</code>	1
<code>\l@wrap@ifnotemptybox</code>	1
<code>\l@wrapcs@ifnotemptybox</code>	1
<code>\label</code>	61
<code>\labelpstartfalse</code>	1
<code>\labelpstarttrue</code>	1, 19
<code>\labelref@list</code>	1
<code>\labelrefsparseline</code>	1
<code>\labelrefsparsesubline</code>	1
<code>\last@page@num</code>	1
Lavagnino, John	12
<code>\led@check@nopb</code>	1
<code>\led@check@pb</code>	1
<code>\led@err@AutoparNotNumbered</code>	1

\led@err@BadAction	1
\led@err@edtextoutsidepstart	1
\led@err@EdtextWithoutNote	1
\led@err@FootnoteNotInSecondArgEdtext	1
\led@err@HighEndColumn	1
\led@err@LineationInNumbered	1
\led@err@LowStartColumn	1
\led@err@ManyLeftnotes	1
\led@err@ManyRightnotes	1
\led@err@ManySidenotes	1
\led@err@NumberingNotStarted	1
\led@err@NumberingShouldHaveStarted	1
\led@err@NumberingStarted	1
\led@err@NumberingWithoutPstart	1
\led@err@PendInEdtext	1
\led@err@PendNoPstart	1
\led@err@PendNotNumbered	1
\led@err@PstartInEdtext	1
\led@err@PstartInPstart	1
\led@err@PstartNotNumbered	1
\led@err@ReverseColumns	1
\led@err@samewordRTL	1
\led@err@toendnotes@outsidenumbering	1
\led@err@TooManyColumns	1
\led@err@UnequalColumns	1
\led@error@fail@patch@@docclearpage	1
\led@error@fail@patch@@iiiminipage	1
\led@error@fail@patch@@makecol	1
\led@error@fail@patch@@reinserts	1
\led@error@fail@patch@addtocounter	1
\led@error@fail@patch@endminipage	1
\led@error@fail@patch@setcounter	1
\led@error@PackageAfterEledmac	1
\led@error@require@glossariesextra	1
\led@mess@NotesChanged	1
\led@mess@SectionContinued	1
\led@nopb	1
\led@nopbnum	1
\led@pb	1
\led@pb@setting	1
\led@pbnum	1
\led@reinit@index@fornote	1
\led@set@index@fornote	1
\led@toksa	1
\led@toksb	1
\led@warn@AppLabelOutSecondArgEdtext	1
\led@warn@BadAction	1
\led@warn@BadAdvancelineLine	1
\led@warn@BadAdvancelineSubline	1
\led@warn@BadHookSeries	1
\led@warn@BadLineation	1
\led@warn@BadLinenummargin	1
\led@warn@BadLockdisp	1
\led@warn@BadSetline	1

<code>\led@warn@BadSetlinenum</code>	1
<code>\led@warn@BadSidenotemargin</code>	1
<code>\led@warn@BadSublockdisp</code>	1
<code>\led@warn@DuplicateLabel</code>	1
<code>\led@warn@edinde@outsidenumbering</code>	1
<code>\led@warn@LineFileObsolete</code>	1
<code>\led@warn@NoFile</code>	1
<code>\led@warn@NoIndexFile</code>	1
<code>\led@warn@NoMarginpars</code>	1
<code>\led@warn@RefUndefined</code>	1
<code>\led@warn@SeriesStillExist</code>	1
<code>\led@warning@hsizeX@deprecated</code>	1
<code>\led@warning@msdatawithoutstop</code>	1
<code>\led@warning@preXnotes@deprecated</code>	1
<code>\led@warning@Xhsize@deprecated</code>	1
<code>ledgroup (environment)</code>	59
<code>ledgroupsize (environment)</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>	1, 21
<code>\leftlinenumannotation</code>	1
<code>\leftltab</code>	1
<code>\leftnoteupfalse</code>	65
<code>\leftpstartnum</code>	1
<code>\leftrtab</code>	1
<code>Leibniz</code>	13
<code>\lemma</code>	1, 30
<code>\letsforverteilen</code>	1

<code>\line@list</code>	1
<code>\line@list@stuff</code>	1
<code>\line@list@version</code>	1
<code>\line@margin</code>	1
<code>\line@num</code>	1
<code>\line@set</code>	1
<code>\lineannot</code>	1, 25
<code>\lineannot@set</code>	1
<code>\lineation</code>	1, 21
<code>\linenum</code>	1, 30
<code>\linenum@out</code>	1
<code>\linenumannotation</code>	1, 23
<code>\linenumannotationothersidettrue</code>	25
<code>\linenumberlist</code>	1, 20
<code>\linenumberstyle</code>	1, 22
<code>\linenumincrement</code>	1, 20
<code>\linenummargin</code>	1, 21
<code>\linenumr@p</code>	1
<code>\linenumrep</code>	1
<code>\linenumsep</code>	1, 21
<code>\linerangesep@</code>	1
<code>\list@clear</code>	1
<code>\list@clearing@reg</code>	1
<code>\list@create</code>	1
<code>\lock@disp</code>	1
<code>\lock@off</code>	1
<code>\lock@on</code>	1
<code>\lockdisp</code>	1, 22
Lorch, Richard	13
<code>\ltab</code>	1
<code>\ltabtext</code>	1
Luecking, Dan	79

M

<code>\m@mmf@check</code>	1
<code>\m@mmf@prepare</code>	1
<code>\M@sect</code>	1
<code>\makehboxofhboxes</code>	1
<code>\managestanza@modulo</code>	1
<code>\maxhnotesX</code>	52
Mayer, Gyula	13
<code>\measurebody</code>	1
<code>\measuremcell</code>	1
<code>\measuremrow</code>	1
<code>\measuretbody</code>	1
<code>\measuretbody</code>	1
<code>\measuretbody</code>	1
<code>\measuretbody</code>	1
<code>\measuretbody</code>	1
Middleton, Thomas	13, 104
minipage (environment)	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, 36
<code>\msdata@c</code>	1
<code>\msdata@cR</code>	1
<code>\multfootsep</code>	1, 38
<code>\multiplefootnotemarker</code>	1

N

<code>\n@num</code>	1
<code>\n@num@stanza</code>	1
<code>\new@line</code>	1
<code>\newhookarg@specific</code>	1
<code>\newhookcommand@series</code>	1
<code>\newhookcommand@series@reload</code>	1
<code>\newhooktoggle@series</code>	1
<code>\newhooktoggle@series@reload</code>	1
<code>\newhooktoggle@specific</code>	1
<code>\newseries@</code>	1
<code>\newverse</code>	1
<code>\NEXT</code>	1
<code>\next@line@list@stuff</code>	1
<code>\no@expands</code>	1
<code>\nomk@</code>	1
<code>\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>	57
<code>\numberstanzatrue</code>	57
<code>\numlabfont</code>	1, 53

O

<code>\old@hsize</code>	1
<code>\one@line</code>	1
<code>optionafter</code>	25
<code>optionantilabe</code>	58, 328
<code>optionauxdir</code>	15, 449
<code>optionbefore</code>	25
<code>optioncontinuousnumberingwithcolumns</code>	90, 155, 162, 448, 452, 455
<code>optiondraft</code>	73
<code>optionfinal</code>	73
<code>optioninnnote</code>	447
<code>optioninnnote</code>	447
<code>optionlinerrangesep</code>	273
<code>optionminus</code>	448
<code>optionnocritical</code>	445, 447
<code>optionnoeledsec</code>	72, 366, 449
<code>optionnoend</code>	447
<code>optionnofamiliar</code>	445, 453
<code>optionnopenalties</code>	75, 449
<code>optionnoprefix</code>	64, 453
<code>optionnoresetlinenumannotation</code>	23, 44, 453
<code>optionnotenumber</code>	447, 450
<code>optionparapparatus</code>	455
<code>optionplus</code>	448
<code>optionprefixmore</code>	64, 453
<code>optionsameparallelpagenum</code>	452
<code>optionswcaseinsensitive</code>	32, 451

P

<code>\page@action</code>	1
<code>\page@num</code>	1

\pagelinesep	1, 66
\pageref	61
\par@line	1
\par@patch@pagenumbering	1
\par@patch@thepage	1
\para@footgroupX	1
\para@footsetup	1
\para@footsetupX	1
\para@vfootnoteX	1
\parafootfmt	1
\parafootfmtX	1
\parafootgroup	1
\parafootsepX	51
\parafootstart	1
\parafootstartX	1
\paravfootnote	1
\parindentX	49
\parse@annot	1
\pausenumbering	1, 19
\pausenumbering@page@num	1
\pend	1, 17
Plato of Tivoli	13
\postbodyfootmark	1
\prebodyfootmark	1
\prenotesX	52
\prepare@edindex@fornote	1
\prepare@prenotesX	1
\prepare@Xgroupbyline	1
\prepare@Xprenotes	1
\prev@nopb	1
\prev@pb	1
\prevpage@num	1
\print@eledsection	1
\print@footnoteXrule	1
\print@leftmargin@eledsection	1
\print@lemma	1
\print@line	1
\print@notesX	1
\print@rightmargin@eledsection	1
\print@Xfootnoterule	1
\print@Xnotes	1
\printendlines	1
\printlineendnote	1
\printlineendnotearea	1
\printlinefootnote	1
\printlinefootnotearea	1
\printlinefootnotenumbers	1
\printlines	1
\printlines@count	1
\printnpnum	1
\printpstart	1
\printsymlineendnotearea	1
\printsymlinefootnotearea	1
\printXafternumber	1

\printXbeforenumber	1
\pstart	1, 17
\pstarteref	1
\pstartnum	1
\pstartref	59
\pstarts@read@L	1
\pstarts@typeset@L	1

Q

\quotation	1
\quote	1

R

\raggedX	51
\raw@text	1
\rbracket	1
\read@linelist	1
\ref	61
\ref@reg@later	1
\Relax	1
\reledmac@error	1
\reledmac@warning	1
\removehboxes	1
\reset@current@cannot	1
\reset@msd@options@iffullpage	1
\resetlinenumannotation	1
\resetprevline@	1, 107
\resetprevpage@	1
\resetprevpage@num	107
\restore@endnotes	1
\restore@familiarnotes	1
\restore@notes	1
\restore@sidenotes	1
\resumenumbering	1, 19
\rightctab	1
\rightlinenum	1, 21
\rightlinenumannotation	1
\rightltab	1
\rightnoteupfalse	65
\rightrtab	1
\rightstartnum	1
\rigidbalance	1
\rigidbalanceX	1
\rtab	1
\rtabtext	1

S

Sacrobosco	13
\sameword	1, 31
\sameword@inedtext	1
Schöpf, Rainer	13
\section@num	1
\select@lemmafонт	1
\select@lemmafонт	1, 53

<code>\Seref</code>	1, 61
<code>\Serefonlypage</code>	61
<code>\Serefwithpage</code>	1, 61
<code>\series</code>	1
<code>\seriesatbegin</code>	1, 39
<code>\seriesatend</code>	1, 39
<code>\set@continuousnumberingforL</code>	1
<code>\set@line</code>	1
<code>\set@line@action</code>	1
<code>\set@this@cc@page</code>	1
<code>\set@txtbeforenotesX</code>	1
<code>\set@Xtxtbeforenotes</code>	1
<code>\setapprefprefixmore</code>	62
<code>\setapprefprefixsingle</code>	62
<code>\setcommand@series</code>	1
<code>\sethangindentX</code>	1
<code>\sethangingsymbol</code>	1, 56
<code>\setistwofollowinglines</code>	1
<code>\setl@dlp@rbox</code>	1
<code>\setl@drpr@box</code>	1
<code>\setline</code>	1, 22
<code>\setlinenum</code>	1, 22
<code>\setlinenumannotationsep</code>	1, 26
<code>\setmcellcenter</code>	1
<code>\setmcellleft</code>	1
<code>\setmcellright</code>	1
<code>\setmrowcenter</code>	1
<code>\setmrowleft</code>	1
<code>\setmrowright</code>	1
<code>\setmsdatalabel</code>	1, 37
<code>\setmsdataposition</code>	1, 37
<code>\setmsdataseries</code>	1, 37
<code>\setnoteswidthliketwocolumnsX@</code>	1
<code>\setnotesXpositionliketwocolumns@</code>	1
<code>\setparindentX</code>	1
<code>\setprintendlines</code>	1
<code>\setprintlines</code>	1
<code>\setSErefonlypageprefixmore</code>	62
<code>\setSErefonlypageprefixsingle</code>	62
<code>\setSErefprefixmore</code>	62
<code>\setSErefprefixsingle</code>	62
<code>\setsidenotesep</code>	65
<code>\setstanzaindents</code>	1, 54
<code>\setstanzapenalties</code>	1, 56
<code>\setstanzavalues</code>	1
<code>\settcellcenter</code>	1
<code>\settcellleft</code>	1
<code>\settcellright</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, 55
<code>\stanzaindent*</code>	1, 55
<code>\stanzaindentbase</code>	1, 54
<code>\stanzanumwrapper</code>	1, 57
<code>\startlock</code>	1, 22
<code>\startsub</code>	1, 21
<code>\stepl@dcolcount</code>	1
<code>\stopmsdata</code>	1, 36
<code>\store@annot@to@absline</code>	1
<code>\strip@szacnt</code>	1
<code>\sub@action</code>	1
<code>\sub@lock</code>	1
<code>\sub@off</code>	1
<code>\sub@on</code>	1
<code>\subline@num</code>	1
<code>\sublinenumberstyle</code>	1, 22
<code>\sublinenumincrement</code>	1, 20
<code>\sublinenumr@p</code>	1
<code>\sublinenumrep</code>	1
<code>\sublineref</code>	1, 59
<code>\sublines@false</code>	1
<code>\sublines@true</code>	1
<code>\sublock@disp</code>	1
<code>\sublockdisp</code>	1
Sullivan, Wayne	13, 74, 89, 94, 186, 187, 281, 321
<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>\thefootnoteA</code>	38
Theodosius	13
<code>\thepageline</code>	1
<code>\thepar@page</code>	1

<code>\thepstart</code>	1, 19
<code>\thestanza</code>	1, 57
<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, 29
<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>	51
<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
<code>\vl@dcsnote</code>	1
<code>\vl@disnote</code>	1
<code>\vl@dlsnote</code>	1
<code>\vl@dosnote</code>	1
<code>\vl@drsnote</code>	1
<code>\vnumfootnoteX</code>	1

W

Whitney, Ron	13
<code>\widthX</code>	53
<code>\wrap@edcrossref</code>	1
<code>\wrapcontentX</code>	49
<code>\wrapped@bodyfootmarkX</code>	1
<code>\wrapped@footfootmarkX</code>	1
<code>\write@lemma</code>	1
Wujastyk, Dominik	12

X

\X@atbegininsertion	1
\X@beforeinsertion	1
\X@doreinfeet	1
\xabslineref	1
\Xafterlemmaseparator	47
\Xafternote	51
\Xafternumber	45
\Xafterrule	52
\Xaftersymlinenum	45
\xannotationref	1, 60
\Xarrangement	1, 40
\Xarrangement@normal	1
\Xarrangement@paragraph	1
\Xarrangement@threecol	1
\Xarrangement@twocol	1
\Xbeforeinserting	50
\Xbeforelemmaseparator	47
\Xbeforenotes	52
\Xbeforenumber	45
\Xbeforepagenumber	42
\Xbeforesymlinenum	45
\Xbhookgroup	52
\Xbhooknote	50
\Xboxlinenum	46
\Xboxlinenumalign	46
\Xboxsymlinenum	46
\Xcolalign	50
\xedindex	1
\xedlabel	1
\xedtext	1
\Xendafterlemmaseparator	47
\Xendafternote	53
\Xendafterpagenumber	42
\Xendaftersymlinenum	45
\Xendaftrenumber	45
\Xendahookinplaceofnumber	49
\Xendahooklinenumber	49
\Xendbeforelemmaseparator	47
\Xendbeforenumber	45
\Xendbeforepagenumber	42
\Xendbeforesymlinenum	45
\Xendbhookinplaceofnumber	49
\Xendbhooklinenumber	49
\Xendbhooknote	50
\Xendboxendlinenum	47
\Xendboxlinenum	47
\Xendboxlinenumalign	47
\Xendboxstartline	47
\Xendboxsymlinenum	46
\Xendhangindent	49
\Xendinplaceoflemmaseparator	47
\Xendinplaceofnumber	46
\Xendinplaceofpagenumber	42

\Xendlemmadisablefontselection	48
\Xendlemmafont	48
\Xendlemmaseparator	47
\Xendlinenumannotationonlyfirst	45
\Xendlinenumannotationonlyfirstintwo	45
\Xendlinenumannotationposition	25
\Xendlineprefixmore	42
\Xendlineprefixsingle	42
\Xendlinerrangeseparator	43
\Xendmorethantwolines	43
\Xendnoidenticallinenumannotation	26
\Xendnolinenumberifannotation	45
\Xendnonumber	43
\Xendnotefontsize	48
\Xendnotenumfont	48
\Xendnumberonlyfirstinline	41
\Xendnumberonlyfirstintwolines	41
\Xendpagenumberonlyfirst	42
\Xendparagraph	53
\Xendsep	53
\Xendstorelineinfo	1
\Xendsublinesep	45
\Xendsymlinenum	41
\Xendsymlinenumannotation	45
\Xendsympagenum	42
\Xendtwolines	43
\Xendtwolinesbutnotmore	43
\Xendtxtbeforenotes	51
\Xendwrapcontent	49
\xflagref	1
\Xgroupbyline	51
\Xgroupbylineseparetwolines	51
\Xhangindent	49
\Xhsizethreecol	50
\Xhsizetwocol	50
\Xinnotemark	66
\Xinplaceoflemmaseparator	47
\Xinplaceofnumber	46
\Xinsertparafootsep	1
\Xledsetnormalparstuff	1
\xleft@appenditem	1
\Xlemmadisablefontselection	48
\Xlemmafont	48
\Xlemmaseparator	47
\Xlinenumannotationonlyfirst	44
\Xlinenumannotationonlyfirstintwo	44
\Xlinenumannotationposition	25
\Xlinenumannotationposition@side	1
\Xlinenumannotationpositionside	25
\Xlinerrangeseparator	43
\xlineref	1, 60
\Xmaxhnotes	52
\Xmorethantwolines	43
\Xnoidenticallinenumannotation	26

\Xnolemmaseparator	1, 47
\Xnolinenumannotation	45
\Xnonbreakableafternumber	45
\Xnonote	41
\Xnonumber	43
\Xnopagenumberifcurrent	43
\Xnotboxingsubline	46
\Xnotefontsize	48
\Xnotenumfont	48
\Xnoteswidthliketwocolumns	53
\Xnumberonlyfirstinline	41
\Xnumberonlyfirstintwoline	41
\Xonlypstart	44
\Xpagelinesep	45
\xpageref	1, 60
\Xparafootsep	51
\Xparindent	49
\Xprenotes	1, 52
\Xprenotes@	1
\Xpstart	44
\Xpstarteverytime	44
\Xpstartonlyfirst	44
\xpstartref	1, 60
\Xpstartseparator	44
\XR@test	1
\XR@test@mac	1
\XR@test@mac@test	1
\Xragged	51
\xright@appenditem	1
\Xrigidbalance	1
\Xsethangindent	1
\Xsetparindent	1
\Xstanza	44
\Xstanzaonlyfirst	44
\Xstanzaseparator	44
\Xstorelineinfo	1
\xsublineref	1, 60
\Xsublinesep	22, 45
\Xsublinesepside	22
\Xsymlinenum	41
\Xsymlinenumannotation	44
\Xtoendnotes	29
\Xtwoline	43
\Xtwolineonlyinsamepage	43
\Xtxtbeforenotes	51
\Xtxtbeforenotesonlyonce	51
\Xtxtbeforenumber	42
\Xunvxh	1
\Xwidth	53
\Xwrapcontent	49
\Xwrapendlemma	48
\Xwraplemma	48
\Xwraplinenumannotation@ref	1
\Xwraplinenumannotation@side	1

\xxref 1, 60

Z

\zz@@@ 1

\zz@@@@ 1

Change History

v0.1.0.	
General: First public release	1
v0.2.0.	
\ifl@dmemoir: Added \ifl@dmemoir for memoir class having been used	81
\morenoexpands: Added \l@dtabnoexpands to \no@expands	132
\reledmac@error: Added \eledmac@error and replaced error messages	82
General: Added tabmac code, and extended indexing	1
v0.2.1.	
\@lab: Removed page setting from \@lab	284
\edlabel: Tweaked \edlabel to get correct page numbers	281
\l@ddodoreintrafeet: Renamed \dodoreintrafeet to \l@ddodoreintrafeet	275
\l@ddoxtrafeet: Renamed \doxtrafeet to \l@ddoxtrafeet	273
\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	132
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	276
v0.2.2.	
\footfudgefiddle: Added \footfudgefiddle	184
\next@line@list@stuff: Added initial write of page number in \line@list@stuff	123
\para@footsetup: Added \footfudgefiddle to \para@footsetup	185
\para@footsetupX: Added \footfudgefiddle to \para@footsetupX	228
General: Improved paragraph footnotes	1
New Dekker example	1
Used \providecommand for \@gobblethree to avoid clash with the amsfonts package	88
v0.3.0.	
\@lab: Replaced \the\line@num by \linenumr@p\line@num in \@lab, and similar for sub-lines	284
\@nl@reg: Added a bunch of code to \@nl for handling \setlinenum	111
\ledlinenum: Added \linenumr@p and \sublinenumr@p to \leftlinenum and \rightlinenum	102
\linenumberlist: Added \linenumberlist mechanism	89
\printendlines: Added \linenumr@p and \sublinenumr@p to \printendlines	246
\printlines@count: Added \linenumr@p and \sublinenumr@p to \printlines	211
\sublinenumr@p: Added \linenumberstyle and \sublinenumberstyle	101
General: Includes edstanza and more	1
v0.3.1.	
General: Not released. Added remarks about the parallel package	1
v0.4.0.	
\@iiiminipage: Modified kernel \@iiiminipage and \endminipage to cater for critical footnotes	307
\Xarrangement@normal: Added minpage footnote setup to \footnormal	180
\edtext: Added \showlemma to \edtext	133
\l@dfeetendmini: Added \l@dfeetbeginmini, \l@dfeetendmini and all their supporting code	305
\mpnormalfootgroup: Added \mpnormalfootgroup	183
\mpnormalvfootnote: Added \mpnormalvfootnote	181
\showlemma: Added \showlemma	89
General: Added final/draft options	78
Added ledgroup environment	308

Added ledgroupsize environment	308
Added minipage, etc., support	1
v0.4.1.	
\do@Xfeet: Changed \do@Xfeet code for easier extensions	274
\edindex: Let eledmac take advantage of memoir's indexing	313
\print@Xnotes: Added \op@Xfeet	274
General: Added code for changing \@docclearpage	277
Not released. Minor editorial improvements and code tweaks	1
Only change \@footnotetext and \@footnotemark if memoir not used	214
v0.5.0.	
\@footnotetext: Enabled regular \footnote in numbered text	215
\@xympar: Eliminated \marginpar disturbance	297
General: Added left and right sidenotes	297
Added sidenotes, familiar footnotes in numbered text	1
v0.5.1.	
\affixline@num: Changed \affixline@num to cater for sidenotes	165
\l@getsidenote@margin: Added \sidenotemargin and \sidenote@margin	298
General: Added moveable sidenote	297
Fixed right line numbers killed in v0.5	1
Only change \hsize in ledgroupsize environment otherwise page number can be in wrong place	308
v0.6.0.	
\@lopR: Added \@pend, \@pendR, \@lopL and \@lopR in anticipation of parallel processing	113
\@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	215
\set@this@page: Extended \new@line to output page numbers	124
General: Fixed long paragraphs looping	1
Fixed minor typos	1
Prepared for eledpar package	1
v0.7.0.	
\@nl@reg: Added \@nl@reg	111
\@ref@reg: Added \@ref@reg	120
\affixline@num: Added skipnumering to \affixline@num	165
\do@actions@fixedcode: Added \do@actions@fixedcode	161
\do@actions@next: Added number skipping to \do@actions	159
\do@insidelinehook: Added \do@linehook for use in \do@line	157
\endnumbering: Changed \endnumbering for eledpar	93
\f@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	178
\get@linelistfile: Added \get@linelistfile	109
\initnumbering@reg: Added \initnumbering@reg	92
\l@advance@parledgroup@beforenormalnotes: Added \l@dunboxmpfoot containing some common code	307
\l@dcstotetext@r: Added \l@emptyd@ta	157
\l@dgetline@margin: Added \l@dgetline@margin	98
\l@dgetlock@disp: Added \l@dgetlock@disp	100
\l@dgetsidenote@margin: Added \l@dgetsidenote@margin	298
\l@dnumpstartsL: Added \l@dnumpstartsL, \ifl@dpairing and \ifpst@rted for/from eledpar	90
\l@drsn@te: Added \l@dlsn@te and \l@drsn@te for use in \do@line	158
\l@dzeropenalties: Added \l@dzeropenalties	152
\ledlinenum: Added \ledlinenum for use by \leftlinenum and \rightlinenum	102

\list@clearing@reg: Added \list@clearing@reg	109
\n@num: Added \n@num	118
\next@line@list@stuff: Deleted \page@start from \line@list@stuff	123
\normalbfnoteX: Removed extraneous space from \normalbfnoteX	220
\pausenumbering@page@num: Changed \resumenumbering for eledpar	95
\setprintlines: Added \setprintlines for use by \printlines	205
\skipnumbering: Added \skipnumbering and supports	129
\sublinenumincrement: Added \firstlinenum, \linenumincrement, \firstsublinenum and \linenumincrement	100
\sublinenumr@p: Using \linenumrep instead of \linenumr@p	101
Using \sublinenumrep instead of \sublinenumr@p	101
\vnumfootnoteX: Removed extraneous space from \vnumfootnoteX	222
General: Added \setprintendlines for use by \printendlines	242
eledmac having been available for 2 years, deleted the commented out original edmac texts	1
Maieul 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
v0.8.0.	
General: Bug on endnotes fixed: in a // text, all endnotes will print and be placed at the ends of columns ()	1
v0.8.1.	
General: Bug on \edtext ; \critex ; \lemma fixed: we can now us non-switching commands	1
v0.9.0.	
General: No more ledpatch. All patches are now in the main file.	1
v0.9.1.	
General: Fix some bugs linked to integrating ledpatch on the main file.	1
v0.10.0.	
General: Corrections to \section and other titles in numbered sections	1
v0.11.0.	
General: Makes it possible to add a symbol on each verse's hanging, as in French typography. Redefines the command \hangingsymbol to define the character.	1
v0.12.0.	
\l@dnumpstartsL: Added \ifledRcol and \ifnumberingR for/from eledpar	90
General: For compatibility with eledpar, possibility to use \autopar on the right side.	1
Possibility to number \pstart.	19
Possibility to number the pstart with the commands \numberpstarttrue.	1
v0.12.1.	
General: Do not number \pstarts of stanza.	1
The numbering of \pstarts restarts on each \beginnumbering.	1
v0.13.0.	
\managestanza@modulo: New stanzaindentsrepetition counter to repeat stanza indents every <i>n</i> verses.	323
General: New stanzaindentsrepetition counter to repeat stanza indents every <i>n</i> verses.	55
New stanzaindentsrepetition counter: to repeat stanza indents every <i>n</i> verses.	1
v0.13.1.	
General: \thepstartL and \thepstartR use now \bfseries and not \bf, which is deprecated and makes conflicts with memoir class.	1
v0.14.0.	
\edlabel: Tweaked \edlabel to get correct line number if the command is first element of a paragraph.	281

General: Tweaked <code>\edlabel</code> to get correct line number if the command is first element of a paragraph.	1
v0.15.0.	
<code>\affixline@num</code> : Line numbering can be disabled.	165
<code>\ifinserthangingsymbol</code> : New management of hangingsymbol insertion, preventing undesirable insertions.	321
<code>\printlines@count</code> : Line numbering can be reset at each <code>pstart</code>	209
General: Line numbering can be reset at each <code>pstart</code>	97
Possibility to print <code>\pstart</code> number at the sides of the text.	19
v0.17.0.	
<code>\ifinserthangingsymbol</code> : New new management of hangingsymbol insertion, preventing undesirable insertions.	321
v1.0.0.	
<code>\morenoexpands</code> : Change to be compatible with new features	132
General: <code>\lemma</code> can contain commands.	30
Debug in lineation command	21
New generic commands to customize footnote display.	40, 263
Options <code>nonum</code> and <code>nosep</code> in <code>\Xfootnote</code>	28
Options of <code>\Xfootnotes</code>	176
Possibility to have commands in sidenotes.	64
Some compatibility break with <code>eledmac</code> . Change of name: <code>eledmac</code>	1
v1.0.1.	
General: Correction on <code>\Xnumberonlyfirstinline</code> with lineation by <code>pstart</code> or by page.	41
v1.1.0.	
<code>\Xprenotes</code> : New skip <code>\Xprenotes@</code>	235
<code>\settoggle@series</code> : <code>\settoggle@series</code> switch the global value of the toggle, not only the local value.	264
General: Add <code>\labelpstarttrue</code>	19
Add <code>\Xnumberonlyfirstintwolines</code>	41
Add <code>\Xpstart</code> and <code>\Xonlypstart</code>	44
New hook to add arbitrary code at the beginning of the notes	49
New options for block of notes.	51
New package option: <code>parapparatus</code>	1
New tools to change order of series	262
Sectioning commands.	71
v1.2.0.	
<code>\Xprenotes</code> : Debug in familiar footnotes (bug introduced by v1.1).	235
<code>\endquote</code> : Compatibility of <code>\ledchapter</code> with the <i>memoir</i> class.	359
v1.3.0.	
<code>\endquote</code> : <i>Quotation</i> and quote environment inside numbered sections.	359
v1.4.0.	
<code>\edtext</code> : Compatibility of <code>\edtext</code> with the right-to-left direction (with Polyglossia).	133
<code>\ledsetnormalparstuffX</code> : Direction of footnotes with polyglossia.	232
<code>\newseries@</code> : Remembers the language of the lemma, in order to create a correct direction for the footnote separator.	253
<code>\rbracket</code> : Switch the right bracket to a left bracket when the lemma is RTL (needs polyglossia or LuaTeX).	198
General: Compatibility with LuaTeX of RTL notes.	1
v1.4.1.	
<code>\affixside@note</code> : Remove spurious spaces.	303
<code>\endquote</code> : New option <i>noquotation</i>	359
<code>\get@thisfootnote</code> : Compatibility of standard footnotes with <code>eledmac</code> when these footnotes contain any commands.	215
<code>\labelrefsparsesubline</code> : Fix a bug with <code>\edlabel</code>	283

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>	359
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.	220
v1.5.0.	
<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>	313
General: Correct indexing when the call is made in critical notes.	309
v1.5.1.	
<code>\managestanza@modulo</code> : Correct <code>stanzaindentsrepetition</code> counter	323
<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.	326
v1.6.1.	
<code>\AtStartEveryPstart</code> : Spurious space in <code>\pstart</code>	148
<code>\ifinserthangingsymbol</code> : Hang verse is now not automatically flush right.	321
<code>\l@dunhbox@line</code> : Move the call to <code>\inserthangingsymbol</code> to allow use <code>\hfill</code> inside.	153
<code>\pend</code> : Spurious space in <code>\pend</code>	150
General: Corrects a false hanging verse when a verse is exactly the length of a line.	1
v1.7.0.	
General: New features for managing page breaks.	73
v1.8.0.	
<code>\endquote</code> : Correction of sectioning commands in parallel texts.	359
<code>\get@index@command</code> : Debug <code>\get@index@command</code> and compatibility with <code>hyperref</code> package.	312
<code>\newhookcommand@series@reload</code> : Debug <code>\beforenotesX</code> and <code>\maxhnotesX</code> which did not work.	266
<code>\prevpage@num</code> : Correct <code>\parafootsep</code> when using with <code>ledgroup</code>	190
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.	309
v1.8.1.	
General: Debug endnotes when more than one series is used (change the position where tools for endnotes are defined).	237
v1.8.2.	
General: Debug compatibility problem with <code>hebrew</code> option of <code>babel</code> package.	1
v1.8.3.	
General: Fix spurious spaces added by v1.7.0.	1

v1.8.5.	
General: Debug indexing in right column, with eledpar.	309
v1.9.0.	
\l@ddoxtrafeet: Add \fnpos to choice the order of footnotes.	273
\l@dfeetendmini: Add \mpfnpos to choice the order of footnotes in minipage /	
ledgroup.	305
v1.10.0.	
\endquote: Correction of sectioning commands in parallel texts.	359
General: Add \pstartref and \xpstartref to refer to a pstart number (extension of	
\edlabel).	1
v1.10.1.	
General: Compatibility with cleveref.	1
v1.10.2.	
General: Compatibility of stanza with v1.8a of babel-greek.	1
v1.10.3.	
General: Debug of cross-referencing.	1
v1.10.4.	
General: Debug of critical notes in edtabular environment.	1
v1.10.5.	
General: Debug of \pausenumbering.	1
Debug of \xxref.	1
v1.10.6.	
General: Debug of interaction between \autopar and \pausenumbering.	1
v1.11.0.	
General: Add hooks to disable the font selection for lemma in footnote.	48
v1.11.1.	
General: Correct a bug when a critical note starts with plus or minus.	1
v1.12.0.	
\@nl@reg: To ensure compatibility with \musixtex, \@l becomes \@1. Consequently,	
\@l@reg becomes \@nl@reg.	110
\AtStartEveryPstart: New optional argument for \pstart, to execute code before it.	148
\edindex: Use correctly default index when imakeidx is loaded.	313
\endquote: \ledxxx sectioning commands are deprecated and replaced by \eledxxx	
commands.	359
\initnumbering@reg: \beginnumbering is defined only on eledmac, not on eledpar.	92
\l@dgetsidenote@margin: \sidenotemargin is now directly defined in eledmac to	
be able to manage eledpar.	298
\l@disnote: \l@dlsnote, \l@drsnote and \l@dcsnote defined only one time, in	
eledmac, including needs for eledpar case.	299
\l@dnumpstartsL: Add \ifledRcol@ for eledpar	90
\l@dunhbox@line: \do@line is split in more little commands.	154
\newhookcommand@series@reload: Debug \beforenotesX and \maxhnotesX which	
did not work when called after \footparagraphX.	266
Debug \Xbeforenotes and \Xmaxhnotes which did not work when called after	
\footparagraph.	266
\pend: New optional argument for \pend, to execute code after it.	150
\stanza: &can have an optional argument: content to be printed after.	326
\Stanza can have an optional argument: content to be printed before.	326
Add \newverse macro, \falseverse deprecated.	326
General: Add \ledinnernote and \ledouternote commands.	64
Add \Xendparagraph and related settings.	53
Add hyperlink to crossref (needs hyperref package).	59
Compatibility with musixtex.	1
Debug eledmac sectioning command after using \resumenumbering.	1

Ensure that imakeidx is loaded <i>before</i> eledmac	309
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
v1.12.1.	
<code>\wrap@edcrossref</code> : Fix spurious spaces.	286
v1.12.2.	
<code>\l@dunhbox@line</code> : Fix a bug with critical notes at the tops of pages (added by v12.0.0)	153
v1.12.3.	
<code>\flag@end</code> : <code>\flag@start</code> and <code>\flag@end</code> are now defined only one time for eledmac and eledpar	125
<code>\flag@start</code> send a error message when a <code>\edtext</code> is done without insert (note)	125
<code>\reledmac@error</code> : Replaced error messages	82
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
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	315
<code>\@ssect</code> : Debug <code>\eledchapter</code> in parallel with <code>memoir</code>	362
<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@dggobbleoptarg</code> : <code>\l@dggobblearg</code> becomes <code>\l@dggobbeloptarg</code>	344
<code>\l@drestoreforedtext</code> : Debug optional arguments of <code>\Xfootnote</code> in tabular context	344
<code>\pausenumbering@page@num</code> : Debug <code>\resumenumbering</code>	95
v1.12.7.	
<code>\wrap@edcrossref</code> : <code>\wrap@edcrossref</code> is now robust	286
v1.12.8.	
<code>\flag@end</code> : <code>\flag@start</code> do not send a error message when a <code>\edtext</code> is done without insert (note) but have a endnote	125
v1.13.0.	
<code>\newhooktoggle@series@reload</code> : Add <code>\newhookcommand@toggle@reload</code>	266
<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.	264
General: Add <code>\Xnoteswidthliketwocolumns</code> and <code>\noteswidthliketwocolumnsX</code>	53
Added <code>widthliketwocolumns</code> option	78
v1.13.1.	
<code>\ifat@every@pstart@star@</code> : Add <code>\l@dzeropenalties</code> in <code>\pstart</code>	148
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
v1.13.2.	
<code>\l@dnumpstartsL</code> : Add <code>\ifl@dpaging</code> for eledpar	90
General: Fix a bug with normal footnotes, added by v1.13.0.	1
v1.13.3.	
General: Fix extra spaces with paragraphed footnotes, added by v1.13.0.	1

v1.13.4.	
General: Fix a bug with index when memoir class is used without hyperref	1
v1.14.0.	
\edindex: Let eledmac take advantage of imakeidx even when memoir class is used	313
General: Debug spurious characters before endnotes.	237
Delete previous override of \l@d@wrindexhyp at the beginning of a document	
when hyperref is not loaded.	317
Move gobbling command	88
Provide \gobblefour	88
v1.14.1.	
\sssect: Debug sectioning commands when using both handout and hyperref	
package.	364
v1.14.2.	
\sssect: Debug \edtext after starred sectioning commands when using memoir class.	362
v1.15.0.	
\edtext@level: New boolean \ifedtext@.	133
\arrangementX@paragraph: Correct bug with paragraphed familiar footnotes setting.	227
\endsub: Restore subline feature (disabled by mistake in v1.8.0).	127
\if@lemmacommand@: New boolean \iflemmacommand@.	139
General: Fix a bug with footnotes layout when using some options of the geometry	
package (bug add by v1.13.0).	1
New commands \AtEveryPstart and \AtEveryPend.	18
New tools to prevent ambiguous references in lemma	31
v1.15.1.	
\next@line@list@stuff: Revert modification of 1.5.2 which makes bug with	
numbering. Leave vertical mode to solve spurious space before minipage.	123
v1.16.0.	
General: \edtext is now defined only in eledmac, not in eledpar. Debug wrong	
numbering when using \sameword + eledpar + \tag command.	133
Compatibility of standard footnotes with some biblatex styles.	1
New \stanzaindent command.	1
v1.16.1.	
\xlineref: \lineref is not defined if defined by some other package, like lineno.	
Eledmac provides \edlineref instead.	287
v1.17.0.	
\edtext: Error message when calling \edtext outside of a numbered paragraph.	133
v1.18.0.	
\edindex@hyperref: Fix spurious space with \edindex when using	
imakeidx/indextools + hyperref.	315
\edlabel: \edlabel is now defined only one time for both eledmac and eledpar	281
\l@d@section: Option parapparatus works for endnotes.	237
\l@dnumprintstartL: Add \ifl@dprintingpages and \dprintingcolumns for	
eledpar	90
\print@line: Compatibility with Lua [®] TeX RTL languages.	154
\printlinefootnote: Code refactoring in \printlinefootnote: the printing of the	
numbers are factorized in \printlinefootnotearea	200
\printpstart: Debug \Xpstart with parallel pages and columns (eledpar)	199
General: Add \Xpstarteverytime	44
Compatibility with Lua [®] TeX RTL languages.	1
Debug \Xonlypstart when using \Xnumberonlyfirstinline and the current line	
number differs from the previous.	44
v1.19.0.	
\footplitskips: \footplitskips doesn't set \floatingpenalty to \@MM when	
processing parallel pages.	178

\xxref: \xxref works also with right side numbers, when \@Rlineflag is not empty.	289
General: \Xmaxhnotes and \maxhnotesX work now for both two-columns and three-columns setting.	1
Compatibility with eledpar v.1.13.0.	1
v1.19.1.	
General: Call \correct@footinsX@box and \correct@Xfootins@box directly in \print@notesX@forpages and \print@Xnotes@forpages, that is in eledpar.	1
v1.20.0.	
\printlines@count: Added \ifl@d@Xmorethantwolines and \ifl@d@Xmorethantwolines to \printlines	211
\stanza: & and \& can be preceded by spaces.	326
\xxref: Debug \xxref when not loading eledpar (Fix a bug added in 1.19.0).	289
General: Add \Xendboxlinenum	47
Add \Xtwolines and \Xmorethantwolines hooks	43
Add series option.	1
Correct \Xinplaceofnumber hook.	1
Explicit error message when calling \Xfootnote outside of \edtext.	1
Fix a bug with line number typesetting direction when using \eledsection and similar commands for RTL texts with Lua \TeX .	1
Fix issues with RTL text in notes when using Lua \TeX .	1
Options fulllines in \Xfootnote.	28
The \newifs are not followed by boolean values set to false, because it is the \TeX default setting.	1
v1.21.0.	
\@edindex@hyperref: Look at the hyperindex option of hyperref before inserting hyperref	315
\l@d@section: \endnotes take five arguments.	237
\ledinnotemark: Add \ledinnotemark.	312
\ledsetnormalparstuffX: \ledsetnormalparstuff is deprecated and becomes \ledsetnormalparstuffX and \Xledsetnormalparstuff.	232
\n@num: \n@num@ref deleted	118
\n@num defined only one time for both Eledmac and Eledpar	118
\newhookcommand@series: \newhookcommand@series can take an optional argument.	265
\newhooktoggle@series: \newhooktoggle@series can take an optional argument.	266
\print@footnoterule: Code refactoring: the spaces after the footnote rules are directly managed in \print@Xfootnoterule and \print@footnoterule	234
\seriesatend: Fix spurious space in \seriesatend	263
\skipnumbering: \skipnumbering defined only one time for both Eledmac and Eledpar.	129
Correct \skipnumbering for stanza.	129
Delete \skipnumbering@reg.	129
General: \AtEveryPstart and \AtEveryPend are now compatible with \autopar	1
\Xafterrule and \afterruleX features no longer create problems of overflowing at the bottom of the page.	1
\chapter inside optional argument of \pstart works when typesetting parallel pages	1
\preXnotes and \prenotesX features no longer create problems of overflowing at the bottom of the page.	1
\seriesatbegin and \seriesatbegin more efficient	262
Add \applabel and related	62
Add \beforenotesX and \Xbeforenotes features for notes set in two and three column.	1
Add \hidenummering	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 1.0.0</code>).	1
Fix a bug with <code>\maxhnotesX</code> when using <code>\foottwocolX</code> or <code>\footthreecolX</code> .	1
Fix a bug with space between columns with notes in two columns (bug added in <code>v1.13.0</code>).	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>	88
v1.22.0.	
<code>\ledinnote</code> : <code>\ledinnote</code> takes a first optional argument, which is the label for hyperlinks.	312
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
v1.22.1.	
<code>\prevpage@num</code> : Correct double symbol when using both <code>\parafootsep</code> and <code>\Xsymlinenum</code> .	190
General: Fix a bug (added on <code>v1.22.0</code>) with <code>\Xinplaceofnumber</code> hook.	1
v1.23.0.	
<code>\@edtext@level</code> : The boolean <code>\if@edtext@</code> becomes the counter <code>\edtext@level</code> .	133
<code>\Serefwithpage</code> : Debug <code>\xendtwolines</code> , <code>\xendmorethantwolines</code> , <code>\xendtwolinesbutnotmore</code> and <code>\xendtwolinesonlyinsamepage</code> when using <code>\apprefwithpage</code> .	293
<code>\lemma</code> : Fix spurious space after <code>\lemma</code> command	139
<code>\newseries@</code> : Prevent spurious spaces when <code>\Afootnote</code> and similar commands are followed by spaces (bug added on <code>1.0.0</code>).	253
<code>\sameword</code> : In order to allow use of <code>\sameword</code> with <code>inputenc</code> , we detokenize its mandatory argument before using it in control sequence names.	143
General: Add <code>\Xboxlinenumalign</code> and <code>\Xendboxlinenumalign</code> .	46
Add <code>\Xboxstartlinenum</code> , <code>\Xendboxstartlinenum</code> , <code>\Xboxendlinenum</code> , <code>\Xendboxendlinenum</code> .	46
Allow use of <code>\sameword</code> with <code>inputenc</code> managing of UTF-8.	1
Compatibility between <code>nofamiliar</code> / <code>nocriticals</code> option and <code>minipage</code> / <code>ledgroup</code> .	1
Error message when using <code>\beginnumbering... \endnumbering</code> without <code>\pstart</code> .	1
Fix a bug with <code>\sameword</code> when the lemma overlaps multiple line.	31
Fix a bug with <code>\sameword</code> when the same lemma is used for multiple notes or for nested <code>\edtexts</code> .	31
Fix a bug with <code>\skipnumbering</code> called immediately after a <code>\pstart</code> .	1
Fix error of <code>\iftrue</code> not closed.	1

Fix spurious space with <code>\skipnumbering</code> (bug added on v1.21.0).	1
New tools to ensure the line-list file uses the right version of commands when upgrading the <code>eledmac</code> version.	1
Optional argument of <code>\sameword</code> can be a comma-separated list of <code>\edtext</code> depth.	31
v1.23.1.	
General: Fix a bug with <code>\lemma</code> command in the right side.	1
v1.23.2.	
General: Compatibility with \TeX 's release 2015.	1
v1.24.0.	
General: We can reinitialize <code>\AtEveryPstart</code> and <code>\AtEveryPend</code> providing to it an empty argument.	1
v1.24.1.	
General: <code>\lemma</code> is disabled when using <code>nocritical</code> option.	1
v1.24.2.	
General: Fix incompatibility between <code>nofamiliar</code> option and <code>memoir</code> package.	1
v1.24.3.	
General: Restore marginal numbers and notes with sectioning command (bug introduced in v1.21.0)	1
v1.24.4.	
General: Fix spurious space with <code>\edindex</code> when using <code>xindy+hyperref</code> option.	1
v1.24.5.	
General: Fix a bug of indent, when a added in 1.1.0, when a <code>\beginnumbering</code> immediately follow a sectioning command.	1
v2.0.0.	
<code>\@iiiminipage</code> : Patch <code>\@iiiminipage</code> instead of redefining it.	307
<code>\@xympar</code> : Patching <code>\@xympar</code> instead of redefining it	297
<code>\endminipage</code> : Patch <code>\endminipage</code> instead of redefining it.	307
<code>\initnumbering@quote</code> : <code>\initnumbering@sectcmd</code> becomes <code>\initnumbering@quote</code>	359
<code>\l@advance@parledgroup@beforenormalnotes</code> : Some conde of <code>\l@dumboxmpfoot</code> moved to <code>\l@advance@parledegroupp@beforenormalnotes</code>	307
<code>\newseries@</code> : One endnotes file by series.	259
General: <code>\@makecol</code> , <code>\@reinserts</code> and <code>\@doclearpage</code> are patched instead of begin redefined	276
<code>\doxtrafeeti</code> becomes <code>\Xdo@feet</code> ; <code>\doxtrafeetii</code> becomes <code>\do@Xfeet</code> ; <code>\@opxtrafeeti</code> becomes <code>\@opfeetX</code> ; <code>\doreinxtrafeetii</code> becomes <code>\X@doreinfeet</code> ; <code>\doreinxtrafeeti</code> becomes <code>\@doreinfeetX</code>	276
Add <code>\Xendinplaceofnumber</code> hook.	1
Add <code>\Xendnonumber</code> hook.	1
Add <code>nonum</code> option for endnotes.	1
Fix a bug when printing only one series of endnotes, but wanted to keep endnotes for other series.	1
In order to have a more consistent name's convention, many names has been changed.	1
Many \TeX 's output macros are now patched and not override.	1
Package's name becomes <code>reledmac</code>	1
Patch <code>\@footnotemark</code> instead of redefine it	214
Suppress indexing command specific to <code>memoir</code>	313
v2.0.1.	
General: Fix a bug in <code>eledmac-compat</code> option	1
Fix incompatibility between optional argument of <code>\pstart</code> and <code>\numberpstarttrue</code>	1
v2.1.0.	
General: Fix a bug with <code>\advanceline</code> at the beginning of a <code>\pstart</code>	1
Fix a bug with <code>\chapter</code> in optional argument of <code>\pstart</code> in parallel typesetting with <code>scrbook</code>	1

Fix a bug with <code>\eledchapter</code> in parallel typesetting with <code>scrbook</code> .	1
Fix a bug with <code>\setline</code> at the beginning of a <code>\pstart</code> .	1
Fix spacing bug with <code>\Xhooknote</code> and <code>\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.	
<code>\ledsetnormalparstuffX</code> : Replaced <code>\noindent</code> with <code>\parindent</code> set to 0pt.	232
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 \LaTeX release of october 2015	1
Fix a bug which inserted double space before paragraphed familiar notes.	1
Fix a bug with <code>\edindex</code> when using not-Latin characters without UTF-8 engines	1
v2.2.0.	
General: Fix a bug with combination of <code>\onehalfspacing</code> and two columns and three columns notes typeset.	1
Fix a bug with some setting command and optimization option.	1
Fix spurious space with paragraphed critical notes when using \LaTeX .	1
Increase line list version number to ensure compatibility with new options of <code>reledpar</code> package.	1
New setting tools for endnotes: <code>\Xendnumberonlyfirstinline</code> , <code>\Xendnumberonlyfirstintwolines</code> , <code>\Xendsymmlinenumber</code> , <code>\Xendbeforenumber</code> , <code>\Xendafternumber</code> , <code>\Xendbeforesymmlinenumber</code> , <code>\Xendaftersymmlinenumber</code> , <code>\Xendboxsymmlinenumber</code> , <code>\Xendhangindent</code> , <code>\Xendbhooklinenumber</code> , <code>\Xendahooklinenumber</code> , <code>\Xendbhookinplaceofnumber</code> , <code>\Xendahookinplaceofnumber</code> .	1
v2.2.1.	
General: Compatibility with \LaTeX 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>bidiv17.9</code> and following.	1
Warning message when using some setting commands inside rightside environment (deprecated behavior)	1
v2.3.1.	
General: Fix spurious space when using optional argument of <code>\stanza</code> (introduced in v2.3.0).	1
v2.4.0.	
<code>\footnoteoptions@</code> : First argument of <code>\footnoteoption@</code> is now mandatory, not optional.	176
General: <code>\Xhooknote</code> and <code>\bhooknoteX</code> work with notes in columns.	1
Fix a bug of <code>\parindentX</code> and <code>\Xparindent</code> with two columns and three columns notes.	1
Fix a bug with <code>\sameword</code> in right side.	1
Fix spurious space in two columns and three columns notes.	1
Fix spurious space when using optional argument of <code>stanza</code> (introduced in v2.3.0).	1
New hooks: <code>\Xlinrangeseparator</code> and <code>\Xendlinrangeseparator</code> .	43

Option linerangesep for critical footnotes and endnotes.	43
v2.4.1.	
General: Fix a bug with <code>\appref</code> and <code>\apprefwithpage</code> (introduced in v2.4.0).	1
Fix a bug with tabular environments when using <code>babel</code> or <code>polyglossia</code> languages	
that override \LaTeX <code>\roman</code> command, like Greek language.	1
Fix a bug with tabular environments when using <code>babel</code> or <code>polyglossia</code> languages	
that override \LaTeX <code>\roman</code> command, like Greek.	1
v2.5.0.	
<code>\SErefwithpage</code> : Debug <code>\setapprefprefixsingle</code>	293
<code>\edlabel</code> : Fix a bug when calling <code>\edlabel</code> in a footnotes of the right side	281
<code>\l@d@section</code> : <code>\endnotes</code> take six arguments.	237
<code>\printlines@count</code> : <code>\printlines</code> takes an eighth argument: the line flag	209
<code>\xlineref</code> : <code>\xlineref</code> does not include anymore the side flag. Use <code>\xflagref</code> to get	
it. Not that <code>\edlineref</code> still contains the flag.	287
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>linerangesep</code> optional argument.	1
<code>\edlabel</code> works now in <code>\Xfootnote</code>	1
<code>\lemma</code> can be used even when the <code>nocritical</code> is enabled.	1
Compatibility with new hook and tools of <code>reledpar</code> 2.6.0.	1
Fix spurious vertical space in <code>astanza</code> environment (<code>reledpar</code>)	1
Log now states “There were undefined references” when using wrong references in	
<code>\edlineref</code> or <code>edpageref</code>	1
New hooks to customize page and line number appearance in endnotes.	1
New hooks: <code>\Xhookgroup</code> and <code>\hookgroupX</code>	1
New tools to easily make cross-reference to a passage defined by a start and an end line	
61	
v2.6.0.	
General: Adds compatibility with <code>innnote</code> and <code>notenumber</code> options of <code>indextools</code>	
package.	1
Fix a bug with footnote counter in <code>ledgroup</code> (added in v2.5.0).	1
Fix bug, introduced in v2.5.0, with footnote numbering in parallel typesetting when	
using <code>perpage</code> package.	1
v2.7.0.	
<code>\@k</code> : <code>\rigidbalance</code> is split in <code>\Xrigidbalance</code> and <code>\rigidbalanceX</code>	190
<code>\l@d@section</code> : <code>\endnotes</code> take seven arguments.	237
General: Add dash as default page range separator for <code>\SEonlypage</code>	1
Debug <code>\SErefonlypage</code> when referring to only one page.	1
Delete parenthesis after <code>\SErefonlypage</code>	1
Fix (again) bugs with footnote numbering in parallel typesetting while using	
<code>ledgroup</code> environments (bug added in v2.5.0).	1
Fix a bug with <code>\SErefwithpage</code>	1
Fix bugs in compatibility with <code>innnote</code> and <code>notenumber</code> options of <code>indextools</code>	
package, when indexing outside of a <code>ledgroup</code>	1
New commands to make glossaries connected to page and linenumber with the	
<code>glossaries</code> package	1
New hooks: <code>\Xhsize</code> and <code>\hsizeX</code>	53
New hooks: <code>\Xlemmafont</code> and <code>\Xendlemmafont</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
v2.7.1.	
General: Debug <code>\Xhookgroup</code> hooks executed on columnar footnotes (moved to a larger group, to take effect).	1
v2.7.2.	
General: <code>\Xhsize</code> and <code>\hsizeX</code> become <code>\Xwidth</code> and <code>\widthX</code>	53
Fix problem of hyphenation when using <code>hyperref</code> package (added in v2.7.0).	1
v2.8.0.	
<code>\l@d@section</code> : <code>\Xendhangindent</code> and <code>\Xendafternote</code> can take values which are relative to the font size of the endnote.	237
General: <code>reledmac</code> cross-referencing can take advantage of <code>xr</code> package.	1
More <code>\edgls...</code> commands.	1
No indentation for paragraphed notes in <code>ledgroup</code> . Can be changed with <code>\Xparindent</code> and <code>\parindentX</code> .	1
v2.8.1.	
General: Warnings for undefined labels are really parsable by <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...</code> <code>\resumenumbering</code> .	1
Fix a bug with three- and two-column footnote setting (added in v2.4.0).	1
Fix spurious space inside three-column familiar footnote.	1
Write correct metadata in numbered files when using <code>\pausenumbering...</code> <code>\resumenumbering</code> .	1
v2.9.1.	
General: Fix a bug when notes start with <code>plus</code> or <code>minus</code> .	1
v2.9.2.	
General: Fix a bug with <code>hyperref</code> package when a lemma starts with <code>plus</code> or “minus” (bug introduced in v. 2.7.0).	1
v2.9.3.	
General: Fix a bug with line number position and reset added by v. 2.9.0	1
v2.10.0.	
<code>\print@lemma</code> : Code refactoring between <code>\parafootfmt</code> , <code>\twocolfootfmt</code> , <code>\threecolfootfmt</code> and <code>\normalfootfmt</code> .	199
General: Add <code>\AtEveryStanza</code> and <code>\AtEveryStopStanza</code> .	1
Fix a bug in <code>\ledlsnotefontsetup</code> and <code>\ledrsnotefontsetup</code> which could not handle <code>\color</code> command properly.	1
More specific error messages.	1
New hooks: <code>\Xwrapcontent</code> , <code>\Xendwrapcontent</code> and <code>\wrapcontentX</code> .	49
New hooks: <code>\Xwraplemma</code> and <code>\Xendwraplemma</code>	48

v2.10.1.	
General: Add <code>nopenalties</code> 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.	
<code>\do@actions@fixedcode</code> : Add action 1010	161
General: Add new tools to produce an apparatus of manuscripts	1
Fix a bug in <code>\Xparafootsep</code> in parallel typesetting	1
Make <code>\parafootsepX</code> work	1
Prevent <code>\Xtxtbeforenotes</code> hook from causing notes to go beyond the bottom margin	1
v2.12.0.	
General: <code>\preXnotes</code> becomes <code>\Xprenotes</code> (naming convention)	1
Add <code>\hidenumberingonleftpage</code> and <code>\hidenumberingonrightpage</code>	1
Add <code>\toendnotes</code> and related.	1
Add <code>auxdir</code> option.	1
Fix a bug in critical and familiar footnotes when using uppercase letters with accent	
mark	1
Fix a bug when using <code>\chapter</code> in optional argument of <code>\pstart</code> in parallel	
typesetting in combination with the <code>noeledsec</code> option.	1
Fix a bug with <code>\ledinnernote</code> and <code>\ledouternote</code> in parallel typesetting	1
Fix a bug with familiar footnote number in optional argument of <code>\pstart</code> or <code>\pend</code>	
in parallel typesetting	1
Fix spurious vertical space in <code>\chapter</code> when used as optional argument of <code>\pstart</code>	
in parallel typesetting.	1
Make endnote compatible with <code>\sameword</code> mechanism	1
More accurate message to control the position of <code>\Xfootnote</code> and <code>\applabel</code> in the	
\LaTeX code	1
v2.13.0.	
General: Version 2.13.0 never existed.	1
v2.13.1.	
General: In critical footnotes, the right side flag is printed only if requested explicitly	
with <code>\Xlineflag</code> (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 <code>\Xendafterpagenumbe</code> affecting <code>\Serefwithpage</code>	1
v2.14.0.	
General: Hyperref with the line number inside critical footnotes is correct when using	
<code>\xxref</code>	1
Some internal changes for new features of <code>reledpar</code>	1
v2.14.1.	
General: Fix a bug when using <code>\footnoteX</code> in the first argument of <code>\edtext</code>	1
v2.14.1a.	
General: Fix problematic typos in the handbook.	1
v2.15.0.	
General: Add “byline” arrangement.	1
Fix <code>\Xtxtbeforenotes</code> in <code>ledgroup</code>	1
v2.15.1.	
General: Fix <code>\edindex</code> in tabular environments.	1
v2.15.2.	
General: Fix a bug with <code>fancyhdr</code> package 3.8 and later.	1
v2.15.3.	
General: Fix a bug with <code>\section</code> in optional argument of <code>\pstart</code> and empty line	
before <code>\pend</code> (bug added in v2.8.2).	1

Simplification of the sectioning command code.	1
v2.16.0.	
General: <code>\Xdo@feet</code> becomes <code>\do@Xfeet</code>	276
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.	276
New options for <code>\fnpos</code> and <code>\mpfnpos</code> to set a customized order for familiar and critical footnotes.	1
When <code>\edindex</code> is called outside of a <code>\beginnumbering... \endnumbering</code> structure, it is automatically switched to <code>\index</code> , with a warning message.	1
When indexing texts in familiar footnotes with <code>\edtext</code> , refer to the line number where the footnote is called.	1
When indexing texts in sidenotes with <code>\edtext</code> , refer to the line number where the sidenote is called.	1
v2.16.1.	
General: Fix a bug with redefinition of the style of the footnote number (bug added in v2.12.0)	1
v2.16.2.	
General: Error message if <code>footmisc</code> is loaded after <code>reledmac</code>	1
Fix a bug introduced by v2.16.1 when using non-expandable control sequence, like <code>\normalfont</code> , in the footnote number style.	1
v2.16.3.	
General: Fix a bug with <code>\Seref</code> (bug added in v2.7.0).	1
v2.16.4.	
General: Fix a bug with vertical space before sectioning command in optional argument of <code>\pstart</code> (bug added in v2.15.3).	1
v2.16.5.	
General: Fix potential spurious spaces in endnotes.	1
v2.16.6.	
General: Fix a bug with the line number style in <code>\doennotes</code> when referring to right side line in parallel typesetting.	1
Take into account <code>\linenumberstyle</code> when using <code>\edlineref</code>	1
v2.16.7.	
General: Fix a bug with <code>\msdata</code> when using multiple <code>\beginnumbering... \endnumbering</code>	1
Fix a bug with <code>\numberpstarttrue</code> when using multiple <code>\beginnumbering... \endnumbering</code>	1
v2.16.8.	
General: Fix a bug with <code>\edindex</code> in footnotes in parallel typesetting.	1
v2.17.0.	
General: Add <code>\edglsadd</code> command.	1
Add <code>\setmsdataposition</code> setting.	1

v2.17.1.	
General: Fix spurious space in paragraphed footnotes when using Lua \TeX without using Right-To-Left text.	1
v2.17.2.	
General: Change log message when numbered files still don't exist, in order to improve compatibility with <i>latexmk</i>	1
v2.17.3.	
General: Fix a bug with <code>\doendnotesbysection</code> and <code>\doendnotes</code>	1
v2.17.4.	
General: Fix a bug with <code>\setSerefonlypageprefixsingle</code> and <code>\setSerefonlypageprefixmore</code>	1
v2.17.5.	
General: Fix a bug with <code>\pstartref</code> when referring to the left side in parallel typesetting.	1
v2.18.0.	
General: Fix a bug when using a <code>\edtext</code> in two lines or more in right-to-left typesetting with \TeX	1
Fix a bug when using both <code>\Xnumberonlyfirstintwolines</code> or <code>\Xnumberonlyfirstinline</code> and <code>\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.	
\next@line@list@stuff: Add \next@line@list@stuff internal hook.	123
General: Add \swnoexpands macro to avoid problems with not fully expandable macro	
inside \sameword.	1
Fix a bug with \edlabel at the beginning of a \pstart.	1
Fix spurious space with \labelpstarttrue	1
When a \setlinenum is used, it is stronger than the \lineation{page} setting.	1
v2.24.1.	
General: Compatibility with reledpar 2.20.2	1
v2.24.2.	
General: Fix bug between tabular environments and endnotes	1
v2.25.0.	
General: Add \Xendtxtbeforenotes hook	1
v2.26.0.	
General: Add a mechanism for annotation of stanza line numbering	1
Fix bug in \xpstartref	1
v2.26.1.	
General: Fix bug with \linenumannotation	1
v2.26.10.	
General: Add error message when insertion a \pstart or \pend inside \edtext	
arguments	1
Really fix bug with page number in footnote after \mainmatter (bug introduced in	
v2.26.7)	1
v2.26.11.	
General: Allow to use \edtext inside \sameword	1
v2.26.12.	
General: Precise minimal version of xparse package required.	1
v2.26.2.	
General: Fix bug with \linenumannotation in parallel typesetting	1
Fix bug with \Xgroupbyline and \Xparafootsep.	1
Fix bug with \Xgroupbyline and optional argument of critical footnotes.	1
Fix bug with \Xgroupbyline 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 \numberpstarttrue and alternating between manual	
\pstart and automatic \pstart produced by \autopar	1
v2.26.5.	
General: Check that users have a enough recent version of xparse	1
Fix bug when using \eledsection with the continuousnumberingwithcolumns	
of reledpar	1
Fix bug with continuousnumberingwithcolumns	1
Fix bug with using \eledsection and related when alternate normal typesetting and	
parallel typesetting	1
v2.26.6.	
General: Fix bug with \lineation by page and nexted \edtext between two pages	
(added in v. 2.24.0)	1
v2.26.7.	
General: Fix page number in footnote and endnotes when using	
sameparallelpagenum option of reledpar 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 ^A T _E X2018/04/01 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>\hidenumering</code>	1
v2.28.0.	
General: Familiar footnotes can take an optional argument to manually define the footnote mark.	1
v2.28.1.	
General: Fix bug with the alignment of the columns of two and three columns critical footnotes.	1
v2.29.0.	
General: Add <code>noresetlinenumannotation</code>	1
Fix bug with <code>\linenumannotation</code> and <code>nofamiliar</code> option.	1
Fix little bug with <code>\msdata</code>	1
Really reset line number annotation at each line	1
v2.30.0.	
General: Add <code>\Xnoidenticallinenumannotation</code> and <code>\Xendnoidenticallinenumannotation</code> hooks	1
Add separator between line number annotations in margin. Add <code>\setlinenumannotationsep</code>	1
v2.31.0.	
General: Add <code>noprefix</code> and <code>prefixmore</code> to <code>\appref</code> and <code>\SEref</code>	1
v2.31.1.	
General: Fix bug in vertical spacing with <code>\Xgroupbyline</code>	1
Fix bug with <code>\Xgroupbyline</code> , which could produce spurious blank lines.	1
Fix bug with <code>\Xhangindent</code> and <code>\hangindentX</code> for right-to-left texts.	1
Fix bug with indentation with <code>\Xgroupbyline</code>	1
Fix bug with multiple footnote layer and <code>\Xgroupbyline</code>	1
Fix incompatibility between <code>\Xgroupbyline</code> and <code>hyperref</code> package.	1
Fix incompatibility between <code>\Xgroupbyline</code> and optional argument of critical footnotes.	1
v2.31.2.	
General: Fix bug with familiar footnote containing non-default font (introduced with v. 2.20.0).	1
v2.31.3.	
General: Fix bug with paragraphed footnote in RTL context.	1
v2.32.0.	
General: Add <code>\Xlinenumannotationonlyfirst</code> , <code>\Xlinenumannotationonlyfirstintwo</code> , <code>\Xsymlinenumannotation</code> , <code>\Xendlinenumannotationonlyfirst</code> , <code>\Xendlinenumannotationonlyfirstintwo</code> and <code>\Xendsymlinenumannotation</code>	1
v2.32.1.	
General: Do not print <code>\Xendsep</code> if we print <code>\Xendsymlinenum</code>	1
When using <code>\parindentX</code> or <code>\Xparindent</code> and stanza, use the normal indentation in the footnote, not the verse indentation.	1
v2.32.2.	
General: Fix spurious vertical space in familiar footnotes with Lua ^L _A T _E 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 twocol arrangement.	1
Fix bug with three and two columns footnotes when using <code>bid</code> .	1
v2.32.5.	
General: Fix bug with <code>\edindex</code> inside <code>\eledsection</code> and related.	1
Fix bug with macro inside <code>\sameword</code> with no- \XeTeX engines.	1
Fix bug with page number in index with <code>\edindex</code> when indexing a critical notes.	1
v2.32.6.	
General: Add <code>hyperref</code> link when inserting glossary entries.	1
Fix bug with glossary entries inside the first argument of <code>\edtext</code> .	1
v2.32.7.	
General: Fix bug with <code>\lineation{pstart}</code> and <code>\edtext</code> in the very first line of a <code>pstart</code> .	1
Fix bug with <code>\parafootsepX</code> .	1
Fix spurious spaces at the very first line of <code>edtabular</code> environments.	1
v2.32.8.	
General: Add subline number in manuscript apparatus.	1
v2.33.0.	
<code>\ledinnotemark</code> : <code>\ledinnotemark</code> redefinition is deprecated, use <code>\Xinnotemark</code> and <code>\innotemarkX</code> instead.	312
<code>\newhookcommand@series</code> : <code>\newhookcommand@series</code> can take an second optional argument (third argument).	265
General: Add <code>\Xinnotemark</code> and <code>\innotemarkX</code> hooks.	1
Add antilabe feature.	1
Add <code>startstanzaindentsrepetition</code> feature.	1
v2.33.1.	
General: Fix a spacing bug when the argument of <code>\antilabe</code> contains more than one word.	1
v2.34.0.	
General: Add <code>\Xpstartonlyfirst</code> and <code>\Xstanzaonlyfirst</code> .	1
Add <code>\Xpstartseparator</code> .	1
v2.34.1.	
General: Fix bug with page number when using <code>stanazindentsrepetition</code> , bug introduced in v2.33.0.	1
v2.34.2.	
General: Fix bug with tabular environment, in some specific use case, added in v2.32.7.	1
v2.34.3.	
General: Fix bug with <code>\parafootsepX</code> when the footnotes are called outside of a numbered section.	1
Fix incompatibility between lineation by <code>pstart</code> and (a)stanza.	1
Make commands like <code>\hidenumbering</code> working at the very beginning of a verse when the first argument of <code>\setstanzaindents</code> is equal to 0.	1
v2.34.4.	
General: Fix bugs with <code>\parafootsepX</code> outside of numbered section.	1
Fix some bugs with lineation by page, when a numbered section starts at the very beginning of page.	1
v2.34.5.	
General: Fix incompatibility between <code>\apprefwithpage</code> / <code>\Serefwithpage</code> and line number annotation mechanism.	1

v2.34.6.	
General: Fix incompatibility with calc package (added with v2.34.4).	1
v2.34.7.	
General: Fix bug with \parafootsepX in some circumstances (added with v2.34.4).	1
v2.35.0.	
General: Add \linenumannotationothersidettrue.	1
Add new hooks: \Xnolinenumber, \Xendnolinenumber,	
\Xnolinenumberifannotation, \Xendnolinenumberifannotation.	1
v2.35.1.	
General: Fix a bug when printing ending page number in endnotes and using the	
hyperref package.	1
v2.36.0.	
General: Add \Xnonote and \nonoteX.	1
Fix bug with sublineation on right side of parallel typesetting.	1
v2.36.1.	
General: Fix bug with crossreferencing in parallel typesetting (added in v2.34.4)	1
v2.37.0.	
General: Fix bug with \Xnonote	1
Fix incompatibility between \eledsection (and related) and hyperref.	
Incompatibility was introduced by some changes in hyperref package.	1
Improve syntax check.	1
v2.37.1.	
General: Fix bug with \edgls and related in \edtext argument. Require	
glossaries-extra is glossaries is loaded.	1
v2.37.2.	
General: Fix bug with \Xgroupbyline and two or three columns notes.	1
Fix bug with \Xstanzaonlyfirst.	1
Fix bug with parapparatus option and nested \edtext.	1
v2.38.0.	
General: Add \Xbeforepagenumber hook.	1
Add \Xnopagenumberifcurrent hook.	1
Add hyperlinks between footnote and lemma when using abbreviated line range.	1
Fix bug with \lemma and right-to-left typesetting with Xe _{La} TeX.	1
Fix indent bugs with stanza and right-to-left typesetting.	1
v2.38.1.	
General: Fix incompatibility between reledpar's option	
continuousnumberingwithcolumns and lineation by page.	1
v2.38.2.	
General: Fix bug when using \applabel in conjunction with hyperref package.	1
Fix some trouble in indentation in stanza, when switching to some language with	
polyglossia.	1
v2.39.0.	
General: Add \Xnotboxingsubline and \Xendnotboxingsubline hooks.	1
v2.39.1.	
General: Fix bug with index in familiar and critical footnotes.	1