

reledmac

Typeset scholarly editions with L^AT_EX*

Maïeul Rouquette[†]

based on the original ledmac by

Peter Wilson

Herries Press

which was based on the original edmac, tabmac and edstanza by

John Lavagnino, Dominik Wujastyk, Herbert Breger and Wayne Sullivan.

Abstract

The `reledmac` provides many tools in order to typeset scholarly editions. It is based on the `eledmac` package, which was based on the `ledmac` package, which was based on the `edmac` T_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	12
1.1 Aim of the package	12
1.2 History	13
1.2.1 edmac	13
1.2.2 ledmac	14
1.2.3 eledmac	15

*This file (`reledmac.dtx`) has version number v2.36.1, last revised 2021/02/14.

[†]maieul at maieul dot net

1.2.4 <code>reledmac</code>	15
1.3 Bibliography	15
1.4 Acknowledgements	15
2 How the package works — the problem of the number of \LaTeX runs	15
3 Compatibility warning	16
4 Options	16
4.1 Specific features	16
4.2 Optimizing package performance	17
5 Text lines and paragraphs numbering	18
5.1 Text lines numbering	18
5.2 Paragraphs	18
5.2.1 Basics	18
5.2.2 Automatically producing <code>\pstart ... \pend</code>	19
5.2.3 Content before specific <code>\pstart</code> and after specific <code>\pend</code>	19
5.2.4 Content before every <code>\pstart</code> and after every <code>\pend</code>	20
5.2.5 Numbering paragraphs (<code>\pstart</code>)	20
5.2.6 Languages written in Right to Left	21
5.2.7 Memory limits	21
5.3 Lineation commands	21
5.3.1 Disabling lineation	21
5.3.2 Setting lineation start and step	22
5.3.3 Setting lineation reset	22
5.3.4 Setting line number margin	23
5.3.5 Other settings	23
5.4 Changing the line numbers	23
5.4.1 Sublineation	23
5.4.2 Locking lineation	24
5.4.3 Setting and changing line number	24
5.4.4 Line number style	24
5.4.5 Skipping and hiding number	25
5.5 Adding annotations to line numbers	25
5.5.1 Resetting annotation	26
5.5.2 Modifying annotation associated with a specific note	27
5.5.3 Changing the position of the annotation	27
5.5.4 Changing the macro in which annotations are wrapped	27
5.5.5 Not printing the second identical annotations	28
5.5.6 Separator between annotations	28
5.5.7 Annotations in the apparatus	28
5.6 Executing code at each line	28
5.7 Executing code at a specific line	29

6 Apparatus commands	29
6.1 Terminology	29
6.2 Critical notes	30
6.2.1 The lemma	30
6.2.2 Footnotes	30
6.2.3 Endnotes	31
6.2.4 Paragraph in critical apparatus	32
6.2.5 Change lemma and line number	33
6.2.6 Changing the names of commands for critical apparatus	33
6.3 Disambiguation of identical words in the apparatus	34
6.3.1 Basic use	34
6.3.2 Case setting	34
6.3.3 Notes about input encoding with UTF-8 processor	34
6.3.4 Notes about right-to-left texts	35
6.3.5 Use with <code>\lemma</code> command	35
6.3.6 Sameword for a group of words	37
6.3.7 Customizing	38
6.3.8 Problems with some macros	38
6.3.9 Automatic sameword annotation	39
6.4 Apparatus of manuscripts	39
6.4.1 Marking sections of text	39
6.4.2 Layout of the apparatus of manuscripts	40
6.4.3 Settings	40
6.5 Familiar notes	41
6.5.1 Basic use	41
6.5.2 Customizing mark	41
6.5.3 Separator for multiple footnotes	41
6.6 Printing the footnote mark without printing the footnote text	42
6.7 Changing series	42
6.7.1 Create a new series	42
6.7.2 Delete series	42
6.7.3 Series order	42
6.8 Position of critical and familiar footnotes	42
7 Apparatus customization	43
7.1 Introduction	43
7.2 Notes arrangement in a series	44
7.3 Disabling footnote	45
7.4 Control number printing	45
7.4.1 Print line number only at first time	45
7.4.2 Print page number only at first time	45
7.4.3 Arbitrary text before line number	46
7.4.4 Separator for line range	46
7.4.5 Abbreviate line range	46
7.4.6 Disabling line number printing	47
7.4.7 Printing pstart number	47

7.4.8 Printing stanza number	48
7.4.9 Disabling line number printing	48
7.4.10 Options for line number annotations	48
7.4.11 Separator between line and subline numbers	49
7.4.12 Separator between page and line numbers	49
7.4.13 Space around number	49
7.4.14 Space around line symbol	49
7.4.15 Space in place of number	49
7.4.16 Boxing line number and line symbol	50
7.5 Separator between the lemma and the note	51
7.5.1 For footnotes	51
7.5.2 For endnotes	51
7.6 Font style	51
7.6.1 For line number	51
7.6.2 For the lemma	52
7.6.3 For all notes	52
7.7 Wrapping notes	52
7.7.1 Wrapping lemmas	52
7.7.2 Wrapping contents	53
7.8 Indent of notes content	53
7.9 Arbitrary code	53
7.9.1 Arbitrary code around line number	53
7.9.2 Arbitrary code at the beginning of notes	53
7.9.3 Arbitrary code before inserting note	54
7.10 Options for footnotes in columns	54
7.10.1 Alignment	54
7.10.2 Size of the columns	54
7.11 Options for paragraphed footnotes and notes grouped by line	55
7.11.1 Mark separation of notes	55
7.11.2 Ragged text	55
7.12 Options for block of notes	55
7.12.1 Grouping notes by line	55
7.12.2 Text before notes	55
7.12.3 Code before notes	56
7.12.4 Spacing	56
7.12.5 Rule	56
7.12.6 Maximum height	56
7.12.7 Width	57
7.13 Footnotes and the <code>reledpar</code> columns	57
7.14 Endnotes in one paragraph	57

9 Verse	58
9.1 Basic	58
9.2 Define stanza indents	58
9.3 Repeating stanza indents	59
9.4 Notes about the scope of the indent settings	60
9.5 Manual stanza indent	60
9.6 Stanza breaking	60
9.7 Hanging symbol	61
9.8 Long verse and page break	61
9.9 Content before/after verses	61
9.10 Numbering stanza	62
9.11 Antilabe	62
9.12 Various tools	63
9.13 Notes on empty lines	63
10 Grouping	63
11 Cross referencing	64
11.1 Basic use	64
11.2 Cross-referencing to a critical note	65
11.3 Cross-referencing which return a number in any case	65
11.3.1 Cross-referencing in order to define line number of a critical note	65
11.4 Not automatic cross-referencing	66
11.5 Normal \LaTeX cross-referencing	66
11.6 References to start and end lines	66
11.6.1 Reference to main text lines	66
11.6.2 References to lines that are commented on in the apparatus	67
11.6.3 Settings	67
11.6.4 Combining multiple references	69
11.7 Compatibility with xr package	69
12 Sidenotes	69
12.1 Basics	69
12.2 Setting	70
12.2.1 Width	70
12.2.2 Vertical position	70
12.2.3 Distance to the main text	70
12.2.4 Font	70
12.2.5 Separator between notes	70
13 Indexing	71
13.1 Basics	71
13.2 Use with imakeidx or indextools	71
13.3 Referring to critical notes	71
13.4 Separator between page and line numbers	72
13.5 Using xindy	72

13.6 Advanced setting	73
14 Glossary	73
14.1 Preamble setting	73
14.2 Commands	73
15 Tabular material	74
16 Sectioning commands	77
16.1 Sectioning commands without line numbers or critical notes	77
16.2 Sectioning commands with line numbering and critical notes	77
16.3 Optimization	78
17 Quotation environments	78
18 Page breaks	78
18.1 Control page breaking	78
18.2 Prevent page break in a long verses	79
19 Miscellaneous	79
19.1 Known and suspected limitations	80
19.1.1 Non-standard geometry	80
19.1.2 floatrow package compatibility	80
19.1.3 “No room for a new”	80
19.1.4 Marginal notes	80
19.1.5 Paragraph shape	81
19.1.6 Paragraphed footnotes	81
19.1.7 Use with other packages	81
19.2 Parallel typesetting	82
I Implementation overview	83
II Preliminaries	83
II.1 Links with original edmac	83
II.2 Package declaration	83
II.3 Package options	84
II.3.1 Options of reledpar	84
II.3.2 Options of reledmac	84
II.4 Loading packages	86
II.5 Compatibility with LuaTeX	87
II.6 Boolean flags	87
II.7 Messages	88
II.8 Gobbling	95
II.9 Wrapping content	95
II.10 Miscellaneous commands	96
II.11 Prepare reledpar	96

II.12 Booleans provided by other optional packages which are required in any case	97
III Sectioning commands	98
IV List macros	103
V Line counting	104
V.1 Choosing the system of lineation	104
V.2 Line number margin	106
V.3 Line number initialization and increment	107
V.4 Line number locking	108
V.5 Line number style	109
V.6 Line number printing	110
V.7 Line number counters and lists	111
V.8 Line number locking counter	113
V.9 Line number associated to lemma	113
V.10 Reading the line-list file	116
V.11 Commands within the line-list file	119
V.12 Writing to the line-list file	132
VI Marking text for notes	140
VI.1 <code>\edtext</code> itself	141
VI.2 Substitute lemma	149
VI.3 Substitute line numbers	150
VI.4 Lemma disambiguation	151
VII Paragraph decomposition and reassembly	158
VII.1 Boxes, counters, <code>\pstart</code> and <code>\pend</code>	158
VII.2 Processing one line	165
VII.2.1 General process	165
VII.2.2 Process for “normal” line	165
VII.2.3 Process for line containing <code>\eledsection</code> command	167
VII.2.4 Hooks	168
VII.2.5 Sidenotes and marginal line number initialization	169
VIII Line and page number computation	170
VIII.1 Continuous line numbering between parallel typesetting and normal typesetting	174
IX Line number annotation	174
X Line number printing	177
XI Pstart number printing inside	182

XII Restoring footnotes and penalties	183
XII.1 Add insertions to the vertical list	183
XII.2 Penalties	185
XII.3 Printing leftover notes	186
XII.4 Text before notes	187
XIII Critical footnotes	189
XIII.1 Fonts	189
XIII.2 Individual note options	189
XIII.3 Notes language	190
XIII.4 General survey of the way we manage notes	191
XIII.5 General setup	191
XIII.6 Footnotes arrangement	192
XIII.6.1 User level macro	192
XIII.6.2 Normal footnote	192
XIII.6.3 Paragraphed footnotes	198
XIII.6.4 Columnar footnotes	205
XIII.7 Footnote paragraph indent	212
XIII.8 Footnote hanging indent	212
XIII.9 Critical notes presentation	213
XIII.9.1 Font tools	213
XIII.9.2 Pstart number in footnote	214
XIII.9.3 Lemma printing	214
XIII.9.4 Line number printing	215
XIII.9.5 Footnote grouped by line	228
XIV Familiar footnotes	229
XIV.1 Adjacent footnotes	229
XIV.2 Regular footnotes for numbered texts	231
XIV.3 Footnote formats	233
XIV.4 Footnote arrangement	233
XIV.4.1 User level macro	233
XIV.4.2 Normal footnotes	233
XIV.4.3 Two columns footnotes	239
XIV.4.4 Three columns footnotes	242
XIV.4.5 Paragraphed footnotes	244
XIV.5 Wrapping footnote marks in hyperlink	248
XV Code common to both critical and familiar footnote in normal arrangement	249
XVI Footnotes' width for two columns	250
XVII Footnotes' order	251
XVIII Footnotes' rule	252

XIX Specific skip for first series of footnotes	252
XIX.1 Overview	252
XIX.2 User level command	253
XIX.3 Internal commands	253
XX Endnotes	254
XX.1 Internal commands	254
XX.2 User level commands	259
XX.2.1 Inserting contents to endnotes	259
XX.2.2 Printing endnotes	259
XXI Generate series of notes	269
XXI.1 Test if series is still existing	269
XXI.2 Init specific to <code>reledpar</code>	269
XXI.3 For critical footnotes	270
XXI.3.1 Options	270
XXI.3.2 Create inserts, needed to add notes in foot	272
XXI.3.3 Create commands for critical apparatus, <code>\Afootnote</code> , <code>\Bfootnote</code> etc.	272
XXI.3.4 Set standard display	275
XXI.4 For familiar footnotes	275
XXI.4.1 Options	275
XXI.4.2 Create tools for familiar footnotes (<code>\footnoteX</code>)	276
XXI.5 The endnotes	278
XXI.5.1 The auxiliary file	278
XXI.5.2 The main macro	279
XXI.5.3 Tools	279
XXI.5.4 Internal commands	280
XXI.5.5 The options	280
XXI.6 Init standards series (A,B,C,D,E)	282
XXII Setting series display	282
XXII.1 Change series order	282
XXII.2 Test series order	283
XXII.2.1 Get the first series	283
XXII.3 Series setting	283
XXII.3.1 General way of working	283
XXII.3.2 Tools to set options	283
XXII.3.3 Tools to generate options commands	285
XXII.3.4 Options for critical notes	287
XXII.3.5 Options for familiar notes	289
XXII.3.6 Options for endnotes	289
XXII.4 Hooks for a particular footnote	291
XXII.5 Alias	293

XXIII Output routine	293
XXIII.1 Extra footnotes output	294
XXIII.2 Patching standard output's commands	297
XXIV Page numbering in parallel typesetting	299
XXV Cross referencing	302
XXV.1 Compatibility with xref	318
XXVI Sidenotes	319
XXVII Minipages and such	327
XXVIII Indexing	332
XXVIII.1 Looking on package order	332
XXVIII.2 Auxiliary macros for \edindex	333
XXVIII.3 Code specific to \edindexin critical footnotes	334
XXVIII.4 Analysis of command in indexed text	335
XXVIII.5 Code for the formatted index	335
XXVIII.6 Main code	336
XXVIII.7 Hyperlink	338
XXVIII.8 "innote" and "notenumber" option of indextols package	341
XXIX Glossaries	341
XXX Verse	344
XXX.1 Hanging symbol management	344
XXX.2 Using & character	345
XXX.3 Code category setting	345
XXX.4 Stanza count and indent	346
XXX.5 Numbering stanza	347
XXX.6 Stanza number in note	348
XXX.7 Main work	349
XXX.8 Restore catcode and penalties	352
XXX.9 Dealing with antilabe	352
XXXI Apparatus of Manuscripts	353
XXXI.1 User level macro	353
XXXI.2 Setting macro	355
XXXI.3 Counters and lists	355
XXXI.4 Auxiliary file macros	356
XXXI.5 Action macro	357
XXXI.6 Inserting footnote	362
XXXI.7 Other	362

XXXII Arrays and tables	363
XXXII.1 Preamble: macro as environment	363
XXXII.2 Tabular environments	367
XXXII.2.1 Disabling and restoring commands	367
XXXII.2.2 Counters, boxes and lengths	371
XXXII.2.3 Tabular typesetting	374
XXXII.2.4 Environments	385
XXXIII Quotation's commands	386
XXXIV Section's title commands	387
XXXIV.1 Commands to disable some feature	387
XXXIV.2 General overview	387
XXXIV.3 <code>\beforeeledchapter</code> command	388
XXXIV.4 Auxiliary commands	388
XXXIV.5 Patching standard commands	389
XXXIV.6 Main code of <code>\eledxxx</code> commands	394
XXXIV.7 Macros written in the auxiliary file	397
XXXV Page breaking or no page breaking depending on specific lines	399
XXXVI Long verse: prevents being separated by a page break	401
XXXVII Tools for hyperref package	402
XXXVIII Compatibility with <code>eledmac</code>	402
Appendix A Things to do when changing versions	405
A.1 Migrating from <code>edmac</code> to <code>ledmac</code>	405
A.2 Migration from <code>ledmac</code> to <code>eledmac</code>	406
A.3 Migration to <code>eledmac</code> 1.5.1	407
A.4 Migration to <code>eledmac</code> 1.12.0	407
A.5 Migration to <code>eledmac</code> 17.1	408
A.6 Migration to <code>eledmac</code> 1.21.0	408
A.6.1 <code>\Xledsetnormalparstuff</code> and <code>\ledsetnormalparstuffX</code>	408
A.6.2 Endnotes	408
A.7 Migration to <code>eledmac</code> 1.22.0	408
A.8 Migration to <code>eledmac</code> 1.23.0	408
A.9 Migration from <code>eledmac</code> to <code>reledmac</code>	409
A.9.1 Risk of 'no room for a new'	409
A.9.2 Multiple indices with memoir	409
A.9.3 Deprecated commands and options	409
A.9.4 <code>\renewcommand</code> replaced by <code>command</code>	410
A.9.5 Commands the names of which have been changed	410
A.9.6 Endnotes	412
A.9.7 Z Series	412
A.9.8 Internal commands	412

A.10 Migration to reledmac 2.1.0	412
A.11 Migration to reledmac 2.1.3	412
A.12 Migration to reledmac 2.3.0	412
A.13 Migration to reledmac 2.4.0	413
A.14 Migration to reledmac 2.5.0	413
A.15 Migration to reledmac 2.7.0	413
A.16 Migration to reledmac 2.7.2	413
A.17 Migration to reledmac 2.8.0	413
A.18 Migration to reledmac 2.13.1	413
A.19 Migration to reledmac 2.18.0	414
A.20 Migration to reledmac 2.21.0	414
A.21 Migration to reledmac 2.24.0	414
A.22 Migration to reledmac 2.26.0	414
A.23 Migration to reledmac 2.27.1	414
A.24 Migration to reledmac 2.30.0	414
A.25 Migration to reledmac 2.31.1	415
A.26 Migration to reledmac 2.32.7	415
A.27 Migration to reledmac 2.33.0	415
Appendix B Auxiliary softwares	415
B.1 <i>samewords</i>	415
B.2 critical-keys for <i>Emacs</i>	415
B.3 critical-marks for <i>Emacs</i>	415
B.4 Import from TEI	415
B.5 Import from TEI	415
References	417
Index	417
Change History	470

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.

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).

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

As of 1994, we were pleased to be able to say that `edmac` was being used for the real-life book production of several interesting editions, such as the Latin texts of Euclid’s *Elements*,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 Quinqe ecclesiensi 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 \TeX to LaTeX.

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. Gädke and others at the Leibniz-Archiv, Niedersächsische Landesbibliothek, Hannover. (see <http://www.nlb-hannover.de/Leibniz>)

¹²Being prepared at Poona and Lausanne Universities.

The starting point was `edmac` version 3.16 as documented on 19 July 1994 (available from CTAN). In August 2003 the `tabmac` functions were added; the starting point for these being version 1.0 of October 1996. The `edstanza` (v0.01) functions were added in February 2004. Sidenotes and regular footnotes in numbered text were added in April 2004. This port was called `ledmac` (\LaTeX `edmac`).

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

1.2.3 `eledmac`

Important changes were put in version 1.0, to make `ledmac` more easily extensible (see 7 p. 43). 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. 406.

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. 409.

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 compila-

tions 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. 56.

If you use the \LuaTeX engine, you need \LuaTeX 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. 62);

- eledmac-compat** help to migrate from eledmac to reledmac (see A.9.5 p. 410);
- 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. 78);
- noresetlinenumannotation** does not reset the annotations to line number at each line (see 5.5 p. 25);
- parapparatus** by default, the apparatus cannot contain paragraph breaks; this option enables paragraphing inside the apparatus;
- swcaseinsensitive** make `\sameword` command case insensitive;
- widthliketwocolumns** set the width of the text printed in a single column to be the same as the width of the text printed in two parallel columns with `reledpar`. This is useful when alternating between normal and parallel typesetting;
- xindy** and `xindy+hyperref` select `xindy` as the index processor (13.5 p. 72).

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. 77).
- 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. 63). However, such features use up computer memory, at the expense of other processing needs. So if you do not need this feature, use `noledgroup` option. This should make `reledmac` faster.
- series** `reledmac` defines five levels of notes: A, B, C, D, E. Using all these levels consumes memory space and processing speed. This is why, if your work does not require the entire A–E series, you can narrow down the available number of series. For example, if you only need A and B series, call the package with `series={A,B}` option.

5 Text lines and paragraphs numbering

5.1 Text lines numbering

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

```
\beginnumbering
Text
\endnumbering
```

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

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

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

5.2 Paragraphs

5.2.1 Basics

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

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

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

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

```

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

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

The lines of this paragraph are
not numbered.

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

```

5.2.2 Automatically producing \pstart ... \pend

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

```

\beginnumbering
\autopar

A paragraph of numbered text.

Another paragraph of numbered
text.

\endnumbering

```

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

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

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

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

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

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

This feature is also useful when typesetting verses (see 9 p. 58) or `reledpar` (see 19.2 p. 82).

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

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

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

`\AtEveryPend`

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

`\AtStartEveryPstart`
`\AtEndEveryPend`

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

```
\AtStartEveryPstart{*}
```

Instead of manually doing

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

5.2.5 Numbering paragraphs (`\pstart`)

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

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

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

On each `\beginnumbering` the numbering restarts.

`\sidepstartnumtrue`

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

`\labelpstarttrue`

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

5.2.6 Languages written in Right to Left

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

5.2.7 Memory limits

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

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

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

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

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

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

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

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

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

5.3 Lineation commands

5.3.1 Disabling lineation

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

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. 22).
 - You want to freeze the line numbering for some line, but still keep critical footnotes. In this case, use `\startlock` (5.4.2 p. 24).

A successor to `reledmac` may disable this feature.

5.3.2 Setting lineation start and step

`\firstlinenum`
`\linenumincrement`

By default, `reledmac` numbers every 5th line. There are two counters that control this 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`
`\sublinenumincrement`
`\linenumberlist`

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

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

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

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

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

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

5.3.3 Setting lineation reset

`\lineation`

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

You may only use this command at places where numbering is not in effect; you can't change the lineation system within a section. You can change it between sections: they don't all have to use the same lineation system. The package's standard setting is

`\lineation{section}`. If the lineation is by `pstart`, the `pstart` number will be printed before the line number in the notes.

5.3.4 Setting line number margin

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

`\linenummargin{left}`

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

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

5.3.5 Other settings

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

5.4 Changing the line numbers

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

5.4.1 Sublineation

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

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

`\Xsublinesep` You can change the separator between the line number and the subline number ei-
`\Xsublinesepside` ther by using `\Xsublinesep` without any optional argument (7.4.11 p. 49) or by using `\Xsublinesepside`. In the second case, it will change the separator only for line numbers in the margins, not in the footnotes.

5.4.2 Locking lineation

`\startlock` The `\startlock` command, used in running text, locks the line number at its current value, until you insert `\endlock`. It can tell for itself whether you are in a patch of line or sub-line numbering. One use for line-number locking is in printing poetry: there the line numbers should be those of verse lines rather than of printed lines, even when a verse line requires several printed lines. But in this case you may use the `\stanza` mechanism, see 9 p. 58.

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

5.4.3 Setting and changing line number

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

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

5.4.4 Line number style

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

`\sublinenumberstyle`

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

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

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

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

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

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

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

5.4.5 Skipping and hiding number

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

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

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

5.5 Adding annotations to line numbers

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

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

In such instances, you must add the second number manually, as `reledmac` can't determine a general pattern for such numbering, which depends too heavily upon the edited text.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.
- 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`.

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

- 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. 33) 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. 64).

¹⁵ As for all \TeX macro with no argument.

- `\Xwraplinenumannotation`{*<cmd>*} for annotations in the margins only.
Note that `\Xwraplinenumannotation` without any optional argument will override this setting.
- `\Xwraplinenumannotationref`{*<cmd>*} for annotations referenced using the `\annotationref` command of the crossref mechanism (11 p. 64).
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>*]

```
\Xnoidenticallinenumannotation
\Xendnoidenticallinenumannotation
```

5.5.6 Separator between annotations

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

```
\setlinenumannotationsep
```

5.5.7 Annotations in the apparatus

Some other options for annotations in the apparatus are described in 7.4.10 p. 48

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

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

number has already been fixed. In many cases, the latter is more useful than the former. The file `examples/2-line_numbers_in_header.tex` provides an example for printing the first and last line numbers of a page in the header.

5.7 Executing code at a specific line

Sometime, you want to execute a code at a precise point in the flow of your text. For example, if the current header of the page corresponds to the position inside the text, you may want to use `\markboth` (or similar) command. The main problem is that you need to execute `\markboth` when `reledmac` typesets the text, and not when it reads it, because that could create damage with vertical spacing.

`\doinsidethislinehook`

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

For example, with the following code:

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

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

See the file `examples/2-subdivision-number-in-header.tex` for an example. Especially, the use of this command with `ℒTEX` counters is quite complex, but the example show you how to deal with this problem.

6 Apparatus commands

6.1 Terminology

We call “critical notes” notes which refer to both a lemma, that is a part of text, and a line number. Critical notes are subdivided in critical footnotes and critical endnotes.

We call “familiar notes” notes which refer to a footnote mark in the main text.

`reledmac` manages many series of notes of each category. A series of notes is identified by an uppercase letter. When the series letter is at the *beginning* of a command name, it refers to a critical footnote. When the series letter is at the *end* of a command name, it refers to a familiar footnote.

So:

- `\Afootnote` is a critical footnote of the series A.
- `\Bendnote` is a critical endnote of the series B.
- `\footnoteC` is a familiar footnote of the series C.

6.2 Critical notes

6.2.1 The lemma

`\edtext` Within numbered paragraphs, all footnotes and endnotes are generated by the `\edtext` macro:

```
\edtext{⟨lemma⟩}{⟨commands⟩}
```

The `⟨lemma⟩` argument is the lemma in the main text: `\edtext` both prints this as part of the text, and makes it available to the `⟨commands⟩` you specify to generate notes.

For example:

```
I am happy:
I saw my friend \edtext{Smith}{
\Afootnote{Jones C, D.}}
on Tuesday.
```

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

1 Smith] Jones C, D.

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

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

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

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

1 Smith] Jones C, D.

1–2 I saw my friend Smith on Tuesday.] The
date was July 16, 1954.

However, `\edtext` cannot handle overlapping but unnested notes—for example, one note covering lines 10–15, and another covering 12–18; an `\edtext` that starts in the `⟨lemma⟩` argument of another `\edtext` must end there, too. (The `\lemma` and `\linenum` commands may be used to generate overlapping notes if necessary.)

6.2.2 Footnotes

The second argument of the `\edtext` macro, `⟨commands⟩`, may contain a series of subsidiary commands that generate various kinds of notes.

`\Afootnote` Five separate series of footnotes are maintained; each macro takes one argument like `\Afootnote{⟨text⟩}`. When all of the five are used, the A notes appear in a layer just below the main text, followed by the rest in turn, down to the E notes at the bottom. `\Cfootnote` These are the main macros that you will use to construct the critical apparatus of your text. `\Dfootnote` `\Efootnote`

If you need more series of critical notes, please look at 6.7.1 p. 42.

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. 46).
- `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. 49).
- `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. 51).
- `linangesep=<c>` to change to `<c>` the separator between start line and end line for this particular note.

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

6.2.3 Endnotes

`\Aendnote` **Inserting endnotes** The package also maintains five separate series of endnotes.
`\Bendnote` If you do not need the endnotes facility, you should use `noend` option when loading
`\Cendnote` `reledmac`.
`\Dendnote` The mechanism is similar to the one for footnotes: each macro takes one or more
`\Eendnote` optional arguments and one single argument, like:
`\Aendnote[<option>]{<text>}`.

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

- `fulllines` to disable `\Xendtwolines` and `\Xendmorethantwolines` features for this particular note (cf. 7.4.5 p. 46).
- `nonum` to disable line number for this particular note.
- `nosep` to disable the lemma separator for this particular note. A horizontal blank space is added instead. You can use `\Xendinplaceoflemmaseparator` to set it (7.5.2 p. 51).
- `linangesep=<c>` to change to `<c>` the separator between start line and end line for this particular note.

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

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

```
\section{Endnotes}
```

```

\subsection{First text}
\doendnotesbysection{A}
\doendnotesbysection{B}
\subsection{Second text}
\doendnotesbysection{A}
\doendnotesbysection{B}

```

Note that by default inside endnotes no separator is used between the lemma and the content. However you can use the `\Xendlemmaseparator` macro to define one (7.5.2 p. 51).

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

`\toendnotes`
`\Xtoendnotes`

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

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

For example:

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

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

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

However, in some cases, you may want to write a once-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!

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.

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. 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. 64) 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.

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)

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`.

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}
           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

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

`\lemma` is called in a `\edtext` “of first level”, $\langle X \rangle$ is equal to 1. If the lemma is called in both 1 and 2 `\edtext` depth, $\langle X \rangle$ is 1,2. If that word is referenced in the lemma of every `\edtext` depth, $\langle X \rangle$ can also be set to `inlemma`.

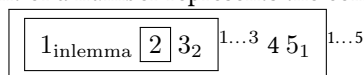
Note that only words that are actually referenced in a `\lemma` need the optional argument. Therefore, the first `\sameword` in the example above should have “1” as its optional argument, to be referenced correctly in the lemma.

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

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

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

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



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

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

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

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

Here, the corresponding code:

```

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

```

6.3.6 Sameword for a group of words

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

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

```

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

```

1 per causam tamen scire causam est per causam cognoscere causam

1 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}}
\edtext{\sameword{causam}}{\Bfootnote{fnote}}
```

1 per causam tamen scire causam est per causam cognoscere causam

```
1 causam2] fnote
1 per causam2] causam rei B
1 causam4] fnote
```

6.3.7 Customizing

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

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

6.3.8 Problems with some macros

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

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

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

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

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

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

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

6.3.9 Automatic sameword annotation

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

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

6.4 Apparatus of manuscripts

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

To produce an apparatus of manuscripts with *reledmac*, you have to insert specific commands that are used to mark the sections for which only part of the manuscripts are relevant. These commands will be processed, and **after the second T_EX run**, corresponding apparatuses of manuscripts will be inserted in the first (viz. ‘A’ series) level of footnotes.

As the insertion of this apparatus can change the page breaks, you may have to run T_EX two or more times. We strongly recommend to use tools like *latexmk* to do that.

6.4.1 Marking sections of text

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

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

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

`\stopmsdata`

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

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

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

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

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

6.4.2 Layout of the apparatus of manuscripts

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

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

6.4.3 Settings

As the apparatus of manuscripts technically consists of first-level critical notes (‘A’ series), any setting available for critical notes can be applied (7 p. 43). 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``\footnoteB``\footnoteC``\footnoteD``\footnoteE`

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

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

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

6.5.2 Customizing mark

`\thefootnoteA``\bodyfootmarkA``\footfootmarkA`

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

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

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

```
\newcommand*{\bodyfootmarkA}{%
```

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

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

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

There are similar command triples for the other series.

6.5.3 Separator for multiple footnotes

The `footmisc` package [Fai03] by Robin Fairbairns has an option whereby sequential footnote marks in the text can be separated by commas^{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` `\footnoteXtext` In certain cases, you can't directly use `\footnoteX`; for example, when using `\uline` command of the `ulem` package. You need to print the footnote mark first, then call the footnote text to be inserted.

For all $\langle X \rangle$ command, `reledmac` provides a `\footnote $\langle X \rangle$ mark` command and a `\footnote $\langle X \rangle$ text` command, equivalent to standard \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` `\seriesatend` However in some specific cases, you need to change the series order at some point inside the document. You can use `\seriesatbegin $\langle s \rangle$` to pull up a given series $\langle s \rangle$ to the beginning, or `\seriesatend $\langle s \rangle$` to push it down to the end.

6.8 Position of critical and familiar footnotes

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

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

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

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

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

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

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

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

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

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

7 Apparatus customization

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

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

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

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

- `paragraph` formats all of the footnotes of a series as a single paragraph; if you use this arrangement, you are strongly encouraged to read 19.1.6 p. 81.
- `twocol` formats them as separate paragraphs, but in two columns;
- `threecol`, in three columns.
- `normal`, restore normal arrangement.

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

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

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

`\hsize` has been set for the pages that use this series of notes; otherwise T_EX 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.

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

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

7.4.2 Print page number only at first time

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

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

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

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

In any case, you can use:

- | | |
|---------------------------------------|---|
| <code>\Xendsympagenum</code> | • <code>\Xendsympagenum[⟨series⟩]{⟨c⟩}</code> to print <code>⟨c⟩</code> when the page number is not printed. |
| <code>\Xendinplaceofpagenumber</code> | • <code>\Xendinplaceofpagenumber[⟨series⟩]{⟨l⟩}</code> to print a <code>⟨l⟩</code> length horizontal space in case no symbol is printed instead of the page number. |

7.4.3 Arbitrary text before line number

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

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

`\Xendafterpagenumber` `\Xendafterpagenumber[⟨s⟩]{⟨text⟩}` defines the text after the page number in endnotes. Default value is) (open parenthesis followed by a single space). `\Xendlineprefixsingle[⟨s⟩]` defines the text before the line number in endnotes when there is only one line. Default value is empty. `\Xendlineprefixmore[⟨s⟩]{⟨text⟩}` defines the text before the line number in endnotes when there is more than one line. Default value is empty. If you don’t define it, it will use the value defined by `\Xendlineprefixsingle`.

7.4.4 Separator for line range

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

7.4.5 Abbreviate line range

`\Xtwolines` `\Xmorethantwolines` If a lemma is printed on two subsequent lines, `reledmac` will print the first and the last line numbers. Instead of this, it is also possible to print an abbreviation which stands for “line 1 and subsequent line(s)”.

To achieve this, use `\Xtwolines[⟨s⟩]{⟨text⟩}` and `\Xmorethantwolines[⟨s⟩]{⟨text⟩}`. The `⟨text⟩` argument of `\Xtwolines` will be printed if the lemma is on two lines, and the `⟨text⟩` argument of `\Xmorethantwolines` will be printed if the lemma is on three or more lines. For example:

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

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

If you use `\Xtwolines` without setting `\Xmorethantwolines`, the $\langle text \rangle$ 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.

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

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

For endnotes, use these macros: `\Xendtwolines`; `\Xendmorethantwolines`; `\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 Printing pstart number

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

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

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

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

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

7.4.8 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.

7.4.9 Disabling line number printing

<code>\Xnolinenumber</code>	Sometime, we want to print only the stanza or pstart number, but not the line number. The <code>\Xnolinenumber[⟨s⟩]</code> macro disables line number printing for critical footnotes.
<code>\Xendnolinenumber</code>	The <code>\Xendnolinenumber</code> macro does the same for endnotes. Note that contrary to <code>\Xnonumber</code> and <code>\Xnoendnum</code> , the page/pstartannotation will be printed, if required.

7.4.10 Options for line number annotations

<code>\Xlinenumannotationonlyfirst</code>	By default, the line number annotation (5.5 p. 25) 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>\Xlinenumannotationonlyfirst</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 <code>⟨symbol⟩</code> must be robust. Use <code>\robustify</code> to robustify a non-robust command.
<code>\Xendlinenumannotationonlyfirst</code>	For endnotes, <code>\Xendlinenumannotationonlyfirst</code> ;
<code>\Xendlinenumannotationonlyfirstintwo</code>	and <code>\Xendsymlinenumannotation</code> are the equivalents of
<code>\Xendsymlinenumannotation</code>	<code>\Xlinenumannotationonlyfirst</code> ;
<code>\Xnolinenumberifannotation</code>	<code>\Xlinenumannotationonlyfirstintwo</code> and <code>\Xsymlinenumannot</code>
<code>\Xendnolinenumberifannotation</code>	The <code>\Xnolinenumberifannotation</code> disable line number printing only for critical notes when there are some annotations to the line number. The <code>\Xendnolinenumberifannotation</code> is the same for endnotes.

7.4.11 Separator between line and subline numbers

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

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

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

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

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

7.4.12 Separator between page and line numbers

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

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

7.4.13 Space around number

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

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

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

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

7.4.14 Space around line symbol

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

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

`\Xendbefore Symlinenum` `\Xendbefore Symlinenum` and `\Xendafter Symlinenum` are the equivalents of `\Xbefore Symlinenum` and `\Xafter Symlinenum` for the endnotes.

7.4.15 Space in place of number

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

`\Xendinplace ofnumber` `\Xendinplace ofnumber[⟨s⟩]{⟨l⟩}` is the same, for critical endnotes.

7.4.16 Boxing line number and line symbol

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

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

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

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

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

L to align left (default value);

R to align right;

C to center.

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

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

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

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

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

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

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

`\Xendboxendlinenumalign`

7.5 Separator between the lemma and the note

7.5.1 For footnotes

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

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

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

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

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

7.5.2 For endnotes

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

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

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

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

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

7.6 Font style

7.6.1 For line number

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

`\Xendnotenumfont` `\Xendnotenumfont[⟨s⟩]{⟨command⟩}` is used to change the font style for line

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

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

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

7.6.2 For the lemma

`\Xlemmadisablefontselection` By default, font of the lemma in footnote is the same as font of the lemma in the main text. For example, if the lemma is in italic in the main text, it is also in italic in note. The `\Xlemmadisablefontselection[\langle s \rangle]` command allows to disable it for a specific series.

`\Xendlemmadisablefontselection` By default, font of the lemma in endnote is the same as font of the lemma in the main text. For example, if the lemma is in italic in the main text, it is also in italic in note. The command allows `\Xendlemmadisablefontselection[\langle s \rangle]` to disable it for a specific series.

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

`\Xendlemmafont`

`\Xlemmafont{\bfseries}`

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

7.6.3 For all notes

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

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

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

7.7 Wrapping notes

7.7.1 Wrapping lemmas

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

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

7.7.2 Wrapping contents

<code>\Xwrapcontent</code>	<code>\Xwrapcontent[⟨s⟩]{⟨cmd⟩}</code> is used to wrap the footnote contents — excluding the lemma — in a \TeX command.
	For example, if the language of your note is not the same as the language of the lemma, use <code>\Xwrapcontent{\foreignlanguage{⟨language⟩}}</code> (with <code>babel</code>) or <code>\Xwrapcontent{\text{⟨language⟩}}</code> (for <code>babel</code>).
<code>\Xendwrapcontent</code>	<code>\Xendwrapcontent[⟨s⟩]{⟨cmd⟩}</code> is the same for endnotes.
<code>\wrapcontentX</code>	<code>\wrapcontentX[⟨s⟩]{⟨cmd⟩}</code> is the same for familiar footnotes.

7.8 Indent of notes content

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

7.9 Arbitrary code

7.9.1 Arbitrary code around line number

<code>\Xendbhooklinenumber</code>	<code>\Xendbhooklinenumber[⟨s⟩]{⟨code⟩}</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[⟨s⟩]{⟨code⟩}</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[⟨s⟩]{⟨code⟩}</code> is used to execute code before a space or symbol which replaces line number in endnotes. The code is executed before the <code>\Xendbeforelymlinenumber</code> space and before the <code>\Xendnotenumfont</code> font setting.
<code>\Xendahookinplaceofnumber</code>	<code>\Xendahookinplaceofnumber[⟨s⟩]{⟨code⟩}</code> is used to execute code after a space or symbol which replaces line number in endnotes. The code is executed after the <code>\Xendaftersymlymlinenumber</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}.}}
```

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

7.9.3 Arbitrary code before inserting note

`\Xbeforeinserting` and `\beforeinsertingX` are very technical commands.

They allow one to add any arbitrary code just before the footnotes are added in the list of footnotes. The main use is to insert text direction code. For example, if you edit right-to-left text with `bid`, 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. 52).

Note that the changes are local to the footnote.

7.10 Options for footnotes in columns

7.10.1 Alignment

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

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

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

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

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

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

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

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

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

7.10.2 Size of the columns

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

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

are displaying in two columns. Default value is `.45 \hspace`.

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

7.11 Options for paragraphed footnotes and notes grouped by line

7.11.1 Mark separation of notes

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

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

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

7.11.2 Ragged text

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

In many cases, you might like to use it in combination with `\Xnumberonlyfirstinline` (7.4.1 p. 45).

`\groupbylineseparetwolines` Note that the `\Xafternote` and `\Xparafootsep` settings are used to determine space and content between footnotes (7.11 p. 55). 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[\langle s \rangle]`.

In many cases, you might like to use it in combination with `\Xnumberonlyfirstintwolines` (7.4.1 p. 45).

7.12.2 Text before notes

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

`\txtbeforenotesX`

You can add text before familiar footnotes with `\txtbeforenotes[⟨s⟩]{⟨text⟩}`.
`\Xendtxtbeforenotes` You can add text before endnotes with `\Xendtxtbeforenotes[⟨s⟩]{⟨text⟩}`. The text will be typeset only if there are endnotes.
`\Xtxtbeforenotesonlyonce` By default, such texts are inserted at the beginning of the groups of notes on each page. You can add `\Xtxtbeforenotesonlyonce` (for critical footnotes) and `\txtbeforenotesonlyonceX` (for familiar footnotes) to insert them only the first time notes are typeset.

7.12.3 Code before notes

`\Xbhookgroup` While `\Xtxtbeforenotes` is for typesetting text before notes, `\Xbhookgroup[⟨s⟩]{⟨code⟩}`
`\bhookgroupX` 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 todoclist.

7.12.7 Width

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

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

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

Note that changes the with of the block of notes. If you want to change the width of each column when typesetting notes in columns, use `\Xhsizetwocol`, `\Xhsizethreecol`, `\hsizetwocolX`, `\hsizethreecolX`, see 7.10.2 p. 54.

7.13 Footnotes and the *reledpar* columns

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

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

7.14 Endnotes in one paragraph

`\Xendparagraph` By default, any new endnote starts a new paragraph. Use `\Xendparagraph[⟨s⟩]` to have all end notes of one given series set in one paragraph.

`\Xendafternote` You can add some space after a endnote series by using `\Xendafternote[⟨s⟩]{⟨l⟩}`. The default value is `1em plus.4em minus.4em`.

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

8 Fonts

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

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

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

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

`\select@lemmafont` We will briefly discuss `\select@lemmafont` here because it is important to know about it now, although it is not one of the macros you would expect to change in the course of a simple job. Hence it is “protected” by having the `@`-sign in its name.

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

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

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

9 Verse

9.1 Basic

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

If you need to add brackets directly after `\stanza`, `&` or `\&`, add `\norelax`. Otherwise, the brackets will be interpreted as delimitation of an optional argument (cf. 9.9 p. 61)

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. 61.

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. 407.

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{stanzaindentrepetition}{1}
\setcounter{startstanzaindentrepetition}{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
\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 `\setstanzaindent` for this verse is skipped, and `{<value>}` is used instead.

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

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

9.6 Stanza breaking

```
\setstanzapenalties
```

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

The command

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

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

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

9.7 Hanging symbol

It is possible to insert a symbol in each line of hanging verse, as in French typography; for example, the opening bracket “[”. To insert it in `reledmac`, use macro `\sethangingsymbol{⟨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. 79 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.
- 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`.
- 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` (7.4.8 p. 48).

`\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 expand argument, you can use one the following syntax:

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

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

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

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

9.12 Various tools

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

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

9.13 Notes on empty lines

Since v2.3.0 of `reledmac`, empty lines when typesetting verses no longer produce new paragraphs, and consequently, do not insert vertical spaces. Use optional argument of `\stanza` or `\newverse` to insert vertical space (9.9 p. 61).

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**\edlineref****\sublineref****\pstartref****\annotationref**

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

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

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

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

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

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

11.2 Cross-referencing to a critical note

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

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

If you want to refer to starting and ending lines, you should use `\appref` and related tools (11.6.2 p. 67).

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. 33).

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. 33) and `\lineannot` (5.5.2 p. 27) and sets the beginning page, line, subline numbers and line annotations to those of the place where `\edlabel{mouse}` was placed, and the ending ones to those where `\edlabel{elephant}` occurs.

For example, one might use the following:

```
\beginnumbering
```

```

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

\pstart
\edtext{Et videtur quod non\edlabel{non}.}{\xxref{Queritur}{non}\lemma{queritur \dots} non}
\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 \TeX cross-referencing

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

11.6 References to start and end lines

11.6.1 Reference to main text lines

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

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

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

`\edlabelSE` Use `\edlabelSE{<label>}` to mark just one location in the text. Contrary to a 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. 33) 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` 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 `\XendbeforepagenumberSerefonly` (see 11.6.3 p. 67). 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.

- `\Xlinrangeseparator`
- `\Xmorethantwolines`
- `\Xsublinesep`
- `\Xtwolines`
- `\Xtwolinesbutnotmore`
- `\Xtwolinesonlyinsamepage`
- `\Xlinenumannotationposition`
- `\Xwraplinenumannotation`
- `\Xnoidenticallylinenumannotation`

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`
- `\Xendlinrangeseparator`
- `\Xendmorethantwolines`
- `\Xendsublinesep`
- `\Xendtwolines`
- `\Xendtwolinesbutnotmore`
- `\Xendtwolinesonlyinsamepage`
- `\Xendlinenumannotationposition`
- `\Xendwraplinenumannotation`
- `\Xemdnoidenticallylinenumannotation`

For one specific command When calling `\appref` and `\SEref`, you can use as a first optional argument, in brackets (`[]`), any optional argument which can be used for critical footnotes (6.2.2 p. 30).

When calling `\apprefwithpage`, `\SErefwithpage` or `\SErefonlypage` you can use as a first optional argument, in brackets (`[]`), any optional argument which can be used for critical endnotes (6.2.3 p. 31).

11.6.4 Combining multiple references

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

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

For example, a reference to “lines 65–66, 72, and 75” might use `\SEref[prefixmore]{ref1}`, `\SEref[noprefix]{ref1}` 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}`.

`\sidenotemargin` The package’s default setting is

`\sidenotemargin{right}`

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

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

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

12.2 Setting

12.2.1 Width

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

12.2.2 Vertical position

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

12.2.3 Distance to the main text

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

12.2.4 Font

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

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

These can of course be changed to suit.

12.2.5 Separator between notes

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

13 Indexing

13.1 Basics

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

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

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

13.2 Use with `imakeidx` or `indextools`

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

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

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

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

13.3 Referring to critical notes

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

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

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

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

`\innotemarkX`

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

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

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

For example:

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

13.4 Separator between page and line numbers

`\pagelinesep`

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

- is the default separator used by the MAKEINDEX program.

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

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

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

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:

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>.

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.

14 Glossary

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

14.1 Preamble setting

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

- Or set `glossaries`:
`\glsSetCompositor{-}`
- Or set `reledmac`:
`\renewcommand{\pagelinesep}{.}`
 In this case, the above will have consequences for your use of `\edindex` and you should set your `.ist` file (13.4 p. 72).

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>]
```

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

15 Tabular material

L^AT_EX'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.

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

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

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

For example:

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

produces the following parallel pair of verses.

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

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

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

```

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

```

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

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

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

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

```

\begin{edtabularr}
1 & & \& 2 & \& 3 & \& 4 & \& 5 & \\
Q & & \& & \& fd & \& h & \& \spreadtext{qwertziohg} & \\
v & & \& wptz & \& x & \& y & \& vb & \\
g & & \& nnn & \& \edrowfill{3}{5}{\upbracefill} & \& \& \\
\edrowfill{1}{3}{\downbracefill} & \& & \& pq & \& dgh & \\
k & & \& & \& 1 & \& co & \& \spreadtext{ghweropjklmnbvcxys} & \\
1 & & \& 2 & \& 3 & \& \edrowfill{4}{5}{\hrulefill} & \& \\
\end{edtabularr}

```

$$\begin{array}{cccccc}
1 & & 2 & 3 & 4 & 5 \\
Q & & & fd & h & \text{qwertziohg} \\
v & wptz & x & y & & vb \\
g & nnn & & & \overbrace{\hspace{1.5cm}} & dgh \\
& \underbrace{\hspace{1.5cm}} & & pq & & \\
k & & 1 & co & \text{ghweropjklmnbvcxys} & \\
1 & 2 & 3 & \underbrace{\hspace{1.5cm}} & &
\end{array}$$

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

```

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

```

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

```

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

```

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

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

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

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

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

For example:

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

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

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

`\edvertdots`

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

```

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. 19):

```

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

```

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

16.2 Sectioning commands with line numbering and critical notes

You have to use the following commands:

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

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

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

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

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

16.3 Optimization

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

17 Quotation environments

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

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

18 Page breaks

18.1 Control page breaking

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

```
\ledpb
\lednopb
```

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

These commands have effect only at the second run.

These two commands take effect at the beginning of line in which they are called. For example, if you call `\ledpb` at l. 444, then l. 443 will be at p. n , and l. 444 at p. $n + 1$.

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

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

18.2 Prevent page break in a long verses

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

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

19 Miscellaneous

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

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

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

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

```
\ifledfinal\else
```

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

19.1 Known and suspected limitations

19.1.1 Non-standard geometry

If you use classes other than `article` or `book`, or if you use the `geometry` package, you should use `maxhnotesX` and/or `\Xmaxhnotes` as explained in 7.12.6 p. 56 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. 78 in order to know which are the available options.

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

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

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

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

19.1.4 Marginal notes

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

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

19.1.5 Paragraph shape

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

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

19.1.6 Paragraphed footnotes

The restriction on explicit line-breaking in paragraphed footnotes, mentioned on 7.2 p. 44, and described in more detail on XIII.6.3 p. 200, really is a nuisance if that is something you need to do. There are some possible solutions, described by Michael Downes, but this area remains unsatisfactory.

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

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

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

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

Any settings to “geometry” must be made before `\Xarrangement` / `\arrangementX`.

Finally, in many cases you should use `\Xmaxhnotes` and / or `\maxhnotesX` (7.12.6 p. 56), 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 \LaTeX whinging “Missing { inserted”, and then things start to fall apart. The trick in this case is to specify either:

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

or

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

(`\@secondoftwo` is an internal \LaTeX 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

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

I Implementation overview

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

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

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

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

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

II Preliminaries

II.1 Links with original `edmac`

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

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

II.2 Package declaration

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

```

1 %<*code>
2 \NeedsTeXFormat{LaTeX2e}
3 \ProvidesPackage{reledmac}[2021/02/14 v2.36.1 typesetting critical editions]
4 %

```

II.3 Package options

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

```

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

```

II.3.1 Options of reledpar

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

```

\newif\ifparledgroup
\newif\ifsameparallelpagenumber%
\newif\ifprevpgnotnumbered%%
%

```

II.3.2 Options of reledmac

```

11 \DeclareOptionX{series}[A,B,C,D,E]{\xdef\default@series{#1}}
12 \ExecuteOptionsX{series}%
13
14 \newif\if@noeled@sec%
15 \DeclareOptionX{noeledsec}{\@noeled@sectrue}
16
17 \newif\ifnocritical@%
18 \DeclareOptionX{nocritical}{\nocritical@true}%
19
20 \newif\ifnofamiliar@%
21 \DeclareOptionX{nofamiliar}{\nofamiliar@true}%
22
23 \newif\ifnoledgroup@%
24 \DeclareOptionX{noledgroup}{\noledgroup@true}%
25
26 \newif\ifnoend@%
27 \DeclareOptionX{noend}{%
28   \let\l@dend@open\@gobble%
29   \let\l@dend@close\relax%
30   \global\let\l@dend@stuff=\relax%
31   \noend@true%
32 }%

```

```

33 \newif\ifnoquotation@
34 \DeclareOptionX{noquotation}{\noquotation@true}
35
36
37 \newif\ifledfinal
38 \DeclareOptionX{final}{\ledfinaltrue}
39 \DeclareOptionX{draft}{\ledfinalfalse}
40 \ExecuteOptionsX{final}
41
42 \newif\ifparapparatus@
43 \DeclareOptionX{parapparatus}{\parapparatus@true}
44
45 \newif\iflednopbinverse
46 \DeclareOptionX{nopbinverse}{\lednopbinversetrue}
47
48 \newif\ifwidthliketwocolumns%
49 \DeclareOptionX{widthliketwocolumns}{\widthliketwocolumnstrue}%
50
51 \newif\ifcontinuousnumberingwithcolumns
52 \DeclareOptionX{continuousnumberingwithcolumns}{\continuousnumberingwithcolumnstrue}%
53
54 \newif\ifxindy@
55 \DeclareOptionX{xindy}[eledmac-markup-attr.xdy]{%
56   \AtBeginDocument{\immediate\openout\eledmac@xindy@out=#1}%
57   \newwrite\eledmac@xindy@out%
58   \xindy@true%
59   \gdef\eledmacmarkuplocdepth{:depth 1}%
60   \AtEndDocument{\immediate\closeout\eledmac@xindy@out}%
61 }%
62
63 \newif\ifxindyhyperref@
64 \DeclareOptionX{xindy+hyperref}{%
65   \xindyhyperref@true%
66 }%
67
68 \newif\ifeledmaccompat@%
69 \DeclareOptionX{eledmac-compat}{%
70   \eledmaccompat@true%
71 }%
72 \DeclareOptionX{nopenalties}{%
73   \AtBeginDocument{\let\add@penalties\relax}%
74 }
75 \def\l@auxdir{}%
76 \DeclareOptionX{auxdir}{%
77   \xdef\l@auxdir{#1}%
78 }%
79
80 \newif\ifsw@caseinsensitive%
81 \DeclareOptionX{swcaseinsensitive}{%

```

```

82 \sw@caseinsensitivetrue%
83 }%
84 \newif\ifnoresetlinenumannotation@
85 \DeclareOptionX{noresetlinenumannotation}{%
86   \noresetlinenumannotation@true%
87 }%
88 \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 `XgTeX` 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}

```

```

112 \RequirePackage{ifxetex}%
113 %

```

II.5 Compatibility with LuaTeX

Here, we enable some primitives for LuaTeX.

```

114 \ifx\directlua\undefined\else%
115   \directlua{tex.enableprimitives("",{"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%
123 %

```

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 \TeX '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@samewordRTL 58 \newcommand*{\led@err@samewordRTL}{%
159 \reledmac@error{You can't use \string\sameword \MessageBreak with XeLaTeX
\MessageBreak when typesetting RTL text.\MessageBreak Please use LuaTeX
instead.}\@ehc}%
160 }%
161 %

\led@err@edtextoutsidestart 62 \newcommand*{\led@err@edtextoutsidestart}{%
\led@err@PstartInEdtext 63 \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@NotesChanged 73 \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@SectionContinued 82 \newcommand*{\led@mess@SectionContinued}[1]{%
183 \message{Section #1 (continuing the previous section)}}
184 %

\led@err@LineationInNumbered 85 \newcommand*{\led@err@LineationInNumbered}{%
186 \reledmac@error{You can't use \string\lineation\space within
187 a numbered section}{\@ehc}}
188 %

\led@warn@BadLineation 89 \newcommand*{\led@warn@BadLineation}{%
\led@warn@BadLinenummargin 90 \reledmac@warning{Bad \string\lineation\space argument}}
\led@warn@BadLockdisp 91 \newcommand*{\led@warn@BadLinenummargin}{%
\led@warn@BadSubblockdisp

```

```

192 \reledmac@warning{Bad \string\linenummargin\space argument}}
193 \newcommand*{\led@warn@BadLockdisp}{%
194 \reledmac@warning{Bad \string\lockdisp\space argument}}
195 \newcommand*{\led@warn@BadSublockdisp}{%
196 \reledmac@warning{Bad \string\sublockdisp\space argument}}
197 %

```

```

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

```

```

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

```

```

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

```

```

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

```

```

\led@err@PstartNotNumbered16 \newcommand*{\led@err@PstartNotNumbered}{%
\led@err@PstartInPstart17 \reledmac@error{\string\pstart\space must be used within a
\led@err@PendNotNumbered18 numbered section %
\led@err@PendNoPstart19 (\string\beginnumbering\space ...\space\string\
\led@err@AutoparNotNumbered endnumbering)}{\@ehc}}%
\led@err@NumberingWithoutPstart20 \newcommand*{\led@err@PstartInPstart}{%
221 \reledmac@error{\string\pstart\space encountered while another
222 \string\pstart\space was in effect}{\@ehc}}
223 \newcommand*{\led@err@PendNotNumbered}{%
224 \reledmac@error{\string\pend\space must be used within a
225 numbered section}{\@ehc}}
226 \newcommand*{\led@err@PendNoPstart}{%
227 \reledmac@error{\string\pend\space must follow a \string\pstart}{\@ehc}}
228 \newcommand*{\led@err@AutoparNotNumbered}{%

```

```

229 \reledmac@error{\string\autopar\space must be used within a
230         numbered section}{\@ehc}}
231 \newcommand*{\led@err@NumberingWithoutPstart}{%
232 \reledmac@error{\string\beginnumbering...\string\endnumbering\space
without \string\pstart}{\@ehc}}%
233 %

```

```

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

```

```

\led@warn@DuplicateLabel237 \newcommand*{\led@warn@DuplicateLabel}[1]{%
ppLabelOutSecondArgEdtext238 \reledmac@warning{Duplicate definition of label `#1'\@gobble}%
\led@warn@RefUndefined239 \@latex@warning@no@line{Label `#1' multiply defined}%
\led@warn@RefUndefined240 }%
241 \newcommand*{\led@warn@AppLabelOutSecondArgEdtext}[1]{%
242 \reledmac@warning{\string\apptocpart\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@BadHookSeries59 \newcommand{\led@warn@BadHookSeries}[2]{%
260 \reledmac@warning{Bad series #1 for #2 hook. Skipped.}%
261 }%
262 %

```

```

\led@warn@BadSidenotemargin63 \newcommand*{\led@warn@BadSidenotemargin}{%
264 \reledmac@warning{Bad \string\sidenotemargin\space argument}}
265 %

```

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

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

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

```
\led@err@ManySidenotes276 \newcommand{\led@err@ManySidenotes}{%
\led@err@ManyLeftnotes277   \ifledRcol{%
\led@err@ManyRightnotes278   \reledmac@warning{\itemcount@space sidenotes on line \the\line@numR\
space p. \the\page@numR}%
279   \else%
280   \reledmac@warning{\itemcount@space sidenotes on line \the\line@num\
space p. \the\page@num}%
281   \fi%
282 }%
283 \newcommand{\led@err@ManyLeftnotes}{%
284   \ifledRcol{%
285   \reledmac@warning{\itemcount@space leftnotes on line \the\line@numR\
space p. \the\page@numR}%
286   \else%
287   \reledmac@warning{\itemcount@space leftnotes on line \the\line@num\
space p. \the\page@num}%
288   \fi%
289 }%
290 \newcommand{\led@err@ManyRightnotes}{%
291   \ifledRcol{%
292   \reledmac@warning{\itemcount@space rightnotes on line \the\line@numR\
space p. \the\page@numR}%
293   \else%
294   \reledmac@warning{\itemcount@space rightnotes on line \the\line@num\
space p. \the\page@num}%
295   \fi%
296 }%
297 %
```

```
\led@err@TooManyColumns298 \newcommand*{\led@err@TooManyColumns}{%
\led@err@UnequalColumns299   \reledmac@error{Too many columns}{\@ehc}}
\led@err@LowStartColumn
\led@err@HighEndColumn
\led@err@ReverseColumns
```

```

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

```

```

endnotes@outsidenumbering10 \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 %

```

```

err@EdtextWithoutFootnote14 \newcommand\led@err@EdtextWithoutFootnote}{%
315   \reledmac@error{edtext without Xfootnote. Check syntax}{\@ehc}%
316 }%
317 %

```

```

tnoteNotInSecondArgEdtext18 \newcommand\led@err@FootnoteNotInSecondArgEdtext}[1]{%
319   \reledmac@error{#1footnote outside of the second argument of an edtext.
    Check syntax}{\@ehc}%
320 }%
321 %

```

```

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

```

```

error@fail@patch@@makecol26 \newcommand\led@error@fail@patch@@makecol}{%
327   \reledmac@error{Fail to patch \string\@makecol\space command}{\@ehc}%
328 }%
329 %

```

```

ror@fail@patch@@reinserts30 \newcommand\led@error@fail@patch@@reinserts}{%
331   \reledmac@error{Fail to patch \string\@reinserts\space command}{\@ehc}%
332 }%
333 %

```

```

\led@error@fail@patch@@doclearpage34 \newcommand{\led@error@fail@patch@@doclearpage}{%
335 \reledmac@error{Fail to patch \string\@doclearpage\space command}{\@ehc}%
336 }%
337 %

\led@error@fail@patch@@iiiminipage38 \newcommand{\led@error@fail@patch@@iiiminipage}{%
339 \reledmac@error{Fail to patch \string\@iiiminipage\space command}{\@ehc}%
340 }%
341 %

\led@error@fail@patch@endminipage42 \newcommand{\led@error@fail@patch@endminipage}{%
343 \reledmac@error{Failed to patch the \string\endminipage\space command}{\@ehc}%
344 }%
345 %

\led@error@fail@patch@endminipage46 \newcommand{\led@error@fail@patch@makeindex}{%
347 \reledmac@error{Failed to patch the \string\makeindex\space command}{\@ehc}%
348 }%
349 %

\led@error@fail@patch@setcounter50 \newcommand{\led@error@fail@patch@setcounter}{%
351 \reledmac@error{Failed to patch the \string\setcounter\space command}{\@ehc}%
352 }%
353 %

\led@error@fail@patch@addtocounter54 \newcommand{\led@error@fail@patch@addtocounter}{%
355 \reledmac@error{Failed to patch the \string\addtocounter\space command}{\@ehc}%
356 }%
357 %

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

\led@warn@edinde@outsidenumbering62 \newcommand{\led@warn@edinde@outsidenumbering}{%
363 \reledmac@warning{\string\edindex\space called outside of \string\beginnumbering\space ... \space \string\endnumbering. \MessageBreak Automatically switched to \string\index.}%
364 }%
365 %

```

```

warning@hsizeX@deprecated 366 \newcommand{\led@warning@hsizeX@deprecated}{%
367 \reledmac@warning{\string\hsizeX\space command deprecated, use \string\
widthX\space instead.}%
368 }%
369 %

```

```

warning@Xhsize@deprecated 370 \newcommand{\led@warning@Xhsize@deprecated}{%
371 \reledmac@warning{\string\Xhsize\space command deprecated, use \string\
Xwidth\space instead.}%
372 }%
373 %

```

```

warning@msdatawithoutstop 374 \newcommand{\led@warning@msdatawithoutstop}{%
375 \reledmac@warning{\string\msdata\space without corresponding \string\
stopmsdata}%
376 }%
377 %

```

```

ning@preXnotes@deprecated 378 \newcommand{\led@warning@preXnotes@deprecated}{%
379 \reledmac@warning@preXnotes@deprecated%
380 }%
381 %

```

II.8 Gobbling

Here, we define some commands which gobble their arguments.

```

\@gobblethree 382 \providecommand*\@gobblethree}[3]{}
\@gobblefour 383 \providecommand*\@gobblefour}[4]{}
\@gobbleseven 384 \providecommand*\@gobbleseven}[7]{}
385 %

```

II.9 Wrapping content

```

\l@wrap@ifnotemptybox
\l@wrapcs@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.

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

```

386 \newcommand{\l@wrap@ifnotemptybox}[2]{%
387   \setbox0=\hbox{#2}%
388   \ifdim\wd0=\z@ \else%
389     #1{#2}%
390   \fi%
391 }%
392 \newcommand{\l@wrapcs@ifnotemptybox}[2]{%
393   \l@wrap@ifnotemptybox{\csname #1\endcsname}{#2}%
394 }%
395 %

```

II.10 Miscellaneous commands

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

```

396 \ifledfinal
397   \newcommand*{\showlemma}[1]{#1}
398 \else
399   \newcommand*{\showlemma}[1]{\textbf{#1}}%
400 \fi
401
402 %

```

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

Initialize it as `\empty`.

```

403 \let\linenumberlist=\empty
404
405 %

```

`\@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).

```

406 \newcount\@l@tempcnta \newcount\@l@tempcntb
407 %

```

II.11 Prepare reledpar

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

```

408 \newif\ifl@dpairing
409 \newif\ifl@dpaging%
410 \newif\ifl@dprintingpages%

```



```

411 \newif\ifl@dprintingcolumns%
412 \newif\ifpst@rtedL
413 \newcount\l@dnumpstartsL
414 %

```

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

```

415 \newif\ifledRcol
416 \newif\ifledRcol@
417 %

```

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

```

418 \newif\ifnumberingR
419 %

```

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

```

420 \newif\ifafterendnumberingR%
421 %

```

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

```

422 \newif\ifXnote@%
423 %

```

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

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

```

424 \providebool{indtl@innote}%
425 \providebool{indtl@notenumber}%
426 %

```

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

```

427 \newif\ifinstanzaL%
428 \newif\ifinstanzaR%
429 %

```

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

```
430 \newif\ifinastanzaL%
431 \newif\ifinastanzaR%
432 %
```

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 `\resumenumbering` 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.

```
433 \newcount\section@num
434 \section@num=0
435 \let\extensionchars=\empty
436 %
```

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

```
437 \newif\ifnumbering
438 %
```

`\beginnumbering` `\beginnumbering` begins a section of numbered text. When it is executed we increment
`\initnumbering@reg` 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.

```

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

```

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.

```

467 \else
468   \begingroup
469   \global\@afterindenttrue%In order to reestablish normal feature if the \
  begingroup was not here
470   \initnumbering@quote
471   \ifwidthliketwocolumns%
472     \setwidthliketwocolumns%

```

```

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

```

`\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.

```

515 \def\endnumbering{%
516   \ifnumbering
517     \global\numberingfalse
518     \normal@pars
519     \ifnum\l@dnumstartsL=0%
520       \led@err@NumberingWithoutPstart%
521     \fi%
522 %

```

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.

```

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

```

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

```

552 \newcount\pausenumbering@page@num%

```

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

```

553 \newcommand{\pausenumbering}{%
554   \ifx\this@section@next@page@num\relax%
555     \global\pausenumbering@page@num=0%
556   \else%
557     \global\pausenumbering@page@num=\this@section@next@page@num%
558   \fi%
559   \ifautopar\global\autopar@pausetrue\fi%
560   \endnumbering\global\numberingtrue}
561 %

```

The `\resumenumbering` macro is a bit more involved, but not much. It does most of the same things as `\beginnumbering`, but without resetting the various counters. Note that no check is made by `\resumenumbering` to ensure that `\pausenumbering` was actually invoked. The boolean `\ifresumenumbering@start` is set to true. That allows us to avoid resetting the line number at the first line of `\resumenumbering` if the lineation is by page. This boolean is set to false after the first action.

```

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

```

```

595 \beginnumbering
596 \fi}
597
598
599 %

```

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

```

600 \newcommand{\set@continuousnumberingforL}{%
601 \ifcontinuousnumberingwithcolumns%
602 \ifl@dpairing%
603 \unless\ifl@dpaging%
604 \global\c@pstartL=\c@pstart%
605 \fi%
606 \fi%
607 \fi%
608 }%
609 %

```

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 3 list, however such migration would take quite time with some risk of error, for a gain which will be minor.

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

```

610 \newcommand*{\list@create}[1]{%
611 \global\let#1=\empty%
612 }%
613 %

```

`\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 .

```

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

```

\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.

```

618 \newtoks\led@toksa \newtoks\led@toksb
619 \global\led@toksa={\}
620 \long\def\xright@appenditem#1\to#2{%
621   \global\led@toksb=\expandafter{#2}%
622   \xdef#2{\the\led@toksb\the\led@toksa\expandafter{#1}}%
623   \global\led@toksb={}}
624 %

```

\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.

```

625 \long\def\xleft@appenditem#1\to#2{%
626   \global\led@toksb=\expandafter{#2}%
627   \xdef#2{\the\led@toksa\expandafter{#1}\the\led@toksb}%
628   \global\led@toksb={}}
629 %

```

\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.

```

630 \def\gl@p#1\to#2{\expandafter\gl@poff#1\gl@poff#1#2}
631 \long\def\gl@poff\#1#2\gl@poff#3#4{\gdef#4{#1}\gdef#3{#2}}
632
633 %

```

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.

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

```

\ifbypstart@
\byppstart@true
\byppstart@false
  \ifbypage@
    \bypage@true
    \bypage@false

```

- line-of-page: byppstart@ = false and byppage@ = true.
- line-of-pstart: byppstart@ = true and byppage@ = false.

reledmac will use the line-of-section system unless instructed otherwise.

```
634 \newif\ifbypage@
635 \newif\ifbypstart@
636 %
```

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@R37 \newif\ifbypage@R
\ifbypstart@R38 \newif\ifbypstart@R
639 %
```

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

```
640 \newcommand*{\lineation}[1]{%
641 %
```

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

```
642 \ifnumbering
643 \led@err@LineationInNumbered
644 \else
645 %
```

If the argument is page.

```
646 \def\@tempa{#1}\def\@tempb{page}%
647 \ifx\@tempa\@tempb
648 \global\bypage@true
649 \global\bypstart@false
650 \unless\ifnocritical@%
651 \Xpstart[] [false]%
652 \fi%
653 %
```

If the argument is pstart.

```
654 \else
655 \def\@tempb{pstart}%
656 \ifx\@tempa\@tempb
657 \global\bypage@false
658 \global\bypstart@true
659 \unless\ifnocritical@%
660 \Xpstart%
661 \fi%
662 %
```

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

```

663     \else
664         \def\@tempb{section}
665         \ifx\@tempa\@tempb
666             \global\bypage@false
667             \global\bystart@false
668             \unless\ifnocritical@%
669                 \Xpstart[] [false]%
670             \fi%
671 %

```

In other case, it is an error.

```

672     \else
673         \led@warn@BadLineation
674     \fi
675 \fi
676 \fi
677 \fi}}
678 %

```

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@dgetline@margin`

```

679 \newcount\line@margin%
680 \newcount\line@margin@columns%Only for parallel typesetting
681 \line@margin@columns=\m@ne%
682
683 \newcommand*{\linenummargin}[1]{%
684     \l@dgetline@margin{#1}%
685     \ifnum\@l@dttempcntb>\m@ne
686         \ifledRcol
687             \global\line@marginR=\@l@dttempcntb
688             \led@warn@setting@in@rightside{\linenummargin}%
689         \else
690             \global\line@margin=\@l@dttempcntb
691         \fi
692     \fi}}
693
694 \newcommand*{\l@dgetline@margin}[1]{%
695     \def\@tempa{#1}\def\@tempb{left}%
696     \ifx\@tempa\@tempb
697         \@l@dttempcntb \z@
698     \else
699         \def\@tempb{right}%
700         \ifx\@tempa\@tempb
701             \@l@dttempcntb \@ne

```

```

702 \else
703 \def\@tempb{outer}%
704 \ifx\@tempa\@tempb
705 \l@dttempcntb \tw@
706 \else
707 \def\@tempb{inner}%
708 \ifx\@tempa\@tempb
709 \l@dttempcntb \thr@@
710 \else
711 \led@warn@BadLinenummargin
712 \l@dttempcntb \m@ne
713 \fi
714 \fi
715 \fi
716 \fi}
717
718 %

```

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.

```

719 \newcounter{firstlinenum}
720 \setcounter{firstlinenum}{5}
721 \newcounter{linenumincrement}
722 \setcounter{linenumincrement}{5}
723 %

```

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

```

724 \newcounter{firstsublinenum}
725 \setcounter{firstsublinenum}{5}
726 \newcounter{sublinenumincrement}
727 \setcounter{sublinenumincrement}{5}
728
729 %

```

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

```

730 \newcommand*{\firstlinenum}[1]{%
731 \ifledRcol%
732 \setcounter{firstlinenumR}{#1}%
733 \led@warn@setting@in@rightside{\firstlinenum}%
734

```

```

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

```

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, `\l@dgetlock@disp` it is set to `first`.

`\lock@disp` encodes the selection: 0 for first, 1 for last, 2 for all.

```

765 \newcount\lock@disp
766 \newcommand*\lockdisp}[1]{%
767 \l@dgetlock@disp{#1}%
768 \ifnum\@l@tempcntb>\m@ne
769 \global\lock@disp=\@l@tempcntb
770 \else
771 \led@warn@BadLockdisp
772 \fi}}
773 \newcommand*\l@dgetlock@disp}[1]{
774 \def\@tempa{#1}\def\@tempb{first}%

```

```

775 \ifx\@tempa\@tempb
776 \l@dttempcntb \z@
777 \else
778 \def\@tempb{last}%
779 \ifx\@tempa\@tempb
780 \l@dttempcntb \@ne
781 \else
782 \def\@tempb{all}%
783 \ifx\@tempa\@tempb
784 \l@dttempcntb \tw@
785 \else
786 \l@dttempcntb \m@ne
787 \fi
788 \fi
789 \fi}
790
791 %

```

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

```

792 \newcount\sublock@disp
793 \newcommand{\sublockdisp}[1]{\{%
794 \l@dgetlock@disp{#1}%
795 \ifnum\l@dttempcntb>\m@ne
796 \global\sublock@disp=\l@dttempcntb
797 \else
798 \led@warn@BadSublockdisp
799 \fi}}
800
801 %

```

V.5 Line number style

`\linenumberstyle` We provide a mechanism for using different representations of the line numbers, not just the normal arabic.

`\linenumrep`

`\linenumr@p`

`\sublinenumberstyle`

`\sublinenumrep`

`\sublinenumr@p`

NOTE: In v0.7 `\linenumrep` and `\sublinenumrep` replaced the internal `\linenumr@p` and `\sublinenumr@p`.

`\linenumberstyle` and `\sublinenumberstyle` are user level macros for setting the number representation (`\linenumrep` and `\sublinenumrep`) for line and sub-line numbers.

```

802 \newcommand*{\linenumberstyle}[1]{\%
803 \def\linenumrep##1{\@nameuse{@#1}{##1}}}
804 \newcommand*{\sublinenumberstyle}[1]{\%
805 \def\sublinenumrep##1{\@nameuse{@#1}{##1}}}
806 %

```

Initialise the number styles to arabic.

```

807 \linenumberstyle{arabic}
808 \let\linenumr@p\linenumrep
809 \sublinenumberstyle{arabic}
810 \let\sublinenumr@p\sublinenumrep
811
812 %

```

V.6 Line number printing

`\leftlinenum` `\rightlinenum` 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 subline) number.

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

```

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

```

```

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

```

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

```

857 \newcommand*{\leftlinenumannotation}{%
858 \l@wrap@ifnotemptybox{\Xwraplinenumannotation@side}{%
859 \csuse{annot@the\absline@num @the\section@num}%
860 }%
861 \kern\linenumsep%
862 }%
863 \newcommand*{\rightlinenumannotation}{%
864 \kern\linenumsep%
865 \l@wrap@ifnotemptybox{\Xwraplinenumannotation@side}{%
866 \csuse{annot@the\absline@num @the\section@num}%
867 }%
868 }%
869 %

```

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`.

```
870 \newcount\line@num
871 %
```

`\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.

```
872 \newcount\subline@num
873 %
```

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

```
874 \newif\ifsublines@
875 \newif\ifsublines@R%
876 %
```

`\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.

```
877 \newcount\absline@num
878 %
```

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.

```
879 \newcount\@lock
880 \newcount\sub@lock
881 %
```

V.9 Line number associated to lemma

`\line@list` Now we can define the list of macros which will be created from the line-list file. We 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/cm/r/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:

```
882 \list@create{\line@list}
883 \list@create{\insertlines@list}
884 \list@create{\actionlines@list}
885 \list@create{\actions@list}
886 \list@create{\annot@list}%
887 %
```

```
\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
888 \newcount\page@num
889 \newcount\endpage@num
890 \newcount\endline@num
891 \newcount\endsubline@num
892 %
```

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

```
893 \newcount\this@section@page@num%
894 %
```

```
\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  $\TeX$ , 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  $\TeX$  two more times; there is no fix needed to the document.
The \ifnoteschanged@ flag is set if such a change in the number of notes is discovered
at any point.
```

```
895 \newif\ifnoteschanged@
896 %
```

`\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@ \newcommand*{\resetprevline@}{%
898   \def\do##1{\global\csundef{prevline##1}}%
899   \dolistloop{\@series}%
900 }
901 %

```

`\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@ \newcommand*{\resetprevpage@num}{%
902   \def\do##1{%
903     \ifcsdef{prevpage##1@num}{%
904       \global\csname prevpage##1@num\endcsname=\z@%
905       \global\csname prevpage##1@numR\endcsname=\z@%
906     }%
907     {}%
908   }%
909   \ifcsdef{##1prevpage@num}{%
910     \global\csname ##1prevpage@num\endcsname=\z@%
911     \global\csname ##1prevpage@numR\endcsname=\z@%
912   }%
913   {}%
914 }%
915 \dolistloop{\@series}%
916 }
917 %

```

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.

```

918 \newread\@inputcheck
919 \newcommand*{\read@linelist}[1]{%
920   \ifledRcol%
921     \list@clearing@regR%
922   \else%
923     \list@clearing@reg%
924   \fi%
925 %

```

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).

```
926 \list@clear{\maxlinesinpar@list}
927 %
```

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 `\resumenumbering`, 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.

```
928 \get@linelistfile{#1}%
929 \@stopmsd%Security if last \endms{} is forgotten
930 \unless\ifledRcol%Get the last line of the last page
931   \csnumgdef{@lastabsline@forpage@the\page@num}{\the\absline@num}%
932   \csnumgdef{@lastline@forpage@the\page@num}{\the\line@num}%
933   \csnumgdef{@lastsubline@forpage@the\page@num}{\the\subline@num}%
934 \else%
935   \csnumgdef{@lastabsline@forpageR@the\page@numR}{\the\absline@numR}%
936   \csnumgdef{@lastline@forpageR@the\page@numR}{\the\line@numR}%
937   \csnumgdef{@lastsubline@forpageR@the\page@numR}{\the\subline@numR}%
938 \fi%
939 \endgroup
940 %
```

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.

```
941 \ifledRcol
942   \global\page@numR=\m@ne
943   \ifx\actionlines@listR\empty
944     \gdef\next@actionlineR{1000000}%
945   \else
946     \glp\actionlines@listR\to\next@actionlineR
947     \glp\actions@listR\to\next@actionR
948   \fi
949 \else
```

```

950 \global\page@num=\m@ne
951 \ifx\actionlines@list\empty
952   \gdef\next@actionline{1000000}%
953 \else
954   \gl@p\actionlines@list\to\next@actionline
955   \gl@p\actions@list\to\next@action
956 \fi
957 \fi
958 }
959 %

```

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

```

960 \newcommand*\list@clearing@reg{%
961   \list@clear{\line@list}%
962   \list@clear{\insertlines@list}%
963   \list@clear{\actionlines@list}%
964   \list@clear{\actions@list}%
965   \list@clear{\linesinpar@listL}%
966   \list@clear{\linesonpage@listL}%
967 }%
968 %

```

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

```

969 \newcommand*\get@linelistfile}[1]{%
970   \InputIfFileExists{\l@auxdir#1}{%
971     \global\noteschanged@false
972     \begingroup
973       \catcode`\[=1 \catcode`\]=2
974       \makeatletter \catcode`\~M=9}{%
975     \led@warn@NoFile{\l@auxdir#1}%
976     \global\noteschanged@true
977     \begingroup}%
978 }
979
980 %

```

This version of `\read@linelist` creates list macros containing data for the entire section, so they could get rather large. It would be no more difficult to read the line-list file incrementally rather than all at once: we could read, at the start of each paragraph, only the commands relating to that paragraph. But this would require that we have two line-lists open at once, one for reading, one for writing, and on systems without version numbers we would have to do some file renaming outside of \TeX for that to work. We have retained this slower approach to avoid that sort of hacking about, but have provided the `\pausenumbering` and `\resumenumbering` macros to help you if you run into macro memory limitations (see 5.2.7 p. 21 above).

V.11 Commands within the line-list file

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

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

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

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

```

981 \newcommand{\line@list@version}[1]{%
982   \IfStrEq{#1}{\this@line@list@version}%
983   {}%
984   {\ifledRcol%
985     \led@warn@Obsolete{\jobname.\extensionchars\the\section@num}%
986     \else%
987     \led@warn@Obsolete{\jobname.\extensionchars\the\section@num}%
988     \fi%
989     \endinput%
990   }%
991 }%
992 %

```

`\@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`.

```

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

```

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

```

1012 \reset@current@annot%
1013 \advance\absline@num \@ne
1014   \ifx\next@page@num\relax \else
1015     \page@action
1016     \let\next@page@num=\relax
1017   \fi
1018   \ifx\sub@change\relax \else
1019     \ifnum\sub@change>\z@
1020       \sublines@true
1021     \else
1022       \sublines@false
1023     \fi
1024     \sub@action
1025     \let\sub@change=\relax
1026   \fi
1027 }%

```

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

```

1028   \ifcase\@lock
1029     \or
1030       \@lock \tw@
1031     \or \or
1032       \@lock \z@
1033     \fi
1034   \ifcase\sub@lock
1035     \or

```



```

1036         \sub@lock \tw@
1037         \or \or
1038         \sub@lock \z@
1039     \fi
1040 %

```

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

```

1041     \ifsublines@
1042         \ifnum\sub@lock<\tw@
1043             \advance\subline@num \@ne
1044         \fi
1045     \else
1046         \ifnum\@lock<\tw@
1047             \advance\line@num \@ne \subline@num \z@
1048         \fi
1049     \fi}
1050 %
1051 %

```

`\last@page@num` `\@page` determines whether or not a new page has been started, based on the page values held by `\@nl`.

```

\this@section@last@page@num
\@page
1052 \newcount\last@page@num
1053 \last@page@num=-10000
1054
1055 \newcount\this@section@last@page@num%
1056 \this@section@last@page@num=-10000%
1057
1058 \newcommand*{\@page}[1]{%
1059     \ifledRcol
1060         \ifnum #1=\last@page@numR
1061         \else
1062             \csnumgdef{@lastabsline@forpageR@the\page@numR}{\the\absline@numR}%
1063             \csnumgdef{@lastline@forpageR@the\page@numR}{\the\line@numR}%
1064             \csnumgdef{@lastsubline@forpageR@the\page@numR}{\the\subline@numR}%
1065             \ifbypage@R
1066                 \ifx\ldchset@num\relax%Not resetting if preceded by a \setlinenum
1067                     \ifboolexpr{%
1068                         bool{resumenumberingR@start}%
1069                         and test {\ifnumequal{\last@page@numR}{-10000}}%
1070                     }%
1071                 {}%
1072                 {%
1073                     \line@numR \z@%
1074                     \subline@numR \z@%
1075                     \global\csdef{reset@lineR\the\numexpr\absline@numR+1\relax @\
the\section@numR}{}%
1076                 }%
1077             \fi%
1078         \fi

```

```

1079 \page@numR=#1%
1080 \global\this@section@page@numR=#1%
1081 \last@page@numR=#1%
1082 \global\this@section@last@page@numR=#1%
1083 \def\next@page@numR{#1}%
1084 \gdef\this@section@next@page@numR{#1}%
1085 \fi
1086 \else
1087 \ifnum #1=\last@page@num
1088 \else
1089 \csnumgdef{lastabsline@forpage@the\page@num}{the\absline@num}%
1090 \csnumgdef{lastline@forpage@the\page@num}{the\line@num}%
1091 \csnumgdef{lastsubline@forpage@the\page@num}{the\subline@num}%
1092 \ifbypage@
1093 \ifx\l@dcchset@num\relax%Not resetting if preceded by a \setlinenum
1094 \line@num \z@%
1095 \subline@num \z@%
1096 \global\csdef{reset@line\the\numexpr\absline@num+1\relax @the\
section@num}{}%
1097 \fi%
1098 \fi
1099 \page@num=#1%
1100 \global\this@section@page@num=#1%
1101 \last@page@num=#1%
1102 \global\this@section@last@page@num=#1%
1103 \def\next@page@num{#1}%
1104 %

```

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

```

1105 \gdef\this@section@next@page@num{#1}%
1106 \listxadd{normal@page@break}{the\absline@num}
1107 \fi
1108 \fi}
1109 %

```

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

```

1110 \newcommand*{\@pend}[1]{}
1111 \newcommand*{\@pendR}[1]{}
1112 \newcommand*{\@lopL}[1]{}
1113 \newcommand*{\@lopR}[1]{}
1114
1115 %

```

\sub@on The \sub@on and \sub@off macros turn sub-lineation on and off: but not directly, since such changes do not really take effect until the next line of text. Instead they set a flag that notifies \@n1 of the necessary action.

```

1116 \newcommand*{\sub@on}{%
1117   \ifboolexpr{%
1118     bool{sublines@}%
1119     or%
1120     (bool{sublines@R} and bool{ledRcol})%
1121   }{%
1122     \let\sub@change=\relax
1123   }{%
1124     \def\sub@change{1}%
1125   }%
1126 }%
1127 \newcommand*{\sub@off}{%
1128   \ifboolexpr{%
1129     bool{sublines@}%
1130     or%
1131     (bool{sublines@R} and bool{ledRcol})%
1132   }{%
1133     \def\sub@change{-1}%
1134   }{%
1135     \let\sub@change=\relax
1136   }%
1137 }%
1138
1139 %

```

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

```

1140
1141 \newcommand*{\@adv}[1]{%
1142   \ifsublines@
1143     \ifledRcol
1144       \advance\subline@numR by #1\relax
1145       \ifnum\subline@numR<\z@
1146         \led@warn@BadAdvancelineSubline
1147         \subline@numR \z@
1148       \fi
1149     \else
1150       \advance\subline@num by #1\relax
1151       \ifnum\subline@num<\z@
1152         \led@warn@BadAdvancelineSubline
1153         \subline@num \z@
1154       \fi
1155     \fi
1156   \else
1157     \ifledRcol
1158       \advance\line@numR by #1\relax
1159       \ifnum\line@numR<\z@
1160         \led@warn@BadAdvancelineLine
1161         \line@numR \z@

```

```

1162     \fi
1163   \else
1164     \advance\line@num by #1\relax
1165     \ifnum\line@num<\z@
1166       \led@warn@BadAdvancelineLine
1167       \line@num \z@
1168     \fi
1169   \fi
1170 \fi
1171 \set@line@action}
1172
1173 %

```

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

```

1174
1175 \newcommand*\@set}[1]{%
1176   \ifledRcol
1177     \ifsublines@R%
1178       \subline@numR=#1\relax
1179     \else
1180       \line@numR=#1\relax
1181     \fi
1182     \set@line@action
1183   \else
1184     \ifsublines@
1185       \subline@num=#1\relax
1186     \else
1187       \line@num=#1\relax
1188     \fi
1189     \set@line@action
1190   \fi}
1191
1192 %

```

\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.

```

1193
1194 \newcommand*\l@d@set}[1]{%
1195   \ifledRcol
1196     \line@numR=#1\relax
1197     \advance\line@numR \@ne
1198     \def\l@dchset@num{#1}
1199   \else
1200     \line@num=#1\relax
1201     \advance\line@num \@ne

```

```

1202 \def\l@dchset@num{#1}
1203 \fi}
1204 \let\l@dchset@num\relax
1205
1206 %

```

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

```

1207
1208 \newcommand*{\page@action}{%
1209 \ifledRcol
1210 \xright@appenditem{\the\absline@numR}\to\actionlines@listR
1211 \xright@appenditem{\next@page@numR}\to\actions@listR
1212 \else
1213 \xright@appenditem{\the\absline@num}\to\actionlines@list
1214 \xright@appenditem{\next@page@num}\to\actions@list
1215 \fi}
1216 %

```

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

```

1217
1218 \newcommand*{\set@line@action}{%
1219 \ifledRcol
1220 \xright@appenditem{\the\absline@numR}\to\actionlines@listR
1221 \ifsublines@R%
1222 \l@dtempcnta=-\subline@numR
1223 \else
1224 \l@dtempcnta=-\line@numR
1225 \fi
1226 \advance\l@dtempcnta by -5000\relax
1227 \xright@appenditem{\the\l@dtempcnta}\to\actions@listR
1228 \else
1229 \xright@appenditem{\the\absline@num}\to\actionlines@list
1230 \ifsublines@
1231 \l@dtempcnta=-\subline@num
1232 \else
1233 \l@dtempcnta=-\line@num
1234 \fi
1235 \advance\l@dtempcnta by -5000\relax
1236 \xright@appenditem{\the\l@dtempcnta}\to\actions@list
1237 \fi}
1238 %

```

\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.

```

1239
1240 \newcommand*{\sub@action}{%

```

```

1241 \ifledRcol
1242   \xright@appenditem{\the\absline@numR}\to\actionlines@listR
1243   \ifsublines@R%
1244     \xright@appenditem{-1001}\to\actions@listR
1245   \else
1246     \xright@appenditem{-1002}\to\actions@listR
1247   \fi
1248 \else
1249   \xright@appenditem{\the\absline@num}\to\actionlines@list
1250   \ifsublines@
1251     \xright@appenditem{-1001}\to\actions@list
1252   \else
1253     \xright@appenditem{-1002}\to\actions@list
1254   \fi
1255 \fi}
1256 %

```

`\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.

```

1257 \newcommand*{\lock@on}{\futurelet\next\do@lockon}
1258
1259 \newcommand*{\do@lockon}{%
1260   \ifx\next\lock@off
1261     \global\let\lock@off=\skip@lockoff
1262   \else
1263     \ifledRcol
1264       \do@lockonR
1265     \else
1266       \do@lockonL
1267     \fi
1268   \fi}
1269
1270
1271 \newcommand*{\do@lockonL}{%
1272   \xright@appenditem{\the\absline@num}\to\actionlines@list
1273   \ifsublines@
1274     \xright@appenditem{-1005}\to\actions@list
1275     \ifnum\sub@lock=\z@
1276       \sub@lock \@ne
1277     \else
1278       \ifnum\sub@lock=\thr@@
1279         \sub@lock \@ne
1280       \fi

```

```

1281 \fi
1282 \else
1283 \xright@appenditem{-1003}\to\actions@list
1284 \ifnum\@lock=\z@
1285 \@lock \@ne
1286 \else
1287 \ifnum\@lock=\thr@@
1288 \@lock \@ne
1289 \fi
1290 \fi
1291 \fi}
1292
1293 %

```

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

```

\do@lockoff
\do@lockoffL
\skip@lockoff
1294 \newcommand*{\do@lockoffL}{%
1295 \xright@appenditem{\the\absline@num}\to\actionlines@list
1296 \ifsublines@
1297 \xright@appenditem{-1006}\to\actions@list
1298 \ifnum\sub@lock=\tw@
1299 \sub@lock \thr@@
1300 \else
1301 \sub@lock \z@
1302 \fi
1303 \else
1304 \xright@appenditem{-1004}\to\actions@list
1305 \ifnum\@lock=\tw@
1306 \@lock \thr@@
1307 \else
1308 \@lock \z@
1309 \fi
1310 \fi}
1311
1312 \newcommand*{\do@lockoff}{%
1313 \reset@current@annot%
1314 \ifledRcol
1315 \do@lockoffR
1316 \else
1317 \do@lockoffL
1318 \fi}
1319 \newcommand*{\skip@lockoff}{\global\let\lock@off=\do@lockoff}
1320 \global\let\lock@off=\do@lockoff
1321
1322 %

```

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

```

1323 \newcommand*{\n@num}{%

```

```

1324 \ifledRcol%
1325 \xright@appenditem{\the\absline@numR}\to\actionlines@listR
1326 \xright@appenditem{-1007}\to\actions@listR
1327 \else%
1328 \xright@appenditem{\the\absline@num}\to\actionlines@list%
1329 \xright@appenditem{-1007}\to\actions@list%
1330 \fi%
1331 }%
1332 %
1333 %

```

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

```

1334 \newcommand*{\n@num@stanza}{%
1335 \ifledRcol%
1336 \xright@appenditem{\the\absline@numR}\to\actionlines@listR%
1337 \xright@appenditem{-1008}\to\actions@listR%
1338 \else%
1339 \xright@appenditem{\the\absline@num}\to\actionlines@list%%
1340 \xright@appenditem{-1008}\to\actions@list%
1341 \fi%
1342 }
1343 %

```

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

```

1344 \newif\ifl@dhidenumber
1345 \newcommand*{\hidenumbering}{%
1346 \ifledRcol%
1347 \write\linenum@outR{\string\hide@num}%
1348 \else%
1349 \write\linenum@out{\string\hide@num}%
1350 \fi%
1351 }%
1352 \newcommand*{\hide@num}{%
1353 \ifledRcol%
1354 \xright@appenditem{\the\absline@numR}\to\actionlines@listR%
1355 \xright@appenditem{-1009}\to\actions@listR%
1356 \else%
1357 \xright@appenditem{\the\absline@num}\to\actionlines@list%%
1358 \xright@appenditem{-1009}\to\actions@list%
1359 \fi%
1360 }
1361 \newcommand*{\hidenumberingonleftpage}{%
1362 \ifledRcol%
1363 \write\linenum@outR{\string\hide@num@left}%
1364 \else%

```



```

1365 \write\linenum@out{\string\hide@num@left}%
1366 \fi%
1367 }%
1368
1369 \newcommand*{\hide@num@left}{%
1370 \ifledRcol%
1371 \ifodd\page@numR\else%
1372 \xright@appenditem{\the\absline@numR}\to\actionlines@listR%
1373 \xright@appenditem{-1009}\to\actions@listR%
1374 \fi%
1375 \else%
1376 \ifodd\page@num\else%
1377 \xright@appenditem{\the\absline@num}\to\actionlines@list%%
1378 \xright@appenditem{-1009}\to\actions@list%
1379 \fi%
1380 \fi%
1381 }%
1382
1383 \newcommand*{\hidenumberingonrightpage}{%
1384 \ifledRcol%
1385 \write\linenum@outR{\string\hide@num@right}%
1386 \else%
1387 \write\linenum@out{\string\hide@num@right}%
1388 \fi%
1389 }%
1390
1391 \newcommand*{\hide@num@right}{%
1392 \ifledRcol%
1393 \ifodd\page@numR%
1394 \xright@appenditem{\the\absline@numR}\to\actionlines@listR%
1395 \xright@appenditem{-1009}\to\actions@listR%
1396 \fi%
1397 \else%
1398 \ifodd\page@num%
1399 \xright@appenditem{\the\absline@num}\to\actionlines@list%%
1400 \xright@appenditem{-1009}\to\actions@list%
1401 \fi%
1402 \fi%
1403 }%
1404
1405 %

```

\@ref \@ref marks the start of a passage, for creation of a footnote reference. It takes two arguments:

- #1, the number of entries to add to \insertlines@list for this reference. This value, here and within \edtext, which computes it and writes it to the line-list file, will be stored in the count \insert@count.

```

1406 \newcount\insert@count

```

```
1407 %
```

- #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.

```
1408 \newcommand*\@dummy@ref}[2]{#2}
1409 %
```

\@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.

```
1410 \newcommand*\@ref}[2]{%
1411   \ifledRcol%
1412     \@ref@regR{#1}{#2}%
1413   \else%
1414     \@ref@reg{#1}{#2}%
1415   \fi%
1416 }%
1417 \newcommand*\@ref@reg}[2]{%
1418   \global\insert@count=#1\relax
1419   \global\advance\@edtext@level by 1%
1420   \loop\ifnum\insert@count>\z@
1421     \xright@appenditem{\the\absline@num}\to\insertlines@list
1422     \global\advance\insert@count \m@ne
1423   \repeat
1424 %
```

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.

```
1425 \begingroup
1426   \let\@ref=\dummy@ref
1427   \let\@lopL\@gobble
1428   \let\page@action=\relax
1429   \let\sub@action=\relax
1430   \let\set@line@action=\relax
1431   \let\@lab=\relax
1432   \let\@lemma=\relax%
1433   \let\@sw\@gobblethree%
1434   \let\store@annot@to@absline\@gobble%
1435   #2
1436   \global\endpage@num=\page@num
1437   \global\endline@num=\line@num
1438   \global\endsubline@num=\subline@num
```

```

1439 \global\let\endcurrent@annot=\current@annot%
1440 \endgroup
1441 %

```

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

```

1442 \xright@appenditem%
1443   {\the\page@num|\the\line@num|%
1444    \ifsublines@ \the\subline@num \else 0\fi}%
1445   \the\endpage@num|\the\endline@num|%
1446   \ifsublines@ \the\endsubline@num \else 0\fi}\to\line@list
1447 \xright@appenditem%
1448   {\current@annot|\endcurrent@annot}\to\annot@list%
1449 %

```

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

```

1450   \@ref@reg@parse{#2}%
1451 %

```

Decrease edtext level counter.

```

1452 \global\advance\@edtext@level by -1%
1453 }
1454 %

```

`\@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.

```

1455 \newcommand{\@ref@reg@parse}[1]{%
1456   \expandafter\list@create\expandafter{\csname sw@list@edtext@tmp@\the\
1457   \@edtext@level\endcsname}%
1458   \providebool{lemmacommand@\the\@edtext@level}%
1459   \boolfalse{lemmacommand@\the\@edtext@level}%
1460 %

```

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

```

1460   #1%
1461 %

```

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.

```

1462 \ifnum\@edtext@level>0%
1463   \def\create@this@edtext@level{\expandafter\list@create\expandafter{\
1464   \csname sw@list@edtext@\the\@edtext@level\endcsname}}%
1465   \ifcsundef{sw@list@edtext@\the\@edtext@level}{\create@this@edtext@level}%
1466   \letcs{\@tmp}{sw@list@edtext@\the\@edtext@level}%

```

```

1466 \letcs{\@tmpp}{sw@list@edtext@tmp@the\@edtext@level}
1467 \xright@appenditem{\expandonce\@tmpp}{to\@tmp%
1468 \global\cslet{sw@list@edtext@the\@edtext@level}{\@tmp}%
1469 \fi%
1470 %
1471 }
1472 %
1473 %

```

\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.

```

1474 \newcommand{\@ref@later}[1]{%
1475 \global\advance\@edtext@level by \@ne%
1476 \ifledRcol%
1477 \@ref@reg@parseR{#1}%
1478 \else%
1479 \@ref@reg@parse{#1}%
1480 \fi%
1481 \global\advance\@edtext@level by -\@ne%
1482 }%
1483 %

```

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`.

```

1484 \newwrite\linenum@out
1485 %

```

\iffirst@linenum@out@ Once any file is opened on this stream, we keep it open forever, or else switch to another file that we keep open. The reason is that we want the output routine to write the page number for every page to this file; otherwise we would have to write it at the start of every line. But it is not very easy for the output routine to tell whether an output stream is open or not. There is no way to test the status of a particular output stream directly, and the asynchronous nature of output routines makes the status hard to determine by other means.

\first@linenum@out@true
\first@linenum@out@false

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`.

```

1486 \newif\iffirst@linenum@out@
1487 \first@linenum@out@true
1488 %

```

\this@line@list@version The commands allowed in the line-list file and their arguments can change between two version of reledmac. The `\this@line@list@version` command is upgraded when it happens. It is written in the file list. If we process a line-list file which used a older version, that means the commands used inside are deprecated, and we can't use them.

```

1489 \newcommand{\this@line@list@version}{8}%
1490 %

```

\line@list@stuff The `\line@list@stuff{<file>}` macro, which is called by `\beginnumbering`, performs all the line-list operations needed at the start of a section. Its argument is the name of the line-list file.

\next@line@list@stuff

```

1491 \let\next@line@list@stuff\relax%
1492 \newcommand*{\line@list@stuff}[1]{%
1493 %

```

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

```

1494 \global\newtoggle{notfirststrun@#1}%
1495 \IfFileExists{\l@auxdir#1}%
1496 {\global\toggletrue{notfirststrun@#1}}%
1497 {\global\togglefalse{notfirststrun@#1}}%
1498 %

```

A internal hook (not used yet).

```

1499 \next@line@list@stuff%
1500 \global\let\next@line@list@stuff\relax%
1501 %

```

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

```

1502 \read@linelist{#1}%
1503 %

```

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 .

```

1504 \iffirst@linenum@out@
1505 \global\first@linenum@out@false%
1506 \immediate\openout\linenum@out=\l@auxdir#1\relax%
1507 \immediate\write\linenum@out{\string\line@list@version{\
this@line@list@version}}%
1508 \ifl@dpaging%

```

```

1509 \immediate\write\linenum@out{\string\@par@sync@option{\
@par@this@sync@option}}}%
1510 \fi%
1511 \else
1512 %

```

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.

```

1513 \if@minipage%
1514 \leavevmode%
1515 \fi%
1516 \closeout\linenum@out%
1517 \openout\linenum@out=\l@auxdir#1\relax%
1518 \write\linenum@out{\string\line@list@version{\this@line@list@version}}%
1519 \ifl@dpaging%
1520 \write\linenum@out{\string\@par@sync@option{\@par@this@sync@option}}%
1521 \fi%
1522 \fi%
1523 }%
1524 %
1525 %

```

`\new@line` The `\new@line` macro sends the `\@nl` command to the line-list file, to mark the start of a new text line, and its page number. It writes the two forms of the page number:

- Raw form (`\the\c@page`, which can be used for numeric tests).
- Formatted form (for example, in Roman).

Actually, only the first form is used by `\@nl`. If we use the `\sameparallelpagenumber` option of `reledpar`, we must write not the real page number (i.e. page counter, defined in standard \LaTeX) 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.

```

1526 \newcommand*{\new@line}{%
1527 \ifnumberline%
1528 \IfStrEq{\led@pb@setting}{after}%
1529 {\xifinlist{\the\absline@num}{\l@prev@nopb}%
1530 {\xifinlist{\the\absline@num}{\normal@page@break}%
1531 {\numgdef{\@next@page}{\c@par@page+\@ne}%
1532 \write\linenum@out{\string\@nl[\@next@page][\@next@page]}}%

```

```

1533     }%
1534     {\write\linenum@out{\string\@nl[\the\@this@c@page][\thepage]}}%
1535   }%
1536   {\write\linenum@out{\string\@nl[\the\@this@c@page][\thepage]}}}%
1537 }%
1538 \IfStrEq{\led@pb@setting}{before}%
1539   {\numdef{\next@absline}{\the\absline@num+\@ne}%
1540   \xifinlist{\next@absline}{\l@prev@nopb}%
1541     {\xifinlist{\the\absline@num}{\normal@page@break}%
1542       {\numgdef{\nc@page}{\c@par@page+\@ne}%
1543       \write\linenum@out{\string\@nl[\nc@page][\nc@page]}}%
1544     }%
1545     {\write\linenum@out{\string\@nl[\the\@this@c@page][\thepage]}}}%
1546   }%
1547   {\write\linenum@out{\string\@nl[\the\@this@c@page][\thepage]}}}%
1548 }%
1549 {}%
1550 \IfStrEqCase{\led@pb@setting}%
1551   {%
1552     {before}{\relax}%
1553     {after}{\relax}%
1554   }%
1555   \write\linenum@out{\string\@nl[\the\@this@c@page][\thepage]}}%
1556 ]%
1557 \fi%
1558 }
1559
1560 \newcount\@this@c@page%%
1561 \newcommand{\set@this@c@page}{%
1562   \ifboolexpr{%
1563     bool{sameparallelpagenumber}%
1564     or bool{prevpgnotnumbered}%
1565   }%
1566   {\global\@this@c@page=\c@par@page}%
1567   {\global\@this@c@page=\c@page}%
1568 }%
1569 %

```

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

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

\flag@end

```

1570 \newif\if@noneed@Footnote%
1571
1572 \newcommand*{\flag@start}{%
1573   \ifledRcol%
1574     \edef\next{\write\linenum@outR{%

```

```

1575 \string\@ref[\the\insert@countR] []}%
1576 \next%
1577 \ifnum\insert@countR<1%
1578 \if@noneed@Footnote\else%
1579 \led@err@EdtextWithoutFootnote%
1580 \fi%
1581 \fi%
1582 \else%
1583 \edef\next{\write\linenum@out{%
1584 \string\@ref[\the\insert@count] []}%
1585 \next%
1586 \ifnum\insert@count<1%
1587 \if@noneed@Footnote\else%
1588 \led@err@EdtextWithoutFootnote%
1589 \fi%
1590 \fi%
1591 \fi}%
1592
1593 \newcommand*{\flag@end}{%
1594 \ifledRcol%
1595 \write\linenum@outR[]}%
1596 \else%
1597 \write\linenum@out[]}%
1598 \fi}%
1599
1600 %
1601 %

```

`\flag@start@RTL`
`\flag@end@RTL`

With \LaTeX , there is a problem when using RTL: the writing of a command in the numbered auxiliary files (.1, .2 etc) is reversed when the first argument of `\edtext` is typeset in one line, but it is **not** reversed when this first argument is typeset in two lines or more.³⁰

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

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

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

³⁰This problem is caused by the way \LaTeX manages right-to-left typesetting. David Carlisle explains it on <http://tex.stackexchange.com/a/333373/7712> and provides a potential solution, using `\vadjust`. However in some cases this adds spurious vertical spaces in reledmac. That is why we are using the solution explained below.

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

```

1602 \newcommand{\flag@start@RTL}{%
1603   \edlabel{edtext:start:\csuse{thisedtext@the\@edtext@level}}%
1604   \IfStrEq{\xabslineref{edtext:start:\csuse{thisedtext@the\@edtext@level
1605     }}}%
1606     {}%
1607     {%
1608       \ifnumequal%
1609         {\xabslineref{edtext:start:\csuse{thisedtext@the\@edtext@level}}}%
1610         {\xabslineref{edtext:end:\csuse{thisedtext@the\@edtext@level}}}%
1611         {\flag@end}%
1612         {\flag@start}%
1613       }%
1614     }%
1615
1616 \newcommand{\flag@end@RTL}{%
1617   \edlabel{edtext:end:\csuse{thisedtext@the\@edtext@level}}%
1618   \IfStrEq{\xabslineref{edtext:start:\csuse{thisedtext@the\@edtext@level
1619     }}}%
1620     {}%
1621     {%
1622       \ifnumequal%
1623         {\xabslineref{edtext:start:\csuse{thisedtext@the\@edtext@level}}}%
1624         {\xabslineref{edtext:end:\csuse{thisedtext@the\@edtext@level}}}%
1625         {\flag@start}%
1626         {\flag@end}%
1627       }%
1628     }%
1629   %

```

`\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`

```

1630 \newcommand*{\flag@start@later}{%
1631   \ifledRcol%
1632     \write\linenum@outR{\string\@ref@later[]}%
1633   \else%
1634     \write\linenum@out{\string\@ref@later[]}%
1635   \fi%
1636 }%
1637 \newcommand{\flag@end@later}{%
1638   \ifledRcol%
1639     \write\linenum@outR{[]}%
1640   \else%
1641     \write\linenum@out{[]}%
1642   \fi%

```

```
1643 }
1644 %
```

`\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.

```
1645
1646
1647 \newcommand*{\startsub}{%
1648   \dimen0\lastskip%
1649   \ifdim\dimen0>0pt%
1650     \unskip%
1651     \fi%
1652     \ifledRcol%
1653       \write\linenum@outR{\string\sub@on}%
1654     \else%
1655       \write\linenum@out{\string\sub@on}%
1656     \fi%
1657     \ifdim\dimen0>0pt%
1658       \hskip\dimen0%
1659     \fi%
1660   }%
1661
1662 \def\endsub{%
1663   \dimen0\lastskip%
1664   \ifdim\dimen0>0pt%
1665     \unskip%
1666     \fi%
1667     \ifledRcol%
1668       \write\linenum@outR{\string\sub@off}%
1669     \else%
1670       \write\linenum@out{\string\sub@off}%
1671     \fi%
1672     \ifdim\dimen0>0pt%
1673       \hskip\dimen0%
1674     \fi%
1675   }%
1676
1677 %
```

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

```

1678 \newcommand*{\advanceline}[1]{\leavevmode%
1679 \ifledRcol \write\linenum@outR{\string\@adv[#1]}%
1680 \else      \write\linenum@out{\string\@adv[#1]}%
1681 \fi%
1682 }
1683 %

```

\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.

```

1684 \newcommand*{\setline}[1]{%
1685 \leavevmode%
1686 \ifnum#1<\z@
1687 \led@warn@BadSetline%
1688 \else%
1689 \ifledRcol%
1690 \write\linenum@outR{\string\@set[#1]}%
1691 \else%
1692 \write\linenum@out{\string\@set[#1]}%
1693 \fi%
1694 \fi%
1695 }%
1696 %
1697
1698 %

```

\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.

```

1699 \newcommand*{\setlinenum}[1]{%
1700 \ifnum#1<\z@%
1701 \led@warn@BadSetlinenum%
1702 \else%
1703 \ifledRcol%
1704 \write\linenum@outR{\string\l@d@set[#1]}%
1705 \else%
1706 \write\linenum@out{\string\l@d@set[#1]}%
1707 \fi%
1708 \fi%
1709 }%
1710 %
1711
1712 %

```

\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.

```

1713
1714 \newcommand*{\startlock}{%
1715   \ifledRcol \write\linenum@outR{\string\lock@on}%
1716   \else      \write\linenum@out{\string\lock@on}%
1717   \fi}
1718 \def\endlock{%
1719   \ifledRcol \write\linenum@outR{\string\lock@off}%
1720   \else      \write\linenum@out{\string\lock@off}%
1721   \fi}
1722 %

```

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

```

1723 \l@dskipnumbertrue \newif\ifl@dskipnumber
1724 \l@dskipnumberfalse \newif\ifl@dskipversenumber%
1725 \skipnumbering \newcommand*{\skipnumbering}{%
1726   \leavevmode%
1727   \ifledRcol%
1728     \ifinstanza%
1729       \write\linenum@outR{\string\n@num@stanza}%
1730     \else%
1731       \write\linenum@outR{\string\n@num}%
1732     \fi%
1733     \advanceline{-1}%
1734   \else%
1735     \ifinstanza%
1736       \write\linenum@out{\string\n@num@stanza}%
1737     \else%
1738       \write\linenum@out{\string\n@num}%
1739     \fi%
1740     \advanceline{-1}%
1741   \fi%
1742 }%
1743
1744 %

```

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. 165). 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 can not 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.

```
1745 \list@create{\end@lemmas}
1746 %
```

`\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.

```
1747 \newcommand{\dummy@edtext}[2]{#1}
1748 %
```

`\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`.

```
1749 \newcommand{\dummy@edtext@showlemma}[2]{\showlemma{#1}}%
1750 %
```

We are going to need another macro that takes one argument and ignores it entirely. This is supplied by the \TeX `\@gobble{<arg>}`.

`\no@expands` We need to turn off macro expansion for certain sorts of macros we are likely to see within the lemma and within the notes.

`\morenoexpands`

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 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.

³¹Since “control sequences equivalent to characters are not expandable”—*The TeXbook*, answer to Exercise 20.14.

(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 Xe \TeX .)

```

1751 \newcommand*{\no@expands}{%
1752   \let\select@lemmafnt=0%
1753   \let\startsub=\relax \let\endsub=\relax
1754   \let\startlock=\relax \let\endlock=\relax
1755   \let\edlabel=\@gobble
1756   \let\setline=\@gobble \let\advanceline=\@gobble
1757   \let\sameword\sameword@inedtext%
1758   \let\edtext=\dummy@edtext
1759   \let\edindex\dummy@edindex%
1760   \l@dtabnoexpands
1761   \l@noexpands@edgl%
1762   \let\linenumannotation=\@gobble%
1763   \morenoexpands}
1764 \let\morenoexpands=\relax
1765
1766 %
```

`\@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.

```

1767 \newcommand{\@tag}{}
1768 %
```

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

```
1769 \newcount\@edtext@level%
1770 \@edtext@level=0%
1771 %
```

`\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.

```
1772 \newif\if@edtext@secondarg%
1773 %
```

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

```
1774 \newcounter{edtext}
1775 \renewcommand{\theedtext}{\edtxt@the\c@edtext}%
1776 %
```

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

```
1777 \newcommand{\edtext}[2]{\leavevmode%
1778 %
```

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

```
1779 \ifnumberedpar%
1780 %
```

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

```
1781 \@check@edtext@args{#1}{#2}%
1782 %
```

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

```
1783 \global\advance\@edtext@level by 1%
1784 %
```

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

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`.

```
1785 \stepcounter{edtext}%
1786 \csxdef{thisedtext@the\@edtext@level}{\theedtext}%
1787 %
```

By default, we do not use `\lemma`

```
1788 \global\@lemmacommand@false%
1789 %
```



```
1790 \begingroup%
1791 %
```

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`.

```
1792 \ifledRcol%
1793 \ifcsvoid{sw@list@edtextR@the\@edtext@level}%
1794 {\global\let\sw@inthisedtext\empty}%
1795 {\expandafter\gl@p\csname sw@list@edtextR@the\
@edtext@level\endcsname\to\sw@inthisedtext}%
1796 \else%
1797 \ifcsvoid{sw@list@edtext@the\@edtext@level}%
1798 {\global\let\sw@inthisedtext\empty}%
1799 {\expandafter\gl@p\csname sw@list@edtext@the\@edtext@level
\endcsname\to\sw@inthisedtext}%
1800 \fi%
1801 %
```

`\@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.

```
1802 \global\renewcommand{\@tag}{%
1803 \no@expands #1%
1804 }%
1805 %
```

`\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`.

```
1806 \set@line%
1807 %
```

`\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`.

```
1808 \ifledRcol \global\insert@countR \z@%
1809 \else \global\insert@count \z@ \fi%
1810 %
```

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.

```

1811 \edtext@secondarg@true%
1812 \ignorespaces #2\relax%
1813 \edtext@secondarg@false%
1814 %

```

With Xe_{La}TeX, you must track whether the language reads left to right (English) or right to left (Arabic). `reledmac` defines an `\if@RTL` boolean test is not already defined.

```

1815 \if@RTL%
1816 \flag@start@RTL%
1817 \else%
1818 \flag@start%
1819 \fi%
1820 %

```

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

```

1821 \if@lemmacommand%
1822 \ifledRcol%
1823 \write\linenum@outR{\string\@lemma}%
1824 \else%
1825 \write\linenum@out{\string\@lemma}%
1826 \fi%
1827 \fi%
1828 %

```

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.

```

1829 \endgroup%
1830 \ifdef\hypertarget%
1831 {%
1832 \Hy@raisedlink@left{\hypertarget{\csuse{thisedtext@the\
@edtext@level}:start}{}}%
1833 \showlemma{#1}%
1834 \Hy@raisedlink{\hypertarget{\csuse{thisedtext@the\@edtext@level}:
end}{}}%
1835 }%
1836 {%
1837 \showlemma{#1}%
1838 }%
1839 %

```

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.

```

1840 \ifx\end@lemmas\empty \else%
1841 \gl@p\end@lemmas\to\x@lemma%
1842 \x@lemma%
1843 \global\let\x@lemma=\relax%
1844 \fi%
1845 \if@RTL%
1846 \flag@end@RTL%
1847 \else%
1848 \flag@end%
1849 \fi%
1850 %

```

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`.

```

1851 \global\@noneed@Footnotefalse%
1852 \global\advance\@edtext@level by -1%
1853 \global\@lemmacommand@false%
1854 %

```

We also reset `\@beforeinsertofthisedtext`

```

1855 \global\let\@beforeinsertofthisedtext\relax%
1856 %

```

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

```

1857 \else%
1858 \showlemma{#1} (\textbf{\textsc{Edtext outside numbered paragraph}})\
led@err@edtextoutsidepstart%
1859 \fi%
1860 }%
1861
1862
1863 %

```

`\@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.

```

1864 \newcommand{\@check@edtext@args}[2]{%
1865 \begingroup%
1866 \noexpandarg%
1867 \IfSubStr{#1}{\pstart}{\led@err@PstartInEdtext{first}}{}%
1868 \IfSubStr{#1}{\pend}{\led@err@PendInEdtext{first}}{}%
1869 \IfSubStr{#2}{\pstart}{\led@err@PstartInEdtext{second}}{}%

```

```

1870 \IfSubStr{#2}{\pend}{\led@err@PendInEdtext{second}}{}}%
1871 \endgroup%
1872 }%
1873 %
1874 %

```

\@beforeinsertofthistext `\@beforeinsertofthistext` 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`.

```

1875 \let\@beforeinsertofthistext\relax
1876 %

```

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

```

1877 \newif\ifnumberline
1878 \numberlinetrue
1879 %

```

\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.

```

1880 \newcommand*{\set@line}{%
1881   \ifledRcol
1882     \ifx\line@listR\empty
1883       \global\noteschanged@true
1884       \xdef\l@d@nums{000|000|000|000|000|000|\edfont@info}%
1885     \else
1886       \gl@p\line@listR\to\@tempb
1887       \xdef\l@d@nums{\@tempb|\edfont@info}%
1888       \global\let\@tempb=\undefined
1889     \fi
1890     \ifx\annot@listR\empty%
1891       \xdef\l@current@annotR{||}%
1892     \else%
1893       \gl@p\annot@listR\to\@tempb%
1894       \xdef\l@current@annotR{\@tempb}%
1895       \global\let\@tempb=\undefined%
1896     \fi%
1897   \else
1898     \ifx\line@list\empty
1899       \global\noteschanged@true

```

```

1900 \xdef\l@d@nums{000|000|000|000|000|000|\edfont@info}%
1901 \else
1902 \gl@p\line@list\to\@tempb
1903 \xdef\l@d@nums{\@tempb|\edfont@info}%
1904 \global\let\@tempb=\undefined
1905 \fi
1906 \ifx\annot@list\empty%
1907 \xdef\l@current@annot{|\}%
1908 \else%
1909 \gl@p\annot@list\to\@tempb%
1910 \xdef\l@current@annot{\@tempb}%
1911 \global\let\@tempb=\undefined%
1912 \fi%
1913 \fi}
1914
1915 %

```

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

```

1916 \newcommand*{\edfont@info}{\f@encoding/\f@family/\f@series/\f@shape}
1917
1918 %

```

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. 145).

```

1919 \newcommand*{\lemma}[1]{%
1920 \global\@lemmacommand@true%
1921 \global\renewcommand{\@tag}{%
1922 \no@expands #1%
1923 }%
1924 \ignorespaces%
1925 }%
1926 %

```

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

```

1927 \newcommand{\@lemma}{%
1928 \booltrue{lemmacommand@the\@edtext@level}%
1929 }%
1930 %

```

\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.

```
1931 \newif\if@lemmacommand@%
1932 %
```

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. 113): 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.

```

1933 \newcommand*{\linenum}[1]{%
1934   \xdef\@tempa{#1|||||noexpand\\l@d@nums}%
1935   \global\let\l@d@nums=\empty
1936   \expandafter\line\set\@tempa|\\ignorespaces}
1937 %

```

`\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.

```

1938 \def\line@set#1|#2\\#3|#4\\{%
1939   \gdef\@tempb{#1}%
1940   \ifx\@tempb\empty
1941     \l@{d@add{#3}%
1942   \else
1943     \l@{d@add{#1}%
1944   \fi
1945   \gdef\@tempb{#4}%
1946   \ifx\@tempb\empty\else
1947     \l@{d@add{}}\line@set#2\\#4\\}%
1948   \fi}
1949 %

```

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

```

1950 \newcommand{\l@d@add}[1]{\xdef\l@d@nums{\l@d@nums#1}}
1951
1952 %

```

`\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.

```

1953 \newcommand*{\lineannot}[1]{%
1954   \lineannot@set#1|}%
1955 }%
1956 \def\lineannot@set#1|#2|{%
1957   \expandafter\parse@annot#1|#2|%
1958   \IfStrEq{#1}{}%
1959     {\let\@tempa\annot@start}%
1960     {\def\@tempa{#1}}%
1961   \IfStrEq{#2}{}%
1962     {\let\@tempb\annot@start}%
1963     {\def\@tempb{#2}}%
1964   \xdef\l@current@annot{\@tempa|\@tempb}%
1965 }%
1966 %

```

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 1967 \newcommand{\get@sw@txt}[1]{%
1968   \begingroup%
1969   \swnoexpands%
1970   %.
```

Using case sensibility option.

```
1971   \ifsw@caseinsensitive%
1972   \def\@tmpa##1{\lowercase{##1}}%
1973   \else%
1974   \def\@tmpa##1{##1}%
1975   \fi%
1976   %
```

And now, define `\sw@txt`.

```
1977   \@tmpa{\protected@xdef\sw@txt{##1}}%
1978   \endgroup%
1979 }%
1980 %
```

Allow some macros inside `\sameword`. We use `\RenewExpandableDocumentCommand` to get expandable command with optional argument. Cf. <https://tex.stackexchange.com/a/384783/7712>.

```

\sw@noexpand\newcommand{\swnoexpands}{%
1982   \let\sameword\l@secondmandarg%Allow to have nested \sameword
1983   \let\emph\@firstofone%
1984   \let\textit\@firstofone%
1985   \let\textbf\@firstofone%
1986   \let\textsc\@firstofone%
1987   \let\framebox\@firstofone%
1988   \let\edtext\dummy@edtext%
1989   \RenewExpandableDocumentCommand{\edindex}{om}{}%
1990   \ifdefined\index%
1991     \RenewExpandableDocumentCommand{\index}{om}{}%
1992   \fi%
1993   \let\selectlanguage\@gobble%
1994   \let\foreignlanguage\@secondoftwo%
1995   \ifdefined\xpg@loaded%
1996     \renewcommand\do[1]{%
1997       \expandafter\RenewExpandableDocumentCommand\expandafter{\csname
text##1\endcsname}{om}{###2}%
1998     }%
1999     \expandafter\docsvlist\expandafter{\xpg@loaded}%
2000   \fi%
2001 }%
2002 %

```

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

```

2003 \newcommandx{\sameword}[2][1,usedefault]{%
2004   \ifxetex%
2005     \if@RTL%
2006       \led@err@samewordRTL%
2007     \fi%
2008   \fi%
2009   \leavevmode%
2010   \get@sw@txt{#2}%
2011 %

```

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

```

2012   \unless\ifledRcol%
2013     \csnumgdef{sw@\sw@txt}{\csuse{sw@\sw@txt}+\@ne}%
2014 %

```

Then, write its value to the numbered file.

```

2015   \protected@write\linenum@out{\string\@sw{\sw@txt}{\csuse{sw@\sw@txt
}}{#1}}%
2016 %

```

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

```

2017   \else%
2018     \csnumgdef{sw@\sw@txt}{\csuse{sw@\sw@txt}+\@ne}%

```

```

2019 \protected@write\linenum@outR{}\string\@sw{\sw@txt}{\csuse{sw@\sw@txt
2020 }}{#1}}%
2021 \fi%
2022 %

```

And print the word.

```

2022 #2%
2023 }%
2024 %

```

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

```

\if@addsw 25 \newif\if@addsw%
2026 %

```

\@sw The command printed in the auxiliary files.

```

2027 \newcommand{\@sw}[3]{%
2028 \get@sw@txt{#1}%
2029 \unless\ifledRcol%
2030 %

```

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

```

2031 \csxdef{sw@\sw@txt @\the\absline@num @\the\section@num}{#2}%
2032 %

```

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

```

2033 \numdef{\prev@line}{\the\absline@num-1}%
2034 \ifcsundef{sw@\sw@txt @\prev@line @\the\section@num}{%
2035 \csnumgdef{sw@\sw@txt @\prev@line @\the\section@num}{#2-1}%
2036 }{}%
2037 %

```

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

```

2038 \numdef{\the@sw}{#2-\csuse{sw@\sw@txt @\prev@line @\the\section@num}}%
2039 %

```

And do the same thing for the right side.

```

2040 \else%
2041 \csxdef{sw@\sw@txt @\the\absline@numR @\the\section@numR @R}{#2}%
2042 \numdef{\prev@line}{\the\absline@numR-1}%
2043 \ifcsundef{sw@\sw@txt @\prev@line @\the\section@numR @R}{%
2044 \csnumgdef{sw@\sw@txt @\prev@line @\the\section@numR @R}{#2-1}%
2045 }{}%
2046 \numdef{\the@sw}{#2-\csuse{sw@\sw@txt @\prev@line @\the\section@numR @R
2047 }}%
2048 \fi%
2049 %

```

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

```

2049 \@tempcnta=\@edtext@level
2050 \@whilenum{\@tempcnta>0}\do{%
2051   \ifcsdef{sw@list@edtext@tmp@\the\@tempcnta}%
2052     {%
2053       \@addswfalse%
2054       \notbool{lemmacommand@\the\@tempcnta}%
2055       {\@addswtrue}%
2056       {\IfStrEq{#3}{inlemma}%
2057        {\@addswtrue}%
2058        {%
2059          \def\do##1{%
2060            \ifnumequal{##1}{\the\@tempcnta}%
2061              {\@addswtrue\listbreak}%
2062              {}}%
2063          }%
2064          \docsvlist{#3}%
2065        }%
2066      }%
2067   \if@addsw%
2068     \letcs{\@tmp}{sw@list@edtext@tmp@\the\@tempcnta}%
2069     \ifledRcol%
2070       \xright@appenditem{\the@sw}{\the\absline@numR}}\to\@tmp%
2071     \else%
2072       \xright@appenditem{\the@sw}{\the\absline@num}}\to\@tmp%
2073     \fi%
2074     \cslet{sw@list@edtext@tmp@\the\@tempcnta}{\@tmp}%
2075     \fi%
2076   }%
2077   {%
2078     \advance\@tempcnta by -1%
2079   }%
2080 }%
2081 %

```

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

```

2082 \newcommandx{\sameword@inedtext}[2][1,usedefault]{%
2083   \get@sw@txt{#2}%
2084   \unless\ifledRcol%
2085 %

```

Just a precaution.

```

2086   \ifx\sw@list@inedtext\empty%
2087     \def\the@sw{999}%
2088     \def\this@absline{-99}%
2089   \else%
2090 %

```

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

```

2091 \glp\sw@list@inedtext\to\@tmp%
2092 \edef\the@sw{\expandafter\@firstoftwo\@tmp}%
2093 \edef\this@absline{\expandafter\@secondoftwo\@tmp}%
2094 \fi%
2095 %

```

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

```

2096 \ifcsdef{sw@sw@txt @\this@absline @\the\section@num}{%
2097 \numdef{\prev@line}{\this@absline-1}%
2098 \numdef{\sw@atthisline}{\csuse{sw@sw@txt @\this@absline @\the\
section@num}-\csuse{sw@sw@txt @\prev@line @\the\section@num}}%
2099 }%
2100 {\numdef{\sw@atthisline}{0}}%
2101 %

```

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

```

2102 \ifnumgreater{\sw@atthisline}{1}%
2103 {\showwordrank{#2}{\the@sw}}%
2104 {#2}%
2105 %

```

And the same for right side.

```

2106 \else%
2107 \ifx\sw@list@inedtext\empty%
2108 \def\the@sw{999}%
2109 \def\this@absline{-99}%
2110 \else%
2111 \glp\sw@list@inedtext\to\@tmp%
2112 \edef\the@sw{\expandafter\@firstoftwo\@tmp}%
2113 \edef\this@absline{\expandafter\@secondoftwo\@tmp}%
2114 \fi%
2115 \ifcsdef{sw@sw@txt @\this@absline @\the\section@numR @R}{%
2116 \numdef{\prev@line}{\this@absline-1}%
2117 \numdef{\sw@atthisline}{\csuse{sw@sw@txt @\this@absline @\the\
section@numR @R}-\csuse{sw@sw@txt @\prev@line @\the\section@numR @R}}%
2118 }%
2119 {\numdef{\sw@atthisline}{0}}%
2120 \ifnumgreater{\sw@atthisline}{1}%
2121 {\showwordrank{#2}{\the@sw}}%
2122 {#2}%
2123 \fi%
2124 }%
2125 %

```

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

```

2126 \newcommand{\showwordrank}[2]{%
2127   #1\textsuperscript{#2}%
2128 }%
2129 %

```

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

Here are numbers and flags that are used internally in the course of the paragraph decomposition.

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.

```

2130 \newbox\raw@text
2131 \newif\ifnumberedpar@
2132 \newcount\num@lines
2133 \newbox\one@line
2134 \newcount\par@line
2135 %

```

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

```

2136 \newcount\pstarts@typeset@L%
2137 \newcount\pstarts@read@L%
2138 %

```

`\pstart` \pstart starts the paragraph by clearing the `\inserts@list` list and other relevant variables, and then arranges for the subsequent text to go into the `\raw@text` box.
`\AtEveryPstart` \pstart needs to appear at the start of every paragraph that is to be numbered; the
`\AtStartEveryPstart` \autopar command below may be used to insert these commands automatically.
`\numberpstarttrue` Beware: everything that occurs between `\pstart` and `\pend` is happening within
`\numberpstartfalse` a group; definitions must be global if you want them to survive past the end of the
`\labelpstarttrue` paragraph.
`\labelpstartfalse`

```

2139 \thepstart
2140 \if@every@pstart@star@
2141 \newcommand{\AtStartEveryPstart}[1]{%
2142   \ifstrempy{#1}%
2143   {\gdef\@at@start@every@pstart{}}%
2144   {\gdef\@at@start@every@pstart{#1}}%
2145 }%
2146 \def\@at@start@every@pstart{}%
2147 \newif\if@every@pstart@star@%
2148 \newcommand{\AtEveryPstart}[1]{%
2149   \ifstrempy{#1}%
2150   {\gdef\@at@every@pstart{}}%
2151   {\gdef\@at@every@pstart{\noindent#1}}%
2152   \global\@at@every@pstart@star@false%
2153 }%
2154 \WithSuffix\newcommand\AtEveryPstart*[1]{%
2155   \ifstrempy{#1}%
2156   {\gdef\@at@every@pstart{}}%
2157   {\gdef\@at@every@pstart{#1}}%
2158   \global\@at@every@pstart@star@true%
2159 }%
2160 \def\@at@every@pstart{}%
2161
2162 \newcounter{pstart}
2163 \renewcommand{\thepstart}{\bfseries\@arabic@c@pstart}. }
2164 \newif\ifnumberpstart
2165 \numberpstartfalse
2166 \newif\iflabelpstart
2167 \labelpstartfalse
2168 \newcommandx*\pstart[2][1,2,usedefault]{%
2169   \normal@pars%
2170   \ifboolexpr{%
2171     test {\ifstrempy{#1}}%
2172     and test {\ifstrempy{#2}}%
2173   }%
2174   {\@at@every@pstart}%
2175   {%
2176     \ifstrempy{#1}{\noindent#1}%
2177     \ifstrempy{#2}{#2}%
2178   }%
2179   \ifautopar%
2180   \autopar%

```

```

2181 \fi%
2182 \ifluatex%
2183   \edef\l@luatexttextdir@L{\the\textdir}%
2184 \fi%
2185 \@nbreaktrue%
2186 \ifnumbering \else%
2187   \led@err@PstartNotNumbered%
2188   \beginnumbering%
2189 \fi%
2190 \ifnumberedpar@%
2191   \led@err@PstartInPstart%
2192   \pend%
2193 \fi%
2194 \list@clear{\inserts@list}%
2195 \global\let\next@insert=\empty%
2196 \begingroup\normal@pars%
2197 \global\advance \l@dnumstartL\@ne
2198 \global\advance \pstart@typeset@L\@ne%
2199 \global\advance \pstart@read@L\@ne%
2200 \global\setbox\raw@text=\vbox\bgroup%
2201 \if@nbreak%
2202   \if@afterindent\else%
2203     \noindent%
2204     \global\@afterindenttrue%
2205   \fi%
2206 \fi%
2207 \ifboolexpr{%
2208   bool{autopar}%
2209   and bool{by@autopar}%
2210 }%
2211 {}%
2212 {%
2213   \ifnumberpstart%
2214     \ifinstanza\else%
2215       \ifsidepstartnum\else%
2216         \thepstart%
2217       \fi%
2218     \fi%
2219   \fi%
2220 }%
2221 \numberedpar@true%
2222 \iflabelpstart%
2223   \protected@edef\@currentlabel{\p@pstart\thepstart}%
2224 \fi%
2225 \l@dzeropenalties%
2226 \@at@start@every@pstart%
2227 \global\by@autoparfalse%
2228 \ignorespaces%because not automatically ignored if an optional argument
is used (classical TeX behavior for space after commands)
2229 }

```



```
2230 %
```

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

```
2231 \newcommand*{\pend}[2][1,2,usedefault]{\ifnumbering \else%
2232   \led@err@PendNotNumbered%
2233   \fi%
2234   \global\l@dskipversenumberfalse%
2235   \ifnumberedpar@ \else%
2236   \led@err@PendNoPstart%
2237   \fi%
2238 %
```

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.

```
2239 \l@dzero penalties%
2240 \@at@end@every@pend%
2241 \endgraf\global\num@lines=\prevgraf\egroup%
2242 \global\par@line=0%
2243 \get@firsthalf@antilabe@wd%
2244 %
```

Output the lines.

```
2245 \loop\ifvbox\raw@text%
2246   \do@line%
2247   \repeat%
2248 %
```

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

```
2249 \flush@notes%
2250 \endgroup%
2251 \ignorespaces%
2252 %
```

Increase pstart counter.

```
2253 \ifnumberpstart%
2254   \global\pstartnumtrue%
2255   \fi%
2256   \addtocounter{pstart}{1}%
2257   \ifcontinuousnumberingwithcolumns%
2258     \addtocounter{pstartL}{1}%
2259     \addtocounter{pstartR}{1}%
2260   \fi%
2261 %
```

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

```

2262 \normal@pars%
2263 \ifboolexpr{%
2264   test {\ifstrempy{#1}}%
2265   and test {\ifstrempy{#2}}%
2266 }%
2267 {\at@every@pend}%
2268 {%
2269   \ifstrempy{#1}{\noindent#1}%
2270   \ifstrempy{#2}{#2}%
2271 }%
2272 %

```

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.

```

2273 \@nobreakfalse%
2274 \ifautopar%
2275   \autopar%
2276 \fi%
2277 %

```

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.

```

2278 \ifbypstart@%
2279   \begingroup%
2280     \unless\ifinstanza%
2281       \let\leavevmode\relax%
2282       \setlinenum{0}%
2283     \fi%
2284   \endgroup%
2285   \resetprevline@%
2286 \fi%
2287 }
2288 %

```

Here, two macros to insert content after every `\pend`, between numbered line. `\AtEveryPend` is the user macro, `\at@every@pend` is macro set by it.

```

\AtEveryPend89
\at@every@pend90
\ifat@every@pend@star91 \newif\ifat@every@pend@star@%
2292 \newcommand{\AtEveryPend}[1]{%
2293   \ifstrempy{#1}%
2294   {\gdef\at@every@pend{}}%
2295   {\gdef\at@every@pend{\noindent#1}}%
2296 \global\at@every@pend@star@false%

```

```

2297 }%
2298 \WithSuffix\newcommand\AtEveryPend*[1]{%
2299   \ifstrempy{#1}%
2300   {\gdef\at@every@pend{}}%
2301   {\gdef\at@every@pend{#1}}%
2302   \global\at@every@pend@star@true%
2303 }%
2304 \xdef\at@every@pend{}%
2305
2306 %

```

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

```

2307 \newcommand{\AtEndEveryPend}[1]{%
2308   \ifstrempy{#1}%
2309   {\xdef\@at@end@every@pend{}}%
2310   {\gdef\@at@end@every@pend{#1}}%
2311 }%
2312 \def\@at@end@every@pend{}%
2313 %

```

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

```

2314 \newcommand*\l@dzeropenalties{%
2315   \brokenpenalty \z@ \clubpenalty \z@
2316   \displaywidowpenalty \z@ \interlinepenalty \z@ \predisplaypenalty \z@
2317   \postdisplaypenalty \z@ \widowpenalty \z@}
2318
2319 %

```

\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.

```

2320 \newif\ifautopar
2321 \newif\ifby@autopar%
2322 \newcommand*{\autopar}{
2323   \ifledRcol
2324     \ifnumberingR \else
2325       \led@err@AutoparNotNumbered
2326       \beginnumberingR
2327       \fi
2328     \else
2329       \ifnumbering \else
2330         \led@err@AutoparNotNumbered
2331         \beginnumbering
2332         \fi
2333       \fi
2334       \autopartrue
2335       \everypar{\setbox0=\lastbox
2336         \endgraf \vskip-\parskip
2337         \global\by@autopartrue%
2338         \pstart \noindent \kern\wd0%
2339         \ifnumberpstart%
2340           \ifinstanza\else%
2341             \thepstart%
2342           \fi%
2343         \fi%
2344         \let\par=\pend}%
2345       \ignorespaces}
2346 %

```

`\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.

```

2347 \newcommand*{\normal@pars}{\everypar{}\let\par\endgraf}
2348
2349 %

```

`\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.

```

2350 \newif\ifautopar@pause
2351 %

```

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.

```

2352 \newcommand*{\l@dunhbox@line}[1]{\unhbox #1}
2353 \newcommand*{\do@line}{%
2354   {\vbadness=10000
2355     \splittopskip=\z@
2356     \do@linehook
2357   \l@demptyd@ta
2358     \global\setbox\one@line=\vsplit\raw@text to\baselineskip}%
2359   \unvbox\one@line \global\setbox\one@line=\lastbox
2360   \getline@num
2361   \IfStrEq{\led@pb@setting}{before}{\led@check@pb\led@check@nopb}{%}
2362   \ifnum\@lock>\@ne
2363     \inserthangingsymboltrue
2364   \else
2365     \inserthangingsymbolfalse
2366   \fi
2367   \check@pb@in@verse
2368   \ifl@dhiddenumber%
2369     \global\l@dhiddenumberfalse%
2370     \f@x@l@cks%
2371   \else%
2372     \affixline@num%
2373   \fi%
2374 }%

```

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

```

2375 \xifinlist{\the\pstarts@typeset@L}{\eled@sections@@}%
2376   {\print@eledsection}%
2377   {\print@line}%
2378   \IfStrEq{\led@pb@setting}{after}{\led@check@pb\led@check@nopb}{%}
2379 }%
2380 %

```

VII.2.2 Process for “normal” line

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

```

2381 \def\print@line{
2382 %

```

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

```
2383 \affixpstart@num%
2384 %
```

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

```
2385 \hb@xt@ \linewidth{%
2386 %
```

User hooks.

```
2387 \do@insidelinehook%
2388 \csuse{insidethis@the\absline@num @the\section@num}%
2389 \global\csundef{insidethis@the\absline@num @the\section@num}%
2390 %
```

Left line number

```
2391 \l@dld@ta%
2392 %
```

Prepare text to be inserted before notes.

```
2393 \if@firstlineofpage%
2394 \set@Xtxtbeforenotes%
2395 \set@txtbeforenotesX%
2396 \global\@firstlineofpagefalse%
2397 \fi%
2398 %
```

Insert footnotes made of manuscripts data and critical footnotes.

```
2399 \ifdefstring{\ms@data@position}{msdata-regular}{%
2400 \insert@msdata%
2401 \add@inserts%
2402 \add@Xgroupbyline%
2403 }{%
2404 \add@inserts%
2405 \add@Xgroupbyline%
2406 \insert@msdata%
2407 }%
2408 %
```

Insert marginal notes.

```
2409 \affixside@note%
2410 %
```

Print left notes.

```
2411 \l@dlsn@te
2412 %
```

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

```

2413     {\ledllfill\hb@xt@ \wd\one@line{%
2414         \new@line%
2415 %

```

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

```

2416     \continuousnumberingwithcolumns@sync@linenumber@singletext%
2417 %

```

If we use Lua_{TeX} then restore the direction.

```

2418     \ifluatex%
2419         \texdir\l@luatextextdir@L%
2420     \fi%
2421 %

```

Insert, if needed, the hanging symbol.

```

2422     \inserthangingsymbol%
2423 %

```

And so, print the line.

```

2424     \l@dunhbox@line{\one@line}}%
2425 %

```

Right line number

```

2426     \ledrlfill\l@drd@ta%
2427 %

```

Print right notes.

```

2428     \l@drsn@te%
2429     }}%
2430 %

```

And reinsert penalties (for page breaking)...

```

2431     \add@penalties%
2432 }
2433 %

```

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.

```

2434 \def\print@eledsection{%
2435     \disable@edindex%
2436     \if@firstlineofpage%
2437         \set@Xtxtbeforenotes%
2438         \set@txtbeforenotesX%

```

```

2439 \global\@firstlineofpagefalse%
2440 \fi%
2441 \ifdefstring{\ms@data@position}{msdata-regular}{%
2442 \insert@msdata%
2443 \add@inserts%
2444 \add@Xgroupbyline%
2445 }{%
2446 \add@inserts%
2447 \add@Xgroupbyline%
2448 \insert@msdata%
2449 }%
2450 \affixside@note%
2451 \numdef{\temp@}{\pstarts@typeset@L-1}%
2452 \xifinlist{\temp@}{\eled@sections@@}{\@nbreaktrue}{\@nbreakfalse}%
2453 \@eled@sectioningtrue%
2454 \csuse{eled@sectioning@\the\pstarts@typeset@L}%
2455 \@eled@sectioningfalse%
2456 \global\csundef{eled@sectioning@\the\pstarts@typeset@L}%
2457 \if@RTL%
2458 \hspace{-3\paperwidth}%
2459 {\hbox{\l@dunhbox@line{\one@line}} \new@line}%
2460 \else%
2461 \hspace{3\paperwidth}%
2462 {\new@line \hbox{\l@dunhbox@line{\one@line}}}%
2463 \fi%
2464 \vskip-\baselineskip%
2465 \continuousnumberingwithcolumns@sync@linenumber@singletext%
2466 \restore@edindex%
2467 }
2468 %

```

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.

```

2469 \newcommand*{\do@linehook}{}
2470 \newcommand*{\do@insidelinehook}{}
2471 %

```

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

```

2472 \newcommand*{\dolinehook}[1]{\gdef\do@linehook{#1}}%
2473 \newcommand*{\doinsidelinehook}[1]{\gdef\do@insidelinehook{#1}}%
2474
2475 %

```


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

```

2476 \newcommand{\doinsidethislinehook}[1]{%
2477   \leavevmode%In case it begins with a \pstart, ensure the \@insidethisline
      is written after \@nl
2478   \ifledRcol%
2479     \write\linenum@outR{\string\@insidethisline[\unexpanded{#1}]}%
2480   \else%
2481     \write\linenum@out{\string\@insidethisline[\unexpanded{#1}]}%
2482   \fi%
2483 }%
2484 %

```

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

```

2485 \newcommand{\@insidethisline}[1]{%
2486   \ifledRcol%
2487     \csgappto{insidethisR@the\absline@numR @the\section@numR}{#1}%
2488   \else%
2489     \csgappto{insidethis@the\absline@num @the\section@num}{#1}%
2490   \fi%
2491 }%
2492 %

```

VII.2.5 Sidenotes and marginal line number initialization

`\l@emptyd@ta` Nulls the `\. . .d@ta`, which may later hold line numbers. Similarly for `\l@dcsnotetext`, `\l@dcsnotetext@l`, `\l@dcsnotetext@r` for the texts of the sidenotes, left and right notes.

```

\l@dcsnotetext
2493 \newcommand*\l@emptyd@ta{%
\l@dcsnotetext@l
2494   \gdef\l@dld@ta{%
\l@dcsnotetext@r
2495     \gdef\l@drd@ta{%
2496       \gdef\l@dcsnotetext@l{%
2497         \gdef\l@dcsnotetext@r{%
2498           \gdef\l@dcsnotetext{}
2499         }
2500       }
2501     }
2502   }
2503 %

```

`\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.

```

2501 \newcommand{\l@dlsn@te}{%
2502   \ifboolexpr{%
2503     bool {l@dprintingcolumns}%
2504     and bool {ledRcol}%
2505   }{% If we are on a right column
2506     \@tempdima=\@morespace@leftnote@rightcolumn%

```

```

2507 }{%
2508   \@tempdima=\z@%
2509 }%
2510 \hb@xt@ \z@{\hss\box\l@dlp@rbox\kern\ledlsnotesep\hskip\@tempdima}%
2511 }%
2512 \newcommand{\l@drsn@te}{%
2513   \ifboolexpr{%
2514     bool {l@dprintingcolumns}%
2515     and not bool {ledRcol}%
2516   }{% If we are on a left column
2517     \@tempdima=\@morespace@rightnote@leftcolumn%
2518   }{%
2519     \@tempdima=\z@%
2520   }%
2521   \hb@xt@ \z@{\hskip\@tempdima\kern\ledrsnotesep\box\l@drp@rbox\hss}%
2522 }%
2523
2524 %

```

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

\ledrlfill

```

2525 \newcommand*{\ledllfill}{\hfil}
2526 \newcommand*{\ledrlfill}{}
2527
2528 %

```

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.

```

2529 \newcommand*{\getline@num}{%
2530   \global\advance\absline@num \@ne%
2531   \do@actions
2532   \do@ballast
2533   \ifnumberline
2534     \ifsublines@
2535       \ifnum\sub@lock<\tw@
2536         \global\advance\subline@num \@ne
2537       \fi
2538     \else
2539       \ifnum\@lock<\tw@
2540         \global\advance\line@num \@ne
2541         \global\subline@num \z@
2542       \fi
2543     \fi
2544   \fi
2545 }

```

2546 %

`\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. 185).

`\ballast@count` First we set up the required counters; they are initially set to zero, and will remain so
`\c@ballast` unless you type `\setcounter{ballast}{<some figure>}` in your document.

```
2547 \newcount\ballast@count
2548 \newcounter{ballast}
2549 \setcounter{ballast}{0}
2550 %
```

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.

```
2551 \newcommand*{\do@ballast}{\global\ballast@count \z@
2552 \begingroup
2553   \advance\absline@num \@ne
2554   \ifnum\next@actionline=\absline@num
2555     \ifnum\next@action>-1001\relax
2556     \global\advance\ballast@count by -\c@ballast
2557   \fi
2558   \fi
2559 \endgroup}
2560 %
```

`\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`.

```
2561 \newcommand*{\do@actions}{%
2562   \global\let\do@actions@next=\relax
2563   \ifnum\absline@num<\next@actionline\else
2564   %
```

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.

```
2565   \ifnum\next@action>-1001
2566     \global\page@num=\next@action
2567     \ifresumenummering@start%
```

```

2568 \setbox0=\hbox{}}%Required to get the correct page number, when the
\resumenumbering is just after a \newpage
2569 \ifnum\pausenumbering@page@num<\page@num%
2570 \global\resumenumbering@startfalse%
2571 \fi%
2572 \fi%
2573 \ifboolexpr{%
2574 bool{resumenumbering@start}%
2575 and test {\ifdimgreater{\pagedepth}{\z@}}%
2576 }%
2577 {}%
2578 {\global\@firstlineofpagetrue}%
2579 \ifcsdef{reset@line\the\absline@num @\the\section@num}%
2580 {%
2581 \global\line@num=\z@%
2582 \global\subline@num=\z@%
2583 \resetprevline@%
2584 }%
2585 {}%
2586 \global\resumenumbering@startfalse%
2587 \add@msdata@firstlineofpage%
2588 %

```

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`.)

```

2589 \else
2590 \ifnum\next@action<-4999
2591 \@l@dtmpcnta=-\next@action
2592 \advance\@l@dtmpcnta by -5001
2593 \ifsublines@
2594 \global\subline@num=\@l@dtmpcnta
2595 \else
2596 \global\line@num=\@l@dtmpcnta
2597 \fi
2598 %

```

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

```

2599 \else
2600 \@l@dtmpcnta=-\next@action
2601 \advance\@l@dtmpcnta by -1000
2602 \do@actions@fixedcode
2603 \fi
2604 \fi
2605 %

```

Now we get information about the next action off the list, and then set `\do@actions@next` so that we will call ourselves recursively: the next action might also be for this line.

There is no warning if we find `\actionlines@list` empty, since that will always happen near the end of the section.

```

2606 \ifx\actionlines@list\empty
2607 \gdef\next@actionline{1000000}%
2608 \else
2609 \gl@p\actionlines@list\to\next@actionline
2610 \gl@p\actions@list\to\next@action
2611 \global\let\do@actions@next=\do@actions
2612 \fi
2613 \fi
2614 %

```

Make the recursive call, if necessary.

```

2615 \do@actions@next}
2616
2617 %

```

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

```

2618 \newcommand*{\do@actions@fixedcode}{%
2619 \ifcase\@l@dttempcnta
2620 \or% % 1001 = starting sublineation
2621 \global\sublines@true
2622 \or% % 1002 = ending sublineation
2623 \global\sublines@false
2624 \or% % 1003 = starting locking number
2625 \global\@lock=\@ne
2626 \or% % 1004 = ending locking number
2627 \ifnum\@lock=\tw@
2628 \global\@lock=\thr@@
2629 \else
2630 \global\@lock=\z@
2631 \fi
2632 \or% % 1005 = starting locking subnumber
2633 \global\sub@lock=\@ne
2634 \or% % 1006 = ending locking subnumber
2635 \ifnum\sub@lock=\tw@
2636 \global\sub@lock=\thr@@
2637 \else
2638 \global\sub@lock=\z@
2639 \fi
2640 \or% % 1007 = skipping numbering
2641 \l@dskipnumbertrue
2642 \or% % 1008 = skipping numbering in stanza
2643 \l@dskipversenumbertrue%
2644 \or% % 1009 = hiding number
2645 \l@dhidenumbertrue
2646 \or% % 1010 = inserting msdata
2647 \add@msdata%
2648 \else
2649 \led@warn@BadAction
2650 \fi}

```

```
2651
2652
2653 %
```

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

`\thcolumns@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.

```
2654 \newcommand{\continuousnumberingwithcolumns@sync@linenumber@singletext}{%
2655   \ifcontinuousnumberingwithcolumns%
2656     \unless\ifafterendnumberingR%
2657       \new@lineR%
2658       \xappto\next@line@list@stuffR{%
2659         \unexpanded{\global\line@numR=}\the\line@num%
2660       }%
2661     \fi%
2662   \fi%
2663 }%
2664 %
```

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`

```

2665 \newcommand{\linenumannotation}[1]{%
2666   \leavevmode%In case it begins with a \pstart, ensure the \@annot is
written after \@nl
2667   \ifledRcol%
2668     \write\linenum@outR{\string\@annot[#1]}%
2669   \else%
2670     \write\linenum@out{\string\@annot[#1]}%
2671   \fi%
2672 }%
2673
2674 \newcommand{\resetlinenumannotation}[0]{%
2675   \leavevmode%In case it begins with a \pstart, ensure the \@resetannot is
written after \@nl
2676   \ifledRcol%
2677     \write\linenum@outR{\string\@resetannot}%
2678   \else%
2679     \write\linenum@out{\string\@resetannot}%
2680   \fi%
2681 }%
2682
2683 %

```

`\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.

```

2684 \def\Xlinenumannotationposition@side{after}%
2685 %

```

`\wraplinenumannotation@ref` The `\Xwraplinenumannotation@ref` macro is applied to line annotations in crossref.
`\wraplinenumannotation@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@ref`

```

2686 \def\Xwraplinenumannotation@ref{\textsuperscript}%
2687 \def\Xwraplinenumannotation@side{\textsuperscript}%
2688 \newtoggle{Xnoidenticallinenumannotation@ref}%
2689 \newtoggle{Xnoidenticallinenumannotation@side}%
2690 %

```

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

```

2691 \newif\iflinenumannotationotherside%
2692 %

```

`\@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.

```

2693 \newcommand{\@annot}[1]{%
2694   \store@annot@to@absline{#1}%
2695   \def\current@annot{#1}%
2696 }%
2697 \newcommand{\store@annot@to@absline}[1]{%
2698   \ifledRcol%
2699     \ifcsdef{annotR@the\absline@numR @the\section@numR}{%
2700       \csgappto{annotR@the\absline@numR @the\section@numR}{\
@linenumannotationsep#1}%
2701     }{%
2702       \csgdef{annotR@the\absline@numR @the\section@numR}{#1}%
2703     }%
2704   \else%
2705     \ifcsdef{annot@the\absline@num @the\section@num}{%
2706       \csgappto{annot@the\absline@num @the\section@num}{\
@linenumannotationsep#1}%
2707     }{%
2708       \csgdef{annot@the\absline@num @the\section@num}{#1}%
2709     }%
2710   \fi%
2711 }%
2712 %

```

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

```

2713 \let\current@annot=\empty%
2714 %

```

`\reset@current@annot` `\reset@current@annot` is called at each `\@n1` in numbered auxiliary files. It resets the
`\@resetannot` annotation of line numbers at each line.

The `\@resetannot` macro is written in the auxiliary files with `\resetlinenumannotation` macro. It resets the annotation.

```

2715 \newcommand{\reset@current@annot}{%
2716   \unless\ifnoresetlinenumannotation@
2717     \let\current@annot\empty%
2718   \fi
2719 }%
2720 \newcommand{\@resetannot}{%

```



```

2721 \let\current@annot\empty%
2722 }%
2723 %

```

\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.

```

2724 \def\parse@annot#1|#2|{%
2725 \gdef\annot@start{#1}%
2726 \gdef\annot@end{#2}%
2727 }%
2728 %

```

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

```

\@linenumannotationsep
2729 \newcommand{\setlinenumannotationsep}[1]{\gdef\@linenumannotationsep{#1}}%
2730 \def\@linenumannotationsep{,}%
2731 %

```

X Line number printing

\affixline@num `\affixline@num` just puts a left line number into `\l@dld@ta` or a right line number into `\l@drd@ta` as required.

To determine whether we need to affix a line number to this line, we compute the following:

$$\begin{aligned}
 n &= \text{int}((\text{linenum} - \text{firstlinenum}) / \text{linenumincrement}) \\
 m &= \text{firstlinenum} + (n \times \text{linenumincrement})
 \end{aligned}$$

(where `int` truncates a real number to an integer). m will be equal to linenum only if we are to paste a number on here. However, the formula breaks down for the first line to number (and any before that), so we check that case separately: if `\line@num ≤ \firstlinenum`, we compare the two directly instead of making these calculations.

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

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

```

2732 \newcommand*{\affixline@num}{%
2733 %

```

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).

```

2734 \ifledgroupnotesL@else
2735 \ifnumberline
2736 \ifl@dskipnumber

```

```

2737 \global\l@dskipnumberfalse
2738 \else
2739 \ifsublines@
2740 \l@dtempcntb=\subline@num
2741 \ifnum\subline@num>\c@firstsublinenum
2742 \l@dtempcnta=\subline@num
2743 \advance\l@dtempcnta by-\c@firstsublinenum
2744 \divide\l@dtempcnta by\c@sublinenumincrement
2745 \multiply\l@dtempcnta by\c@sublinenumincrement
2746 \advance\l@dtempcnta by\c@firstsublinenum
2747 \else
2748 \l@dtempcnta=\c@firstsublinenum
2749 \fi
2750 %

```

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.

```

2751 \ch@cksub@l@ck
2752 %

```

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

```

2753 \else
2754 \l@dtempcntb=\line@num
2755 %

```

Check on the `\linenumberlist` If it is `\empty` use the standard algorithm.

```

2756 \ifx\linenumberlist\empty
2757 \ifnum\line@num>\c@firstlinenum
2758 \l@dtempcnta=\line@num
2759 \advance\l@dtempcnta by-\c@firstlinenum
2760 \divide\l@dtempcnta by\c@linenumincrement
2761 \multiply\l@dtempcnta by\c@linenumincrement
2762 \advance\l@dtempcnta by\c@firstlinenum
2763 \else
2764 \l@dtempcnta=\c@firstlinenum
2765 \fi
2766 \else
2767 %

```

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

```

2768 \l@dtempcnta=\line@num
2769 \edef\rem@inder{\,\linenumberlist,\number\line@num,}%
2770 \edef\sc@n@list{\def\noexpand\sc@n@list
2771 ###1,\number\l@dtempcnta,###2|{\def\noexpand\rem@inder
2772 {####2}}}%
2773 \sc@n@list\expandafter\sc@n@list\rem@inder|%
2774 \ifx\rem@inder\empty%

```

```

2774         \advance\@l@tempcnta\@ne
2775     \fi
2776 \fi
2777 %

```

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

```

2778     \ch@ck@l@ck
2779 \fi
2780 %

```

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

```

2781     \ifnum\@l@tempcnta=\@l@tempcntb
2782     \ifl@dskipversenumber\else
2783 %

```

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
2784     \if@twocolumn
2785     \if@firstcolumn
2786         \gdef\l@dld@ta{\llap{\leftlinenum}}}%
2787         \iflinenumannotationotherside%
2788         \gdef\l@drd@ta{\rlap{\rightlinenumannotation}}}%
2789     \fi%
2790     \else
2791         \gdef\l@drd@ta{\rlap{\rightlinenum}}}%
2792         \iflinenumannotationotherside%
2793         \gdef\l@dld@ta{\llap{\leftlinenumannotation}}}%
2794     \fi%
2795 \fi
2796 \else
2797     \ifboolexpr{bool {l@dprintingcolumns} and test {\
ifnumgreater{\line@margin@columns}{\m@ne}}}%
2798         {\@l@tempcntb=\line@margin@columns}%
2799         {\@l@tempcntb=\line@margin}%
2800     \ifnum\@l@tempcntb>\@ne
2801         \advance\@l@tempcntb \page@num
2802     \fi
2803     \ifboolexpr{%
2804         bool {l@dprintingcolumns}%

```

```

2805         and (%)
2806         (test {\ifdefstring{\linenum@OnlyPages@ForColumns}{left
}}%
2807         and test {\ifnumodd{\page@num}}%
2808         )%
2809         or%
2810         (test {\ifdefstring{\linenum@OnlyPages@ForColumns}{right
}}%
2811         and not test {\ifnumodd{\page@num}}%
2812         )%
2813         )%
2814         }%
2815         {%
2816         {%
2817         \ifodd\l@dtmptcntb%
2818         \gdef\l@drd@ta{\rlap{{\rightlinenum}}}%
2819         \iflinenumannotationotherside%
2820         \gdef\l@dld@ta{\llap{{\leftlinenumannotation}}}%
2821         \fi%
2822         \else%
2823         \gdef\l@dld@ta{\llap{{\leftlinenum}}}%
2824         \iflinenumannotationotherside%
2825         \gdef\l@drd@ta{\rlap{{\rightlinenumannotation}}}%
2826         \fi%
2827         \fi%
2828         }%
2829         \fi
2830         \fi
2831         \fi
2832         %

```

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

```

2833         \f@x@l@cks
2834         \fi
2835         \fi
2836         \fi
2837     }
2838
2839     %

```

`\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.

```

2840 \newcommand*{\ch@cksub@l@ck}{%
2841     \ifcase\sub@lock
2842     \or
2843     \ifnum\sublock@disp=\@ne

```

```

2844         \@l@tempcntb=\z@ \@l@tempcnta=\@ne
2845     \fi
2846 \or
2847     \ifnum\sublock@disp=\tw@ \else
2848         \@l@tempcntb=\z@ \@l@tempcnta=\@ne
2849     \fi
2850 \or
2851     \ifnum\sublock@disp=\z@
2852         \@l@tempcntb=\z@ \@l@tempcnta=\@ne
2853     \fi
2854 \fi}
2855 %

```

Similarly for line numbers.

```

2856 \newcommand*{\check@l@ck}{%
2857     \ifcase\@lock
2858     \or
2859         \ifnum\lock@disp=\@ne
2860             \@l@tempcntb=\z@ \@l@tempcnta=\@ne
2861         \fi
2862     \or
2863         \ifnum\lock@disp=\tw@ \else
2864             \@l@tempcntb=\z@ \@l@tempcnta=\@ne
2865         \fi
2866     \or
2867         \ifnum\lock@disp=\z@
2868             \@l@tempcntb=\z@ \@l@tempcnta=\@ne
2869         \fi
2870 \fi}
2871 %

```

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

```

2872 \newcommand*{\fix@l@cks}{%
2873     \ifcase\@lock
2874     \or
2875         \global\@lock=\tw@
2876     \or \or
2877         \global\@lock=\z@
2878     \fi
2879     \ifcase\sub@lock
2880     \or
2881         \global\sub@lock=\tw@
2882     \or \or
2883         \global\sub@lock=\z@
2884     \fi}
2885 %
2886 %

```

XI Pstart number printing inside

Inside, the printing of the pstart number runs like the printing of the line number. There are only a few differences:

- The pstarts counter is upgraded in the `\pend` command. Consequently, the `\affixpstart@num` command has not to upgrade it, unlike the `\affixline@num` which upgrades the lines counter.
- To print the pstart number only at the beginning of a pstart, and not in every line, a boolean test is made. The `\pstartnum` boolean is set to TRUE at every `\pend`. It is tried in the `\leftpstartnum` and `\rightpstartnum` commands. After the try, it is set to FALSE.

```

\leftpstartnum 287
\rightpstartnum 288
\ifsidepstartnum 289
2890   \ifsidepstartnum
2891       \if@twocolumn
2892           \if@firstcolumn
2893               \gdef\l@dld@ta{\llap{\leftpstartnum}}}%
2894           \else
2895               \gdef\l@drd@ta{\rlap{\rightpstartnum}}}%
2896           \fi
2897       \else
2898           \@l@tempcntb=\line@margin%
2899           \ifnum\@l@tempcntb>\@ne
2900               \advance\@l@tempcntb \page@num
2901           \fi
2902           \ifodd\@l@tempcntb
2903               \gdef\l@drd@ta{\rlap{\rightpstartnum}}}%
2904           \else
2905               \gdef\l@dld@ta{\llap{\leftpstartnum}}}%
2906           \fi
2907       \fi
2908   \fi
2909
2910 }
2911 %
2912
2913 \newif\ifpstartnum
2914 \pstartnumtrue
2915 \newcommand*{\leftpstartnum}{
2916     \ifpstartnum\thepstart
2917     \kern\linenumsep\fi
2918     \global\pstartnumfalse
2919 }
2920 \newcommand*{\rightpstartnum}{
2921     \ifpstartnum

```

```

2922 \kern\linenumsep
2923 \thepstart
2924 \fi
2925 \global\pstartnumfalse
2926 }
2927 %

```

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.

```

2928 \list@create{\inserts@list}
2929 %

```

`\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`.

```

2930 \newcommand*{\add@inserts}{%
2931 \global\let\add@inserts@next=\relax
2932 %

```

If `\inserts@list` is empty, there are not any more notes or insertions for this paragraph, and we need not waste our time.

```

2933 \ifx\inserts@list\empty \else
2934 %

```

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.

```

2935 \ifx\next@insert\empty
2936 \ifx\insertlines@list\empty
2937 \global\noteschanged@true
2938 \gdef\next@insert{100000}%
2939 \else
2940 \glp\insertlines@list\to\next@insert
2941 \fi
2942 \fi
2943 %

```

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.

```

2944 \ifnum\next@insert=\absline@num
2945 \glp\inserts@list\to\@insert
2946 \@insert
2947 \global\let\@insert=\undefined
2948 \global\let\next@insert=\empty
2949 \global\let\add@inserts@next=\add@inserts
2950 \fi
2951 \fi
2952 %

```

Make the recursive call, if necessary.

```

2953 \add@inserts@next}
2954
2955 %

```

\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`.

```

2956 \newcommand{\add@Xgroupbyline}{%
2957 \unless\ifnocritical%
2958 \def\do##1{%Looping on the series
2959 \let\olddo\do%Save the old \do macro, that is this macro itself!
2960 \def\do####1{%Looping on the ##1@forinserting command
2961 \ifcsdef{##1@forinserting@####1}{%
2962 \X@beforeinsertion{##1}%
2963 \if@ledgroup%
2964 \global\setbox\@nameuse{mp##1footins}=\vbox%
2965 \else%
2966 \insert\csname ##1footins\endcsname%
2967 \fi%
2968 {%
2969 \Xsetparindent{##1}%
2970 \ifcsdef{Xhsize\csuse{series@display##1}@##1}%
2971 {\hsize \csuse{Xhsize\csuse{series@display##1}@##1}}%
2972 }%

```



```

2973     \if@ledgroup%
2974         \unvbox\@nameuse{mp##1footins}%
2975     \fi%
2976     \X@atbegininsertion{##1}%
2977     \ifcsstring{series@display##1}{normal}%
2978     {%
2979         \Xledsetnormalparstuff{##1}%
2980         \rule\z@\splittopskip%
2981     }%
2982     {}%
2983     \csuse{##1@forinserting@###1}%
2984     \strut\par%
2985 }%
2986 \global\csundef{##1@forinserting@###1}%
2987 }%
2988 {}%
2989 }%
2990 \ifcsdef{##1@forinserting}{%
2991     \dolistcsloop{##1@forinserting}%
2992 }{}%
2993 \global\csundef{##1@forinserting}%
2994 \let\do\olddo%Restore old do
2995 }%
2996 \dolistloop{\@series}%
2997 \fi%
2998 }%
2999
3000
3001 %

```

XII.2 Penalties

\add@penalties \add@penalties is the last macro used by \do@line. It adds up the club, widow, and interline penalties, and puts a single penalty of the appropriate size back into the paragraph; these penalties get removed by the \vsplit operation. \displaywidowpenalty and \brokenpenalty are not restored, since we have no easy way to find out where we should insert them.

In this code, \num@lines is the number of lines in the whole paragraph, and \par@line is the line we are working on at the moment. The count \@l@tempcnta is used to calculate and accumulate the penalty; it is initially set to the value of \ballast@count, which has been worked out in \do@ballast above (VIII p. 171). Finally, the penalty is checked to see that it does not go below -10000 .

```

3002 \newcommand*{\add@penalties}{\@l@tempcnta=\ballast@count
3003     \ifnum\num@lines>\@ne
3004         \global\advance\par@line \@ne
3005         \ifnum\par@line=\@ne
3006             \advance\@l@tempcnta \clubpenalty
3007         \fi

```

```

3008 \l@dttempcntb=\par@line \advance\l@dttempcntb \@ne
3009 \ifnum\l@dttempcntb=\num@lines
3010 \advance\l@dttempcnta \widowpenalty
3011 \fi
3012 \ifnum\par@line<\num@lines
3013 \advance\l@dttempcnta \interlinepenalty
3014 \fi
3015 \fi
3016 \ifnum\l@dttempcnta=\z@
3017 \relax
3018 \else
3019 \ifnum\l@dttempcnta>-10000
3020 \penalty\l@dttempcnta
3021 \else
3022 \penalty -10000
3023 \fi
3024 \fi}
3025
3026 %

```

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

```

3027 \newcommand*{\flush@notes}{%
3028 \iftoggle{notfirstrun@jobname.\extensionchars\the\section@num}{%
3029 \@xloop%
3030 \ifx\inserts@list\empty \else%
3031 \gl@p\inserts@list\to\@insert%
3032 \@insert%
3033 \global\let\@insert=\undefined%
3034 \repeat%
3035 }{}%
3036 }%
3037
3038
3039 %

```

\@xloop `\@xloop` is a variant of the PLAIN \TeX `\loop` macro, useful when it's hard to construct a positive test using the \TeX `\if` commands—as in `\flush@notes` above. One types

`\@xloop ... \if ... \else ... \repeat`, and the action following `\else` is repeated as long as the `\if` test fails. (This macro will work wherever the `PLAIN TEX \loop` is used, too, so we could just call it `\loop`; but it seems preferable not to change the definitions of any of the standard macros.)

This variant of `\loop` was introduced by Alois Kabsch in *TUGboat* 8 (1987), pp. 184–5.

```

3040 \def\@xloop#1\repeat{%
3041   \def\body{#1\expandafter\body\fi}%
3042   \body}
3043
3044 %

```

XII.4 Text before notes

`\set@Xtxtbeforenotes` The `\set@Xtxtbeforenotes` macro resets the `Xtxtbeforenotes@⟨series⟩@typeset` boolean to false. Just before the first note of the `⟨series⟩` in a page, the `\Xtextbeforenotes` will be inserted.

```

3045 \newcommand{\set@Xtxtbeforenotes}{%
3046   \unless\ifnocritical{%
3047     \def\do##1{%
3048       \nottoggle{Xtxtbeforenotesonlyonce@##1}{%
3049         \global\togglefalse{Xtxtbeforenotes@##1@typeset}%
3050       }{}%
3051     }%
3052     \dolistloop{\@series}%
3053   \fi%
3054 }%
3055 %

```

`\set@txtbeforenotesX` The `\set@txtbeforenotesX` does the same for the `\txtbeforenotesX`.

```

3056 \newcommand{\set@txtbeforenotesX}{%
3057   \unless\ifnofamiliar{%
3058     \def\do##1{%
3059       \nottoggle{txtbeforenotesonlyonceX@##1}{%
3060         \global\togglefalse{txtbeforenotesX@##1@typeset}%
3061       }{}%
3062     }%
3063     \dolistloop{\@series}%
3064   \fi%
3065 }%
3066 %

```

`\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.

```

3067 \newcommand{\insert@Xtxtbeforenotes}[1]{%
3068 \nottoggle{Xtxtbeforenotes@#1@typeset}{%
3069 \global\toggletrue{Xtxtbeforenotes@#1@typeset}%
3070 \ifcsvoid{Xtxtbeforenotes@#1}{-}{%
3071 \ifcsstring{series@display#1}{paragraph}%
3072 {\noindent\csuse{Xtxtbeforenotes@#1}}%
3073 {\expandafter\insert\csname#1footins\endcsname%
3074 \bgroup%
3075 \noindent%
3076 \ifcsdef{\csuse{series@display#1}@begin@insert}{%
3077 \csuse{\csuse{series@display#1}@begin@insert}{#1}%
3078 }{}%
3079 \strut\csuse{Xnotefontsize@#1}%
3080 \csuse{Xtxtbeforenotes@#1}%
3081 \egroup%
3082 }%
3083 }%
3084 }%
3085 {}%
3086 }%
3087
3088 \newcommand{\insert@txtbeforenotesX}[1]{%
3089 \nottoggle{txtbeforenotesX@#1@typeset}{%
3090 \global\toggletrue{txtbeforenotesX@#1@typeset}%
3091 \ifcsvoid{txtbeforenotesX@#1}{-}{%
3092 \ifcsstring{series@displayX#1}{paragraph}%
3093 {\noindent\csuse{txtbeforenotesX@#1}}%
3094 {\expandafter\insert\csname footins#1\endcsname%
3095 \bgroup%
3096 \noindent%
3097 \ifcsdef{\csuse{series@displayX#1}@begin@insert}{%
3098 \csuse{\csuse{series@displayX#1}@begin@insert}{#1}%
3099 }{}%
3100 \strut\csuse{notefontsizeX@#1}\csuse{txtbeforenotesX@#1}%
3101 \egroup%
3102 }%
3103 }%
3104 }%
3105 {}%
3106 }%
3107 }%
3108
3109 %
3110 %

```

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@lemmfont` `\select@lemmfont` 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.

```

3111 \def\select@lemmfont#1|#2|#3|#4|#5|#6|#7|{\select@lemmfont#7|}
3112 \def\select@lemmfont#1/#2/#3/#4|%
3113   {\fontencoding{#1}\fontfamily{#2}\fontseries{#3}\fontshape{#4}%
3114   \selectfont}
3115
3116 %

```

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.

```

3117 \newcommand*\footnoteoptions@[3]{%
3118   \def\do##1{%
3119     \ifstrequal{#1}{L}{% On the left side
3120       \xright@appenditem{\noexpand\setkeys[mac]{#3footnoteoption}{\
3121         unexpanded{##1}}}{\to\inserts@list%
3122       \global\advance\insert@count \@ne% Increment the left insert
3123       counter.
3124       }%
3125       }%
3126       \xright@appenditem{\noexpand\setkeys[mac]{#3footnoteoption}{\
3127         unexpanded{##1}}}{\to\inserts@listR%
3128       \global\advance\insert@countR \@ne% Increment the right insert
3129       counter insert.
3130       }%
3131       }%
3132   \notblank{#2}{\docsvlist{#2}}}% Parsing all options
3133 }
3134 %

```

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.

```

3131 \newcommandx*{\footnotelang@lua}[1][1=L,usedefault]{%
3132   \ifstrequal{#1}{L}{%
3133     \xright@appenditem{\csxdef{footnote@luatextextdir}{\the\textdir}}{\to\
inserts@list%Know the dir of lemma
3134     \global\advance\insert@count \@ne%
3135     \xright@appenditem{\csxdef{footnote@luatexpardir}{\the\pardir}}{\to\
inserts@list%Know the dir of lemma
3136     \global\advance\insert@count \@ne%
3137   }%
3138   {%
3139     \xright@appenditem{\csxdef{footnote@luatextextdir}{\the\textdir}}{\to\
inserts@listR%Know the dir of lemma
3140     \global\advance\insert@countR \@ne%
3141     \xright@appenditem{\csxdef{footnote@luatexpardir}{\the\pardir}}{\to\
inserts@listR%Know the dir of lemma
3142     \global\advance\insert@countR \@ne%
3143   }%
3144 }
3145 %

```

`\footnotelang@poly` `\footnotelang@poly` is called to remember the information about the language of a lemma when polyglossia is used.

```

3146 \newcommandx*{\footnotelang@poly}[1][1=L,usedefault]{%
3147   \ifstrequal{#1}{L}{%
3148     \if@RTL%
3149       \xright@appenditem{\csxdef{footnote@dir}{@RTLtrue}}{\to\
inserts@list%Know the language used in the lemma
3150       \global\advance\insert@count \@ne%
3151     }%
3152     \else
3153       \xright@appenditem{\csxdef{footnote@dir}{@RTLfalse}}{\to\
inserts@list%Know the language of lemma
3154       \global\advance\insert@count \@ne%
3155     }%
3156     \fi%
3157     \xright@appenditem{\csxdef{footnote@lang}{\expandonce\language}}{\to\
inserts@list%Know the language of lemma
3158     \global\advance\insert@count \@ne%
3159   }%
3160   {%
3161     \if@RTL%
3162       \xright@appenditem{\csxdef{footnote@dir}{@RTLtrue}}{\to\
inserts@listR%Know the language of lemma
3163       \global\advance\insert@countR \@ne%
3164     }%
3165     \else
3166       \xright@appenditem{\csxdef{footnote@dir}{@RTLfalse}}{\to\
inserts@listR%Know the language of lemma

```

```

3164 \global\advance\insert@countR \@ne%
3165 \fi
3166 \xright@appenditem{{\csxdef{footnote@lang}{\expandonce\language}}}\
to\inserts@listR%Know the language of lemma
3167 \global\advance\insert@countR \@ne%
3168 }%
3169 }
3170 %

```

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

`\footplitskips` 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 \LaTeX `\floatingpenalty`.

```

3171 \newcommand*{\footplitskips}{%
3172 \interlinepenalty=\interfootnotelinepenalty
3173 \unless\ifl@dprintingpages%
3174 \floatingpenalty=\@MM%
3175 \fi%
3176 \splittopskip=\ht\strutbox \splitmaxdepth=\dp\strutbox

```

```

3177 \leftskip=\z@skip \rightskip=\z@skip}
3178
3179 %

```

\normalfootnoterule `\normalfootnoterule` is a standard footnote-rule macro, for use by a `footstart` macro: just the same as the PLAIN \TeX footnote rule.

```

3180 \let\normalfootnoterule=\footnoterule
3181 %

```

XIII.6 Footnotes arrangement

XIII.6.1 User level macro

\Xarrangement `\Xarrangement[⟨s⟩]{⟨arrangement⟩}` The command calls, for each series, a specific command which set many counters and commands in order to define specific arrangement.

```

3182 \newcommand{\Xarrangement}[2][1,usedefault]{%
3183   \def\do##1{%
3184     \csname Xarrangement@#2\endcsname{##1}%
3185   }%
3186   \ifstrempy{#1}%
3187   {%
3188     \dolistloop{\@series}%
3189   }%
3190   {
3191     \docsvlist{#1}%
3192   }%
3193 }%
3194 %

```

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\Afootnote=\normalvfootnote \let\Afootfmt=\normalfootfmt
\let\Afootstart=\normalfootstart \let\Afootgroup=\normalfootgroup
\let\Afootnoterule=\normalfootnoterule

```

(Read *The TeXbook* in order to understand what are the counter, skip and dimen associated to an insertion.)

Instead of repeating ourselves, we define a `\Xarrangement@normal` macro that makes all these assignments for us, for any given series letter. This command is called when people use `\Xarrangement[⟨series⟩]{normal}`

Now we set up the `\Xarrangement@normal` macro itself. It takes one argument: the footnote series letter.

```

3195 \newcommand*{\Xarrangement@normal}[1]{%
3196   \csgdef{series@display#1}{normal}
3197   \expandafter\let\csname #1footstart\endcsname=\normalfootstart
3198   \expandafter\let\csname v#1footnote\endcsname=\normalvfootnote
3199   \expandafter\let\csname #1footfmt\endcsname=\normalfootfmt
3200   \expandafter\let\csname #1footgroup\endcsname=\normalfootgroup
3201   \expandafter\let\csname #1footnoterule\endcsname=%
3202                                     \normalfootnoterule
3203   \count\csname #1footins\endcsname=1000
3204   \dimen\csname #1footins\endcsname=\csuse{Xmaxhnotes@#1}
3205   \skip\csname #1footins\endcsname=\csuse{Xbeforenotes@#1}%
3206   \advance\skip\csname #1footins\endcsname by\csuse{Xafterrule@#1}%
3207   %

```

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 $\mathrm{T}_{\mathrm{E}}\mathrm{X}$'s inserts.

```

3208   \csxdef{default@#1footins}{1000}%Use this to confine the notes to one
side only
3209   %

```

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).

```

3210   \ifnoledgroup@else%
3211     \expandafter\let\csname mpv#1footnote\endcsname=\mpnormalvfootnote
3212     \expandafter\let\csname mp#1footgroup\endcsname=\mpnormalfootgroup
3213     \count\csname mp#1footins\endcsname=1000
3214     \dimen\csname mp#1footins\endcsname=\csuse{Xmaxhnotes@#1}
3215     \skip\csname mp#1footins\endcsname=\csuse{Xbeforenotes@#1}%
3216     \advance\skip\csname mp#1footins\endcsname by\csuse{Xafterrule@#1}%
3217   \fi
3218 }
3219
3220 %

```

`\normalvfootnote` We now begin a series of commands that do “normal” footnote formatting: a format much like that implemented in PLAIN $\mathrm{T}_{\mathrm{E}}\mathrm{X}$, in which each footnote is a separate paragraph.

`\normalvfootnote` takes the series letter as #1 and the entire text of the footnote is #2. It does the `\insert` for this note, calling on the `\footfmt` macro for this note series to format the text of the note.

```

3221 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\normalvfootnote}[2]{%

```

```

3222 \iftoggle{Xgroupbyline@#1}{%In the case we use \Xgroupbyline, the
insertion is done later, in \add@Xgroupbyline.
3223 \prepare@Xgroupbyline{#1}{#2}{\normalvfootnote@inserted}%
3224 }{%In the case we don't use \Xgroupbyline, the insertion is made directly
3225 \X@beforeinsertion{#1}%
3226 \insert\csname #1footins\endcsname{%
3227 \X@atbegininsertion{#1}%
3228 \normalvfootnote@inserted{#1}{#2}%
3229 }%
3230 }%
3231 }%
3232 %

```

\normalvfootnote@inserted The `\normalvfootnote@inserted` macro is expanded to the content to be added to a `\insert` for normal critical footnote.

```

3233 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\
normalvfootnote@inserted}[2]{%
3234 \nottoggle{Xgroupbyline@#1}{\noindent}{\csuse{Xhooknote@#1}%
3235 \csuse{Xnotefontsize@#1}%
3236 \iftoggle{Xgroupbyline@#1}{\strut}{}%
3237 \footsplitskips
3238 \ifl@dpairing\ifl@dpadding\else%
3239 \setXnoteswidthliketwocolumns@{#1}%
3240 \fi\fi%
3241 \setXnotespositionliketwocolumns@{#1}%
3242 \spaceskip=\z@skip \xspaceskip=\z@skip%
3243 \csname #1footfmt\endcsname #2{#1}%
3244 }%
3245 %

```

```

\X@beforeinsertion 3246 \newcommand{\X@beforeinsertion}[1]{%
3247 \if@ledgroup\else%
3248 \insert@Xtxtbeforenotes{#1}%
3249 \fi%
3250 \csuse{Xbeforeinserting@#1}%
3251 }%
3252 %

```

```

\beforeinsertion@X 3253 \newcommand{\beforeinsertion@X}[1]{%
3254 \if@ledgroup\else%
3255 \insert@txtbeforenotesX{#1}%
3256 \fi%
3257 \csuse{beforeinsertingX@#1}%
3258 }%
3259 %

```

```

\X@atbegininsertion60 \newcommand{\X@atbegininsertion}[1]{%
3261   \hspace=\expandafter\dimexpr\csuse{Xwidth@#1}\relax%
3262 }%
3263 %

```

And somewhat different versions of `\normalvfootnote` and `\normalvfootnote@inserted` for minipages.

```

\mpnormalvfootnote64 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\mpnormalvfootnote}[2]{%
3265   \iftoggle{Xgroupbyline@#1}{%
3266     \prepare@Xgroupbyline{#1}{#2}{\mpnormalvfootnote@inserted}%
3267   }%
3268   {%
3269     \global\setbox\@nameuse{mp#1footins}%
3270     \vbox{%
3271       \unvbox\@nameuse{mp#1footins}%
3272       \mpnormalvfootnote@inserted{#1}{#2}%
3273     }%
3274   }%
3275 }%
3276 %
3277 %

```

```

\normalvfootnote@inserted78 \newcommand{\mpnormalvfootnote@inserted}[2]{%
3279   \noindent\csuse{Xbhooknote@#1}%
3280   \csuse{Xnotefontsize@#1}%
3281   \hspace\columnwidth%
3282   \@parboxrestore%
3283   \color@begingroup%
3284   \csname #1footfmt\endcsname #2{#1}\color@endgroup%
3285 }%
3286 %

```

`\normalfootfmt` `\normalfootfmt` is a “normal” macro to take the footnote line and page number information (see V.9 p. 113), 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.

```

3287 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\normalfootfmt}[4]{%
3288   \Xstorelineinfo{#1}{#4}%
3289   \nottoggle{Xgroupbyline@#4}{\Xledsetnormalparstuff{#4}}{}%
3290   \Xsethangindent{#4}%
3291   \nottoggle{Xgroupbyline@#4}{\rule\z@\splittopskip}{}%
3292   {\printlinefootnote{#1}{#4}}%
3293   \print@lemma{#1}{#2}{#4}%
3294   \csuse{Xwrapcontent@#4}{#3}%

```

```

3295 \nottoggle{Xgroupbyline@#4}{\strut\par}{}%
3296 }%
3297 %

```

\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`.

```

3298 \newcommand*{\normalfootstart}[1]{%
3299 %

```

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. 252. Here is part of this algorithm, when the block of notes are ready to be printed.

```

3300 \ifdimequal{0pt}{\Xprenotes@}{}%
3301 {%
3302 \iftoggle{Xprenotes@}{%
3303 \togglefalse{Xprenotes@}%
3304 \skip\csname #1footins\endcsname=%
3305 \glueexpr\csuse{Xprenotes@}+\csuse{Xafterrule@#1}\relax%
3306 }%
3307 }%
3308 }%
3309 \vskip\skip\csname #1footins\endcsname%
3310 %

```

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. 250 for the general description of the problem.

```

3311 \leftskip0pt \rightskip0pt
3312 \ifl@dpairing\else%
3313 \hsize=\old@hsize%
3314 \fi%
3315 \setXnoteswidthliketwocolumns@{#1}%

```

```

3316 \setXnotespositionliketwocolumns@{#1}%
3317 %

```

And now, print the footnote's rule to finish the footnote's introduction.

```

3318 \print@Xfootnoterule{#1}%
3319 }%
3320 %

```

\normalfootgroup \normalfootgroup is a standard footnote-grouping macro: it sends the contents of the footnote-insert box to the output page without alteration.

```

3321 \newcommand*{\normalfootgroup}[1]{%
3322 \csuse{Xbhookgroup@#1}%
3323 \unvbox\csname #1footins\endcsname%
3324 \hsize=\old@hsize%
3325 }%
3326 %
3327 %

```

\mpnormalfootgroup A somewhat different version for minipages. Note that, in this case, we do not make distinctions between the \Xfootgroup and \Xfootstarts macros.

```

3328 \unless\ifnoledgroup@
3329 \newcommand*{\mpnormalfootgroup}[1]{%
3330 \vskip\skip\@nameuse{mp#1footins}
3331 \ifl@dpairing\ifparledgroup%
3332 \leavevmode\marks\parledgroup@{begin}%
3333 \marks\parledgroup@series{#1}%
3334 \marks\parledgroup@type{Xfootnote}%
3335 \fi\fi\normalcolor%
3336 \ifparledgroup%
3337 \ifl@dpairing%
3338 \else%
3339 \setXnoteswidthliketwocolumns@{#1}%
3340 \setXnotespositionliketwocolumns@{#1}%
3341 \print@Xfootnoterule{#1}%
3342 \fi%
3343 \else%
3344 \setXnoteswidthliketwocolumns@{#1}%
3345 \setXnotespositionliketwocolumns@{#1}%
3346 \print@Xfootnoterule{#1}%
3347 \fi%
3348 \setlength{\parindent}{0pt}
3349 \csuse{Xbhookgroup@#1}%
3350 \unvbox\csname mp#1footins\endcsname}}
3351 \fi
3352 %

```

XIII.6.3 Paraphrased footnotes

The paraphrased-footnote option reformats all the footnotes of one series for a page into a single paragraph; this is especially appropriate when the notes are numerous and brief. The code is based on *The TeXbook*, pp. 398–400, with alterations for our environment. This algorithm uses a considerable amount of save-stack space: a T_EX 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 paraphrased; 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 paraphrased 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 paraphrased.

```

3353 \newcommand*{\Xarrangement@paragraph}[1]{%
3354   \csgdef{series@display#1}{paragraph}
3355   \expandafter\let\csname #1footstart\endcsname=\parafootstart
3356   \expandafter\let\csname v#1footnote\endcsname=\paravfootnote
3357   \expandafter\let\csname #1footfmt\endcsname=\parafootfmt
3358   \expandafter\let\csname #1footgroup\endcsname=\parafootgroup
3359   \count\csname #1footins\endcsname=1000
3360   \csxdef{default@#1footins}{1000}%Use this to confine the notes to one
side only
3361   \dimen\csname #1footins\endcsname=\csuse{Xmaxhnotes@#1}
3362   \skip\csname #1footins\endcsname=\csuse{Xbeforenotes@#1}%
3363   \advance\skip\csname #1footins\endcsname by\csuse{Xafterrule@#1}%
3364   \para@footsetup{#1}
3365   %

```

And the extra setup for minipages.

```

3366   \ifnoledgroup@else
3367     \expandafter\let\csname mpv#1footnote\endcsname=\mpparavfootnote
3368     \expandafter\let\csname mp#1footgroup\endcsname=\mpparafootgroup
3369     \count\csname mp#1footins\endcsname=1000
3370     \dimen\csname mp#1footins\endcsname=\csuse{Xmaxhnotes@#1}
3371     \skip\csname mp#1footins\endcsname=\csuse{Xbeforenotes@#1}%
3372     \advance\skip\csname mp#1footins\endcsname by\csuse{Xafterrule@#1}%
3373   \fi
3374 }
3375 %

```

`\footfudgefiddle` For paraphrased footnotes T_EX 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.

```

3376 \providecommand{\footfudgefiddle}{64}
3377 %

```

`\para@footsetup` `\footparagraph` calls the `\para@footsetup` macro to calculate a special fudge factor, which is the ratio of the `\baselineskip` to the `\hsize`. We assume that the proper value of `\baselineskip` for the footnotes (normally 9 pt) has been set already. The argument of the macro is again the note series letter.

Peter Wilson thinks that `\columnwidth` should be used here for \LaTeX not `\hsize`. Peter Wilson have also included `\footfudgefiddle`.

```

3378 \newcommand*{\para@footsetup}[1]{\csuse{Xbhookgroup@#1}\csuse{
Xnotesfontsize@#1}
3379 \setXnoteswidthliketwocolumns@{#1}%
3380 \ifcempty{Xwidth@#1}%
3381 {}%
3382 {\columnwidth=\expandafter\dimexpr\csuse{Xwidth@#1}\relax}%
3383 \dimen0=\baselineskip
3384 \multiply\dimen0 by 1024
3385 \divide \dimen0 by \columnwidth \multiply\dimen0 by \footfudgefiddle\
relax
3386 \csxdef{#1footfudgefactor}{%
3387 \expandafter\strip@pt\dimen0 }}
3388
3389 %

```

`\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.

```

3390 \newcommand*{\parafootstart}[1]{%
3391 \rightskip=0pt \leftskip=0pt%
3392 \Xsetparindent{#1}%
3393 \ifdimequal{0pt}{\Xprenotes@}{}%
3394 {%
3395 \iftoggle{Xprenotes@}{%
3396 \togglefalse{Xprenotes@}%
3397 \skip\csname #1footins\endcsname=%
3398 \glueexpr\csuse{Xprenotes@}+\csuse{Xafterterrule@#1}\relax%
3399 }%
3400 }%
3401 }%
3402 \vskip\skip\csname #1footins\endcsname%
3403 \setXnoteswidthliketwocolumns@{#1}%
3404 \setXnotespositionliketwocolumns@{#1}%
3405 \print@Xfootnoterule{#1}%
3406 \let\old@bidi@RTL@everypar\bidi@RTL@everypar%
3407 \let\bidi@RTL@everypar\@empty%
3408 \noindent\leavevmode%

```

```

3409 \let\bidir@RTL@everypar\old@bidir@RTL@everypar%
3410 }%
3411 %

```

`\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

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.

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. 196 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.

```

3412 \newcommand*{\paravfootnote}[2]{%
3413   \csuse{Xbeforeinserting@#1}%
3414   \insert\csname #1footins\endcsname
3415   \bgroup
3416     \csuse{Xnotefontsize@#1}
3417     \footplitskips
3418     \setbox0=\vbox{\hsize=\maxdimen%
3419       \let\bidir@RTL@everypar\@empty%
3420       \insert@Xtxtbeforenotes{#1}%
3421       \noindent\csuse{Xhooknote@#1}%
3422       \csname #1footfmt\endcsname #2{#1}}%
3423     \setbox0=\hbox{\Xunvxh{0}{#1}}%
3424     \dp0=0pt
3425     \ht0=\csname #1footfudgefactor\endcsname\wd0
3426   %

```

Here we produce the contents of the footnote from box 0, and add a penalty of 0 between boxes in this insert.

```

3427   \if@RTL\noindent \leavevmode\fi\box0%
3428   \penalty0
3429 \egroup}
3430
3431 %

```

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.

```

3432 \newcommand*{\mpparavfootnote}[2]{%
3433   \global\setbox\@nameuse{mp#1footins}\vbox{%
3434     \unvbox\@nameuse{mp#1footins}%
3435     \csuse{Xnotefontsize@#1}

```

```

3436 \footsplitskips
3437 \setbox0=\vbox{\hsize=\maxdimen%
3438   \let\bidir@RTL@everypar\@empty%
3439   \insert@Xtxtbeforenotes{#1}%
3440   \noindent\color@begingroup%
3441   \csuse{Xbhooknote@#1}%
3442   \csname #1footfmt\endcsname #2{#1}\color@endgroup}%
3443 \setbox0=\hbox{\Xunvxh{0}{#1}}%
3444 \dp0=\z@
3445 \ht0=\csname #1footfudgefactor\endcsname\wd0
3446 \box0
3447 \penalty0
3448 }}
3449
3450 %

```

\Xunvxh Here is (modified) Michael’s definition of `\unvxh`, used above. Michael’s macro also takes care to remove some unwanted penalties and glue that \TeX automatically attaches to the end of paragraphs. When \TeX finishes a paragraph, it throws away any remaining glue, and then tacks on the following items: a `\penalty` of 10000, a `\parfillskip` and a `\rightskip` (*The TeXbook*, pp. 99–100). `\unvxh` cancels these unwanted paragraph-final items using `\unskip` and `\unpenalty`.

```

3451 \newcommand*{\Xunvxh}[2]{%
3452   \setbox0=\vbox{\unvbox#1%
3453     \global\setbox1=\lastbox}%
3454   \unhbox1
3455   \unskip           % remove \rightskip,
3456   \unskip           % remove \parfillskip,
3457   \unpenalty        % remove \penalty of 10000,
3458   \hskip\csuse{Xafternote@#2}\relax}% add the glue to go between the notes
3459
3460 %

```

\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).

```

3461 \newcommand*{\parafootfmt}[4]{%
3462   \Xstorelineinfo{#1}{#4}%
3463   \Xinsertparafootsep{#4}%
3464   \ledsetnormalparstuff@common%
3465   \printlinefootnote{#1}{#4}%
3466   \print@lemma{#1}{#2}{#4}%
3467   \csuse{Xwrapcontent@#4}{#3}%
3468   \penalty-10 }
3469 %

```

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.

```

3470 \newcommand*{\parafootgroup}[1]{%
3471   \hsize=\expandafter\dimexpr\csuse{Xwidth@#1}\relax%
3472   \unvbox\csname #1footins\endcsname
3473   \ifcsstring{Xragged@#1}{L}{\RaggedLeft}{}%
3474   \ifcsstring{Xragged@#1}{R}{\RaggedRight}{}%
3475   \makehboxofhboxes
3476   \setbox0=\hbox{\unhbox0 \removehboxes}%
3477   \csuse{Xhookgroup@#1}%
3478   \csuse{Xnotefontsize@#1}%
3479   \unhbox0\par%
3480   \global\hsize=\old@hsize%
3481 }%
3482
3483 %

```

`\mpparafootgroup` The minipage version.

```

3484 \newcommand*{\mpparafootgroup}[1]{%
3485   \setXnoteswidthliketwocolumns@{#1}%
3486   \vskip\skip\@nameuse{mp#1footins}
3487   \ifl@dpairing\ifparledgroup%
3488     \leavevmode\marks\parledgroup@{begin}%
3489     \marks\parledgroup@series{#1}%
3490     \marks\parledgroup@type{Xfootnote}%
3491   \fi\fi\normalcolor
3492   \ifparledgroup%
3493     \ifl@dpairing%
3494     \else%
3495       \setXnoteswidthliketwocolumns@{#1}%
3496       \setXnotespositionliketwocolumns@{#1}%
3497       \print@Xfootnoterule{#1}%
3498     \fi%
3499   \else%
3500     \setXnoteswidthliketwocolumns@{#1}%
3501     \setXnotespositionliketwocolumns@{#1}%
3502     \print@Xfootnoterule{#1}%
3503   \fi%
3504   \unvbox\csname mp#1footins\endcsname
3505   \ifcsstring{Xragged@#1}{L}{\RaggedLeft}{}%

```

```

3506 \ifcsstring{Xragged@#1}{R}{\RaggedRight}{}%
3507 \makehboxofhboxes
3508 \setbox0=\hbox{\unhbox0 \removehboxes}%
3509 \csuse{Xhookgroup@#1}%
3510 \csuse{Xnotefontsize@#1}%
3511 \Xsetparindent{#1}%
3512 \unhbox0\par}}
3513
3514 %

```

And finally, the two macros which are required to transform the long horizontal box stored in the insert' box to a printable text.

```

\makehboxofhboxes15 \newcommand*\makehboxofhboxes{\setbox0=\hbox{}}%
\removehboxes16 \loop
3517 \unpenalty
3518 \setbox2=\lastbox
3519 \ifhbox2
3520 \setbox0=\hbox{\box2\unhbox0}%
3521 \repeat}
3522
3523 \newcommand*\removehboxes{\setbox0=\lastbox
3524 \ifhbox0{\removehboxes}\unhbox0 \fi}
3525
3526 %

```

Insertion of the footnotes separator The command `\Xinsertparafootsep{<series>}` must be called at the beginning of `\parafootfm`.

```

\prevpage@num27 \newcommand{\Xinsertparafootsep}[1]{%
\Xinsertparafootsep28 \ifledRcol%
3529 \ifnumequal{\csuse{#1prevpage@numR}}{\page@numR}%
3530 {\ifcsdef{prevline#1}% Be sur \prevline#1 exists.
3531 {\ifcsequal{prevline#1}{lineinfo@}%
3532 {\ifcsequal{Xsymlinenum@#1}{\csuse{Xparafootsep@#1}}{}}%
3533 {\csuse{Xparafootsep@#1}}%
3534 }%
3535 {\csuse{Xparafootsep@#1}}%
3536 }%
3537 {}%
3538 \global\csname #1prevpage@numR\endcsname=\page@numR%
3539 \else%
3540 \ifnumequal{\csuse{#1prevpage@num}}{\page@num}%
3541 {\ifcsdef{prevline#1}% Be sur \prevline#1 exists.
3542 {\ifcsequal{prevline#1}{lineinfo@}%
3543 {\ifcsequal{Xsymlinenum@#1}{\csuse{Xparafootsep@#1}}{}}%
3544 {\csuse{Xparafootsep@#1}}%
3545 }%

```

```

3546     {\csuse{Xparafootsep@#1}}}%
3547   }%
3548   {}%
3549   \global\csname #1prevpage@num\endcsname=\page@num%
3550   \fi%
3551 }
3552 %

```

XIII.6.4 Columnar footnotes

Common tools

`\rigidbalance` We will now define macros for three-column notes and two-column notes. Both sets of macros will use `\rigidbalance`, which splits a box (#1) into a number (#2) of columns, each with a space (#3) between the top baseline and the top of the `\vbox`. The `\Xrigidbalance` `\dosplits` `\splitoff` `\h` `\k` `\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`.

```

3553 \newcount\@k \newdimen\@h
3554 \newcommand*\Xrigidbalance}[3]{%
3555   \hsize=\expandafter\dimexpr\csuse{Xwidth@\@currentseries}\relax%
3556   \rigidbalance{#1}{#2}{#3}%
3557 }%
3558
3559 \newcommand*\rigidbalanceX}[3]{%
3560   \hsize=\expandafter\dimexpr\csuse{widthX@\@currentseries}\relax%
3561   \rigidbalance{#1}{#2}{#3}%
3562 }%
3563
3564 \newcommand*\rigidbalance}[3]{%
3565   \setbox0=\box#1 \@k=#2 \@h=#3%
3566   \@@line{\splittopskip=\@h \vbadness=\@M \hfilneg
3567   \valign{##\vfil\cr\dosplits}}}%
3568
3569 \newcommand*\dosplits{\ifnum\@k>0 \noalign{\hfil}\splitoff
3570 \global\advance\@k-1\cr\dosplits\fi}
3571
3572 \newcommand*\splitoff{\dimen0=\ht0
3573 \divide\dimen0 by\@k \advance\dimen0 by\@h
3574 \setbox2 \vsplit0 to \dimen0
3575 \unvbox2 }
3576
3577 %

```

Three columns

```
\Xarrangement@threecol \newcommand*{\Xarrangement@threecol}[1]{%
3579 \csgdef{series@display#1}{threecol}
3580 \expandafter\let\csname v#1footnote\endcsname=\threecolvfootnote
3581 \expandafter\let\csname #1footfmt\endcsname=\threecolfootfmt
3582 \expandafter\let\csname #1footgroup\endcsname=\threecolfootgroup
3583 \dimen\csname #1footins\endcsname=\csuse{Xmaxhnotes@#1}%
3584 \skip\csname #1footins\endcsname=\csuse{Xbeforenotes@#1}%
3585 \advance\skip\csname #1footins\endcsname by\csuse{Xafterrule@#1}%
3586 \threecolfootsetup{#1}
3587 %
```

The additional setup for minipages.

```
3588 \ifnoledgroup@else
3589 \expandafter\let\csname mpv#1footnote\endcsname=\mpnormalvfootnote
3590 \expandafter\let\csname mp#1footgroup\endcsname=\mpthreecolfootgroup
3591 \skip\csname mp#1footins\endcsname=\csuse{Xbeforenotes@#1}%
3592 \advance\skip\csname mp#1footins\endcsname by\csuse{Xafterrule@#1}%
3593 \mpthreecolfootsetup{#1}
3594 \fi
3595 }
3596
3597 %
```

The `\footstart` and `\footnoterule` macros for these notes assume the normal values (XIII.6.2 p. 196 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.

```
3598 \newcommand*{\threecolfootsetup}[1]{%
3599 \count\csname #1footins\endcsname 333
3600 \csxdef{default@#1footins}{333}%Use this to confine the notes to one
side only
3601 \multiply\dimen\csname #1footins\endcsname \thr@@
3602 %
```

`\mpthreecolfootsetup` The setup for minipages.

```

3603 \newcommand*\mpthreecolfootsetup}[1]{%
3604   \count\csname mp#1footins\endcsname 333
3605   \multiply\dimen\csname mp#1footins\endcsname \thr@@}
3606
3607 %

```

`\threecolvfootnote` `\threecolvfootnote` This is the `\vfootnote` command for three-column notes. However, most of the code is deported on `\threecolvfootnote@inserted`. The call to `\Xnotefontsize@<s>` ensures that the `\splittopskip` and `\splitmaxdepth` take their values from the right `\strutbox`: the one used in a footnotes. Note especially the importance of temporarily reducing the `\hsize` to 0.3 of its normal value. This determines the widths of the individual columns. So if the normal `\hsize` is (say) 10 cm, then each column will be $0.3 \times 10 = 3$ cm wide, leaving a gap of 1 cm spread equally between columns (i.e., .5 cm between each).

The arguments are #1 the note series letter and #1 the full text of the note (including numbers, lemma and text).

```

3608 \notbool{parapparatus@}\newcommand*\newcommand{\threecolvfootnote}[2]{%
3609   \iftoggle{Xgroupbyline@#1}{%
3610     \prepare@Xgroupbyline{#1}{#2}{\threecolvfootnote@inserted}%
3611   }%
3612   {%
3613     \let\bidir@RTL@everypar\relax%
3614     \X@beforeinsertion{#1}%
3615     \insert\csname #1footins\endcsname{%
3616       \threecolvfootnote@inserted{#1}{#2}%
3617     }%
3618   }%
3619 }%
3620 %

```

```

\threecolvfootnote@inserted 21 \notbool{parapparatus@}\newcommand*\newcommand{\
\threecolvfootnote@inserted}[2]{%
3622   \hsize=\expandafter\dimexpr\csuse{Xwidth@#1}\relax%
3623   \noindent\csuse{Xhooknote@#1}%
3624   \csuse{Xnotefontsize@#1}%
3625   \footssplitsskip%
3626   \csname #1footfmt\endcsname #2{#1}%
3627 }%
3628 %

```

`\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.

```

3629 \notbool{parapparatus@}\newcommand*\newcommand{\threecolfootfmt}[4]{%

```

```

3630 \Xstorelineinfo{#1}{#4}%
3631 \threecol@begin@insert{#4}%
3632 \hspace{\parindent}%
3633 \printlinefootnote{#1}{#4}%
3634 \print@lemma{#1}{#2}{#4}%
3635 \csuse{Xwrapcontent@#4}{#3}%
3636 \nottoggle{Xgroupbyline@#4}%
3637   {\strut\par\allowbreak}%
3638   {}%
3639 }%
3640 %

```

`\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`.

```

3641 \newcommand{\threecol@begin@insert}[1]{%
3642   \normal@pars%
3643   \nottoggle{Xgroupbyline@#1}%
3644   {\hsize \csuse{Xhsizethreecol@#1}}%
3645   {}%
3646   \Xsetparindent{#1}%
3647   \tolerance=5000%
3648   \Xsethangindent{#1}%
3649   \@tempdima=\parindent%
3650   \csuse{Xcolalign@#1}%
3651   \parindent=\@tempdima%
3652   \strut%
3653 }%
3654 %

```

`\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.

```

3655 \newcommand*{\threecolfootgroup}[1]{%
3656   \begin@group%
3657   \csuse{Xbhookgroup@#1}%
3658   \csuse{Xnotefontsize@#1}%
3659   \par%
3660   \splittopskip=\ht\strutbox%
3661   \expandafter%
3662   \Xrigidbalance\csname #1footins\endcsname \thr@@ \splittopskip%

```



```

3663 \endgroup%
3664 }%
3665 %

```

\mpthreecolfootgroup The setup for minipages.

```

3666 \newcommand*{\mpthreecolfootgroup}[1]{%
3667 \vskip\skip\@nameuse{mp#1footins}
3668 \ifl@dpairing\ifparledgroup%
3669 \leavevmode\marks\parledgroup@{begin}%
3670 \marks\parledgroup@series{#1}%
3671 \marks\parledgroup@type{Xfootnote}%
3672 \fi\fi\normalcolor
3673 \ifparledgroup%
3674 \ifl@dpairing%
3675 \else%
3676 \setXnoteswidthliketwocolumns@{#1}%
3677 \setXnotespositionliketwocolumns@{#1}%
3678 \print@Xfootnoterule{#1}%
3679 \fi%
3680 \else%
3681 \setXnoteswidthliketwocolumns@{#1}%
3682 \setXnotespositionliketwocolumns@{#1}%
3683 \print@Xfootnoterule{#1}%
3684 \fi%
3685 \csuse{Xhookgroup@#1}\par%
3686 \splittopskip=\ht\strutbox
3687 \expandafter
3688 \Xrigidbalance\csname mp#1footins\endcsname \thr@@ \splittopskip}}
3689 %
3690 %

```

Two columns

```

\Xarrangement@twocol 3691 \newcommand*{\Xarrangement@twocol}[1]{%
3692 \csgdef{series@display#1}{twocol}
3693 \expandafter\let\csname v#1footnote\endcsname=\twocolvfootnote
3694 \expandafter\let\csname #1footfmt\endcsname=\twocolfootfmt
3695 \expandafter\let\csname #1footgroup\endcsname=\twocolfootgroup
3696 \dimen\csname #1footins\endcsname=\csuse{Xmaxhnotes@#1}%
3697 \skip\csname #1footins\endcsname=\csuse{Xbeforenotes@#1}%
3698 \advance\skip\csname #1footins\endcsname by\csuse{Xafterrule@#1}%
3699 \twocolfootsetup{#1}
3700 %

```

The additional setup for minipages.

```

3701 \ifnoledgroup\else
3702 \expandafter\let\csname mpv#1footnote\endcsname=\mpnormalvfootnote
3703 \expandafter\let\csname mp#1footgroup\endcsname=\mptwocolfootgroup

```

```

3704 \skip\csname mp#1footins\endcsname=\csuse{Xbeforenotes@#1}%
3705 \advance\skip\csname mp#1footins\endcsname by\csuse{Xafterrule@#1}%
3706 \mptwocolfootsetup{#1}
3707 \fi
3708 }
3709
3710 %

```

`\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 `\hsz`.

```

\twocolvfootnote
\twocolvfootnote@inserted
\twocolfootfmt
\twocolfootgroup
3711 \newcommand*{\twocolfootsetup}[1]{%
3712 \count\csname #1footins\endcsname 500
3713 \csxdef{default@#1footins}{500}%
3714 \multiply\dimen\csname #1footins\endcsname \tw@}
3715 %

3716 \notbool{parapparatus@}{\newcommand*{\newcommand}{\twocolvfootnote}[2]{%
3717 \iftoggle{Xgroupbyline@#1}{%
3718 \prepare@Xgroupbyline{#1}{#2}{\twocolvfootnote@inserted}%
3719 }{%
3720 \let\bid@RTL@everypar\relax%
3721 \X@beforeinsertion{#1}%
3722 \insert\csname #1footins\endcsname{%
3723 \twocolvfootnote@inserted{#1}{#2}%
3724 }%
3725 }%
3726 }%
3727 \notbool{parapparatus@}{\newcommand*{\newcommand}{\twocolvfootnote@inserted}[2]{%
3728 \hsz=\expandafter\dimexpr\csuse{Xwidth@#1}\relax%
3729 \noindent\csuse{Xhooknote@#1}%
3730 \csuse{Xnotefontsize@#1}%
3731 \footssplitsskip%
3732 \csname #1footfmt\endcsname #2{#1}%
3733 }%
3734 %

```

```

3735 \notbool{parapparatus@}{\newcommand*{\newcommand}{\twocolfootfmt}[4]{% 4th
arg is optional, for backward compatibility
3736 \Xstorelineinfo{#1}{#4}%
3737 \twocol@begin@insert{#4}%
3738 \hspace{\parindent}%
3739 \printlinefootnote{#1}{#4}%
3740 \print@lemma{#1}{#2}{#4}%
3741 \csuse{Xwrapcontent@#4}{#3}%
3742 \nottoggle{Xgroupbyline@#4}%
3743 {\strut\par\allowbreak}%
3744 }%

```

```

3745 }%
3746 %

3747 \newcommand{\twocol@begin@insert}[1]{%
3748   \normal@pars%
3749   \nottoggle{Xgroupbyline@#1}%
3750   {\hsize \csuse{Xhsizetwocol@#1}}%
3751   {}%
3752   \Xsetparindent{#1}%
3753   \tolerance=5000%
3754   \Xsethangindent{#1}%
3755   \@tempdima=\parindent%
3756   \csuse{Xcolalign@#1}%
3757   \parindent=\@tempdima%
3758   \strut%
3759 }%
3760
3761 \newcommand*{\twocolfootgroup}[1]{%
3762   \begingroup%
3763   \csuse{Xbhookgroup@#1}%
3764   \csuse{Xnotefontsize@#1}%
3765   \par%
3766   \splittopskip=\ht\strutbox%
3767   \expandafter%
3768   \Xrigidbalance\csname #1footins\endcsname \tw@ \splittopskip%
3769   \endgroup%
3770 }%
3771
3772 %

```

`\mptwocolfootsetup` The versions for minipages.

```

\mptwocolfootgroup
3773 \newcommand*{\mptwocolfootsetup}[1]{%
3774   \count\csname mp#1footins\endcsname 500
3775   \multiply\dimen\csname mp#1footins\endcsname \tw@}
3776 %

3777 \newcommand*{\mptwocolfootgroup}[1]{%
3778   \vskip\skip\@nameuse{mp#1footins}
3779   \ifl@dpairing\ifparledgroup%
3780     \leavevmode\marks\parledgroup@{begin}%
3781     \marks\parledgroup@series{#1}%
3782     \marks\parledgroup@type{Xfootnote}%
3783   \fi\fi\normalcolor
3784   \ifparledgroup%
3785     \ifl@dpairing%
3786     \else%
3787       \setXnoteswidthliketwocolumns@{#1}%
3788       \setXnotespositionliketwocolumns@{#1}%
3789       \print@Xfootnoterule{#1}%

```

```

3790 \fi%
3791 \else%
3792 \setXnoteswidthliketwocolumns@{#1}%
3793 \setXnotespositionliketwocolumns@{#1}%
3794 \print@Xfootnoterule{#1}%
3795 \fi%
3796 \csuse{Xbhookgroup@#1}\par%
3797 \splittopskip=\ht\strutbox
3798 \expandafter
3799 \Xrigidbalance\csname mp#1footins\endcsname \tw@ \splittopskip}}
3800
3801 %

```

XIII.7 Footnote paragraph indent

\Xsetparindent These two commands set the paragraph indentation of the footnotes, depending on the settings of the user.

\setparindentX

```

3802 \newcommand{\Xsetparindent}[1]{%
3803 \nottoggle{Xparindent@#1}{%
3804 \parindent=\z@%
3805 }%
3806 {%
3807 \ifdef{\parindent@beforestanza}{\parindent=\parindent@beforestanza}{}%
3808 }%
3809 }%
3810 \newcommand{\setparindentX}[1]{%
3811 \nottoggle{parindentX@#1}{%
3812 \parindent=\z@%
3813 }%
3814 {%
3815 \ifdef{\parindent@beforestanza}{\parindent=\parindent@beforestanza}{}%
3816 }%
3817 }%
3818 %

```

XIII.8 Footnote hanging indent

\Xsethangindent \Xsethangindent sets the hangindent for a critical footnote, while \sethangindentX does it for a familiar footnote.

\sethangindentX

```

3819 \newcommand{\Xsethangindent}[1]{%
3820 \if@RTL%
3821 \hangindent=-\csuse{Xhangindent@#1}%
3822 \everypar{\hangindent=-\csuse{Xhangindent@#1}}%
3823 \else%
3824 \hangindent=\csuse{Xhangindent@#1}%
3825 \everypar{\hangindent=\csuse{Xhangindent@#1}}%
3826 \fi%

```

```

3827 }%
3828 \newcommand{\sethangindentX}[1]{%
3829   \if@RTL%
3830     \hangindent=-\csuse{hangindentX@#1}%
3831     \everypar{\hangindent=-\csuse{hangindentX@#1}}%
3832   \else%
3833     \hangindent=\csuse{hangindentX@#1}%
3834     \everypar{\hangindent=\csuse{hangindentX@#1}}%
3835   \fi%
3836 }%
3837 %

```

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 `\Xlinrangeseparator`, `\Xendlinrangeseparator`, `\Xsublinesep`, `\Xendsublinesep` and `\Xlemmaseparator` macros.

With `polyglossia`, each critical note has a `\footnote@lang` which shows the language of the lemma, and which can be used to switch the bracket from right to left.

```

3838 \def\endashchar{\textnormal{--}}
3839
3840 \newcommand*{\fullstop}{\textnormal{.}}
3841 \def\Xsublinesep@side{\fullstop}
3842
3843 \newcommand*{\rbracket}{\textnormal{%
3844   \csuse{text\csuse{footnote@lang}}{%
3845     \ifluatex%
3846       \ifdefstring{\footnote@luatextextdir}{TRT}{\thinspace}{\thinspace}
3847     }%
3848   \else%

```

```

3848         \thinspace]%
3849         \fi}%
3850     }%
3851 }
3852
3853 %

```

XIII.9.2 Pstart number in footnote

\printpstart The `\printpstart` macro prints the pstart number for a note.

```

3854 \newcommand{\printpstart}[0]{%
3855     \ifboolexpr{bool{l@dpairing} or bool{l@dprintingpages} or bool{
3856         l@dprintingcolumns}}{%
3857         \ifledRcol%
3858             \thepstartR%
3859         \else%
3860             \thepstartL%
3861         \fi%
3862     }{%
3863         \thepstart%
3864     }%
3865 }

```

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).

```

3866 %
3867 \newcommand{\print@lemma}[3]{%
3868     \bgroup%
3869     \nottoggle{Xlemmadisablefontselection@#3}%
3870     {\select@lemmafnt#1|}%
3871     }%
3872     \bgroup%
3873     \csuse{Xlemmafnt@#3}%Deprecated
3874     \csuse{Xwraplemma@#3}{#2}%
3875     \egroup%
3876     \egroup%
3877     \iftoggle{nosep@}{%
3878         \hskip\csuse{Xinplaceoflemmaseparator@#3}%
3879         \relax%
3880     }%
3881     {\ifcempty{Xlemmaseparator@#3}%
3882         {%
3883             \hskip\csuse{Xinplaceoflemmaseparator@#3}%
3884             \relax%

```

```

3885     }%
3886     {%
3887         \nobreak%
3888         \hskip\csuse{Xbeforelemmaseparator@#3}%
3889         \csuse{Xlemmaseparator@#3}%
3890         \hskip\csuse{Xafterlemmaseparator@#3}%
3891         \relax%
3892     }%
3893 }%
3894 }%
3895 %

```

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.

```

3896 \newcommand{\Xstorelineinfo}[2]{%
3897     \l@dp@rsefootspec#1|}%
3898     \iftoggle{Xnumberonlyfirstintwolines@#2}{%
3899         \xdef\lineinfo@{\l@dparsedstartline - \l@dparsedstartsub - \
3900         \l@dparsedendline - \l@dparsedendsub}%
3901     }%
3902     {%
3903         \xdef\lineinfo@{\l@dparsedstartline - \l@dparsedstartsub}%
3904     }%
3905 }%

```

\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`.

```

3906 \newcommand{\printlinefootnote}[2]{%
3907     \xdef\@currentseries{#2}%
3908     \iftoggle{nonum@}{%Try if the line number must printed for this specific
3909     not (by default, yes)
3910         \hspace{\csuse{Xinplaceofnumber@#2}}}%
3911     }%
3912     {%
3913         \iftoggle{Xnonumber@#2}%Try if the line number must printed (by
3914         default, yes)
3915         {%
3916             \hspace{\csuse{Xinplaceofnumber@#2}}}%
3917     }%

```

```

3917         {%
3918         {\iftoggle{Xnumberonlyfirstinline@#2}% If for this series the
line number must be printed only in the first time.
3919         {%
3920         \ifcsdef{prevline#2}%
3921         {%Be sure the \prevline exists.
3922         \ifcsequal{prevline#2}{\lineinfo@}%Try it
3923         {%
3924         \ifcsequal{Xsymlinenum@#2}%
3925         {%
3926         \hspace{\csuse{Xinplaceofnumber@#2}}%
3927         }%
3928         {\printsymlinefootnotearea{#2}}%
3929         }%
3930         {%
3931         \printlinefootnotearea{#1}{#2}%
3932         }%
3933         }%
3934         {%
3935         \printlinefootnotearea{#1}{#2}%
3936         }%
3937         }%
3938         {%
3939         \printlinefootnotearea{#1}{#2}%
3940         }%
3941         \csxdef{prevline#2}{\lineinfo@}%
3942         }%
3943     }%
3944 }%
3945 }%
3946 }
3947 %

```

\printsymlinefootnotearea This macro prints the space before the line symbol, changes the font, when prints the line symbol and the space after it.

```

3948 \newcommand{\printsymlinefootnotearea}[1]{%
3949 \hspace{\csuse{Xbeforesymlinenum@#1}}%
3950 \csuse{Xnotenumfont@#1}%
3951 \ifdimequal{\csuse{Xboxsymlinenum@#1}}{\z@}%
3952 {\csuse{Xsymlinenum@#1}}%
3953 {\hbox to \csuse{Xboxsymlinenum@#1}%
3954 {\csuse{Xsymlinenum@#1}\hfill}%
3955 }%
3956 \hspace{\csuse{Xaftersymlinenum@#1}}%
3957 }%
3958 %

```

\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.)

```

3959 \newcommand{\printlinefootnotearea}[2]{%
3960   \printXbeforenumber{#2}%
3961   \csuse{Xnotenumfont@#2}%
3962   \boxfootnotenumbers{#1}{#2}%
3963   \printXafternumber{#2}%
3964 }%
3965 %

```

\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.

```

3966 \newcommand{\boxfootnotenumbers}[2]{%
3967   \ifdimequal{\csuse{Xboxlinenum@#2}}{0pt}{%
3968     \printlinefootnotenumbers{#1}{#2}%
3969   }%
3970   {%
3971     \hbox to \csuse{Xboxlinenum@#2}%
3972     {%
3973       \IfSubStr{RC}{\csuse{Xboxlinenumalign@#2}}{\hfill}{}%
3974       \printlinefootnotenumbers{#1}{#2}%
3975       \IfSubStr{LC}{\csuse{Xboxlinenumalign@#2}}{\hfill}{}%
3976     }%
3977   }%
3978 }%
3979 %

```

\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.

```

3980 \newcommand{\printlinefootnotenumbers}[2]{%
3981   \get@pstartcounter%
3982   \ifboolexpr{%
3983     (togl{Xpstart@#2} and bool{numberpstart})%
3984     or togl{Xpstarteverytime@#2}%
3985     and not (%
3986       togl{Xpstartonlyfirst@#2}%
3987       and test {\ifnumequal{\csuse{Xprevpstart@#2}}{\current@pstart}}%
3988     )%
3989   }%
3990   {%
3991     \printpstart%
3992     \csuse{Xpstartseparator@#2}%
3993   }{%
3994     \global\cslet{Xprevpstart@#2}{\current@pstart}%
3995     \get@stanzacounter%

```

```

3996 \ifboolexpr{%
3997     togl{Xstanza@#2}%
3998     and bool {numberstanza}%
3999     and not (%
4000         togl{Xpstartonlyfirst@#2}%
4001         and test {\ifnumequal{\csuse{Xprevstanza@#2}}{\current@stanza}}}%
4002     )%
4003 }{%
4004     \printstanza%
4005     \csuse{Xstanzaseparator@#2}%
4006 }{%
4007     \global\cslet{Xprevstanza@#2}{\current@stanza}%
4008     \iftoggle{Xonlypstart@#2}{%
4009         \csuse{Xtxtbeforenumber@#2}%
4010         \printlines#1|\ifledRcol@{\@Rlineflag\fi}%
4011     }%
4012 %

```

`\get@pstartcounter` A single macro to get the current pstart counter.

```

4013 \newcommand{\get@pstartcounter}[0]{%
4014     \ifboolexpr{bool{!@dprintingpages} or bool{!@dprintingcolumns}}%
4015     {%
4016         \ifledRcol%
4017             \global\numdef\current@pstart{\the\c@pstartR}%
4018         \else%
4019             \global\numdef\current@pstart{\the\c@pstartL}%
4020         \fi%
4021     }%
4022     {%
4023         \global\numdef\current@pstart{\the\c@pstart}%
4024     }%
4025 }%
4026 %

```

`\get@stanzacounter` The same to get the current stanza counter.

```

4027 \newcommand{\get@stanzacounter}[0]{%
4028     \ifboolexpr{bool{!@dprintingpages} or bool{!@dprintingcolumns}}%
4029     {%
4030         \ifledRcol%
4031             \global\numdef\current@stanza{\the\c@stanzaR}%
4032         \else%
4033             \global\numdef\current@stanza{\the\c@stanzaL}%
4034         \fi%
4035     }%
4036     {%
4037         \global\numdef\current@stanza{\the\c@stanza}%
4038     }%
4039 }%

```

```
4040 %
```

\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.)

```
4041 \newcommand{\printXbeforenumber}[1]{%
4042   \hspace{\csuse{Xbeforenumber@#1}}%
4043 }%
4044 %
```

\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

```
4045 \newcommand{\printXafternumber}[1]{%
4046   \iftoggle{Xnonbreakableafternumber@#1}{\nobreak}{}%
4047   \hspace{\csuse{Xafternumber@#1}}%
4048 }%
4049 %
```

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. 113: 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@pnun 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.

Maieul Rouquette has added:

- \ifl@d@Xtwolines and \ifl@d@Xmorethantwolines to print a symbol which stands for “and subsequent” when there are two, three or more lines;
- \@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@pnum50 \newif\ifl@d@pnum
\ifl@d@ssub51 \newif\ifl@d@ssub
\ifl@d@elin52 \newif\ifl@d@elin
\ifl@d@esl53 \newif\ifl@d@esl
\ifl@d@dash54 \newif\ifl@d@dash
\ifl@d@Xtwolines55 \newif\ifl@d@Xtwolines%
\ifl@d@Xmorethantwolines56 \newif\ifl@d@Xmorethantwolines%
\ifl@d@Xlinenumber57 \let\@annot@start@print\relax%
\@annot@start@print58 \let\@annot@end@print\relax%
\@annot@end@print59 \newif\ifl@d@Xlinenumber%
\@annot@end@print60 %

```

```

\l@dp@rsefootsspec \l@dp@rsefootsspec{<spec>}{<lemma>}{<text>} parses a footnote specification. <lemma>
\l@dp@rsefootsspec and <text> are the lemma and text respectively. <spec> is the line and page num-
\l@dp@rsefootsspec ber and lemma font specifier in \l@d@nums style format. The real work is done by
\l@dp@rsefootsspec \l@dp@rsefootsspec which defines macros holding the numeric values. In many cases,
\l@dp@rsefootsspec this last command is called directly. Just a reminder of the arguments:
\l@dp@rsefootsspec \printlines #1 | #2 | #3 | #4 | #5 | #6 | #7
\l@dp@rsefootsspec \printlines start-page | line | subline | end-page | line | subline | fontflag
\l@dp@rsefootsspec
4061 \newcommand*\l@dp@rsefootsspec}[3]{\l@dp@rsefootsspec#1|}
4062 \def\l@dp@rsefootsspec#1|#2|#3|#4|#5|#6|#7|{%
4063 \gdef\l@dp@rsefootsspec#1|#2|{%
4064 \gdef\l@dp@rsefootsspec#1|#2|{%
4065 \gdef\l@dp@rsefootsspec#1|#2|{%
4066 \gdef\l@dp@rsefootsspec#1|#2|{%
4067 \gdef\l@dp@rsefootsspec#1|#2|{%
4068 \gdef\l@dp@rsefootsspec#1|#2|{%
4069 }
4070 %

```

Initialise the several number value macros.

```

4071 \def\l@dp@rsefootsspec#1|{%
4072 \def\l@dp@rsefootsspec#1|{%
4073 \def\l@dp@rsefootsspec#1|{%
4074 \def\l@dp@rsefootsspec#1|{%
4075 \def\l@dp@rsefootsspec#1|{%
4076 \def\l@dp@rsefootsspec#1|{%
4077
4078 %

```

\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`.

```

4079 \newcommand*\setprintlines}[6]{%
4080 \let\@annot@start@print\relax%
4081 \let\@annot@end@print\relax%
4082 \l@d@pnumfalse%
4083 \l@d@dashfalse%

```

```

4084 \l@d@elinfalse%
4085 %

```

Should we print the line number? It depends on hooks and annotations.

```

4086 \l@d@Xlinenumbertrue%
4087 \ifboolexpr{togl{Xnolinenumber@\@currentseries}%
4088   or (%
4089     togl{Xnolinenumberifannotation@\@currentseries}%
4090     and not test{\ifdefvoid{\annot@start}}}%
4091     and not test{\ifdefvoid{\annot@end}}}%
4092   )%
4093 }{%
4094   \l@d@Xlinenumberfalse%
4095 }{%
4096 %

```

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.

```

4097 \ifbypage@
4098   \ifnum#4=#1 \else
4099     \l@d@pnumtrue
4100     \l@d@dashtrue
4101   \fi
4102 \fi
4103 %

```

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.

```

4104 \ifboolexpr{%
4105   bool{\l@d@pnum}%
4106   or (%
4107     not test{\ifnumequal{#2}{#5}}%
4108     and bool{\l@d@Xlinenumber}%
4109   )%
4110 }{%
4111   \l@d@elintrue%
4112   \l@d@dashtrue%
4113   \unless\ifx\relax\annot@end%
4114     \def\@annot@end@print{%
4115       \l@wrapcs@ifnotemptybox{Xwraplinenumannotation@\@currentseries}{\
annot@end}%
4116     }%
4117   \fi%

```

```

4118   }{}%
4119   %

```

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.

```

4120   \ifl@d@elin%
4121   \def\@annot@start@print{%
4122     \l@wrapcs@ifnotemptybox{Xwraplinenumannotation@\@currentseries}{\
annot@start}%
4123   }%
4124   \else%
4125     \ifx\@annot@start\annot@end%
4126       \unless\ifx\@annot@start\relax%
4127         \def\@annot@start@print{%
4128           \l@wrapcs@ifnotemptybox{Xwraplinenumannotation@\@currentseries}{\
annot@start}%
4129         }%
4130       \fi%
4131     \else%
4132       \ifx\@annot@end@print\relax%
4133         \def\@annot@start@print{%
4134           \l@wrapcs@ifnotemptybox{Xwraplinenumannotation@\@currentseries}{%
4135             \ifx\annot@start\empty\else%
4136               \annot@start%
4137               \ifdefined\linangesep%
4138                 \linangesep%
4139             \else%
4140               \csuse{Xlinangeseparator@\@currentseries}%
4141             \fi%
4142           \fi%
4143           \annot@end%
4144         }%
4145       \else%
4146         \let\@annot@start@print\@annot@end@print%
4147         \let\@annot@end@print\relax%
4148       \fi%
4149     \fi%
4150   \fi%
4151 \fi%
4152 %

```

We print the starting sub-line if it is nonzero.

```

4153   \l@d@ssubfalse
4154   \ifnum#3=0 \else
4155     \l@d@ssubtrue
4156   \fi
4157 %

```

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.

```

4158 \l@d@eslfalse
4159 \ifnum#6=0 \else
4160   \ifnum#6=#3
4161     \ifl@d@elin \l@d@esltrue \else \l@d@eslfalse \fi
4162   \else
4163     \l@d@esltrue
4164     \l@d@dashtrue
4165   \fi
4166 \fi%
4167 %

```

However, if the `\Xtwolines` is set for the current series, we do not print the last line number.

```

4168 \ifl@d@dash%
4169   \ifboolexpr{togl{fulllines@} or test{\ifcsemt{Xtwolines@}
4170   \currentseries}}}%
4171   {%
4172   \setistwofollowinglines{#1}{#2}{#4}{#5}%
4173   \ifboolexpr{%
4174     (%
4175       togl {Xtwolinesbutnotmore@}\currentseries}%
4176     and not%
4177     (%
4178       bool {istwofollowinglines@}%
4179     )%
4180   )%
4181   or%
4182   (%
4183     (not test{\ifnumequal{#1}{#4}})%
4184     and togl{Xtwolinesonlyinsamepage@}\currentseries}%
4185   )%
4186   }%
4187   {%
4188   \l@d@dashfalse%
4189   \l@d@Xtwolinestrue%
4190   \l@d@elinfalse%
4191   \l@d@eslfalse%
4192   \ifcsemt{Xmorethantwolines@}\currentseries}%
4193   {%
4194     \ifistwofollowinglines@ \else%
4195       \l@d@Xmorethantwolinestrue%
4196     \fi%
4197   }%
4198   }%
4199   }%
4200 }%

```

```
4201 \fi%
4202 %
```

If the `\Xnoidenticallylinenumannotation` 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

```
4203 \iftoggle{Xnoidenticallylinenumannotation@}\@currentseries}{%
4204 \ifx\annot@start\annot@end%
4205 \let\@annot@end@print\relax%
4206 \ifboolexpr{%Do not print the dash if we do not print the line number
4207 test {\ifdefempty{\linenumrep}}}%
4208 or not bool {l@d@Xlinenum}%
4209 }{%
4210 \l@d@dashfalse%
4211 }{%
4212 \fi%
4213 }{%
4214 %
```

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.

```
4215 \iftoggle{Xlinenumannotationonlyfirst@}\@currentseries}{%
4216 \ifboolexpr{%
4217 (%
4218 togl{Xlinenumannotationonlyfirstintwo@}\@currentseries}%
4219 and test{\ifcsequal{annot@start}{prevannot@start@}\@currentseries}}%
4220 and test{\ifcsequal{annot@end}{prevannot@end@}\@currentseries}}%
4221 )%
4222 or%
4223 (%
4224 not togl{Xlinenumannotationonlyfirstintwo@}\@currentseries}%
4225 test{\ifcsequal{annot@start}{prevannot@start@}\@currentseries}}%
4226 )%
4227 }{%
4228 \def\@annot@start@print{%
4229 \l@wrapcs@ifnotemptybox{Xwraplinenumannotation@}\@currentseries}{\
csuse{Xsymlinenumannotation@}\@currentseries}}%
4230 }%
4231 \let\@annot@end@print\relax%
4232 \ifboolexpr{%Do not print the dash if we do not print the line number
4233 test {\ifdefempty{\linenumrep}}}%
4234 or not bool {l@d@Xlinenum}%
4235 }{%
4236 \l@d@dashfalse%
4237 }{%
4238 }{%
4239 \global\cslet{prevannot@start@}\@currentseries}{\annot@start}%
4240 \global\cslet{prevannot@end@}\@currentseries}{\annot@end}%
4241 }%
```



```
4242 {}%
4243 %
```

End of \setprintlines.

```
4244 }%
4245 %
```

\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.

```
4246 \newif\ifistwofollowinglines@%
4247 \newcommand{\setistwofollowinglines}[4]{%
4248   \ifcsdef{lastlinenumberon@#1}%
4249     {\numdef{\tmp}{\csuse{lastlinenumberon@#1}}}%
4250     {\numdef{\tmp}{0}}}%
4251   \istwofollowinglines@false%
4252   \ifnumequal{#4-#2}{1}%
4253     {\istwofollowinglines@true}%
4254     {\ifbypage@%
4255       \ifnumequal{#3-#1}{1}%
4256       {%
4257         \ifnumequal{#2}{\tmp}%
4258         {\ifnumequal{#4}{1}{\istwofollowinglines@true}{}}%
4259         {}%
4260       }%
4261       {}%
4262     \fi%
4263   }%
4264 }%
4265 %
```

`\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

```
4266 \def\printlines#1|#2|#3|#4|#5|#6|#7|#8|{%
4267   \begingroup%
4268   %
```

Decide which part of line number components we will print.

```
4269   \setprintlines{#1}{#2}{#3}{#4}{#5}{#6}%
4270   %
```

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.

```
4271   \ifdimequal{\csuse{Xboxstartlinenum@\@currentseries}}{0pt}%
4272     {\bgroup}%
4273     {\leavevmode\hbox to \csuse{Xboxstartlinenum@\@currentseries}\bgroup\
hfill}%
4274   \ifcsstring{Xlinenumannotationposition@\@currentseries}{before}%
4275     {\@annot@start@print}%
4276     {}%
4277   \ifl@d@pnum%
4278     \wrap@edcrossref{\@this@crossref@start}{#1}%
4279     \ifl@d@Xlinenum%
4280       \csuse{Xpagelinesep@\@currentseries}%
4281       \fi%
4282   \fi%
4283   \ifl@d@Xlinenum%
4284     \wrap@edcrossref{\@this@crossref@start}{%
4285       \linenumrep{#2}%
4286       \iftoggle{Xlineflag@\@currentseries}{#8}{}}%
4287     }%
4288   \ifl@d@ssub%
4289     \csuse{Xsublinesep@\@currentseries}%
4290     \wrap@edcrossref{\@this@crossref@start}{\sublinenumrep{#3}}%
4291     \fi%
4292   \fi%
4293   \ifcsstring{Xlinenumannotationposition@\@currentseries}{after}%
4294     {\@annot@start@print}%
4295     {}%
4296   \egroup%
4297   %
```

Then print the dash + end line number, or the range symbol.

```
4298   \ifdimequal{\csuse{Xboxendlinenum@\@currentseries}}{0pt}%
4299     {\bgroup}%

```

```

4300 {\hbox to \csuse{Xboxendlinenum@\@currentseries}\bgroup}%
4301 \ifl@d@Xtwolines%
4302 \ifl@d@Xlinenumber%
4303 \ifl@d@Xmorethantwolines%
4304 \csuse{Xmorethantwolines@\@currentseries}%
4305 \else%
4306 \csuse{Xtwolines@\@currentseries}%
4307 \fi%
4308 \fi%
4309 \else%
4310 \ifl@d@dash%
4311 \ifdefined\linrangesep%
4312 \linrangesep%
4313 \else%
4314 \csuse{Xlinrangeseparator@\@currentseries}%
4315 \fi%
4316 \fi%
4317 \ifcsstring{Xlinenumannotationposition@\@currentseries}{before}%
4318 {\@annot@end@print}%
4319 {}%
4320 \ifl@d@pnum%
4321 \wrap@edcrossref{\@this@crossref@end}{#4}%
4322 \csuse{Xpagelinesep@\@currentseries}%
4323 \fi%
4324 \ifl@d@Xlinenumber%
4325 \ifl@d@elin%
4326 \wrap@edcrossref{\@this@crossref@end}{%
4327 \linenumrep{#5}%
4328 \iftoggle{Xlineflag@\@currentseries}{#8}{}}%
4329 }%
4330 \fi%
4331 \ifl@d@esl%
4332 \ifl@d@elin%
4333 \csuse{Xsublinesep@\@currentseries}%
4334 \fi%
4335 \wrap@edcrossref{\@this@crossref@end}{\sublinenumrep{#6}}%
4336 \fi%
4337 \fi%
4338 \ifcsstring{Xlinenumannotationposition@\@currentseries}{after}%
4339 {\@annot@end@print}%
4340 {}%
4341 \fi%
4342 \ifdimequal{\csuse{Xboxendlinenum@\@currentseries}}{0pt}%
4343 {}%
4344 {\hfill}%Prevent underfull hbox
4345 \egroup%
4346 \endgroup%
4347 }%
4348 %

```

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

```

4349 \newcommand{\prepare@Xgroupbyline}[3]{%
4350   \iftoggle{Xgroupbylineseparatetwolines@#1}{%
4351     \l@dparsfootspec#2%
4352     \ifcsdef{#1@forinserting@l@dparseendpage-l@dparseendline-l@
l@dparseendsub}%
4353     {%
4354       \csgappto{#1@forinserting@l@dparseendpage-l@dparseendline-l@
l@dparseendsub}{%
4355         \hskip\csuse{Xafternote@#1}\relax%
4356       }%
4357     }%
4358   }%
4359   \add@hooktoggle@specific@to@cs{#1@forinserting@l@dparseendpage-l@
l@dparseendline-l@dparseendsub}%
4360   \add@hookarg@specific@to@cs{#1@forinserting@l@dparseendpage-l@
l@dparseendline-l@dparseendsub}%
4361   \csxappto%
4362     {#1@forinserting@l@dparseendpage-l@dparseendline-l@dparseendsub
}%
4363   {%
4364     \keep@this@crossref@forinserting%
4365     \unexpanded{%
4366       \ifcsemt{Xsymlinum@#1}%
4367         {\csuse{Xparafootsep@#1}}%
4368       }%
4369       #3{#1}{#2}%
4370     }%
4371   }%
4372   \listcsxadd{#1@forinserting}{l@dparseendpage-l@dparseendline-l@
l@dparseendsub}%
4373 }{%
4374   \ifcsdef{#1@forinserting@all}{%
4375     \csgappto%
4376       {#1@forinserting@all}%
4377       {\hskip\csuse{Xafternote@#1}\relax}%
4378     }{%
4379       \add@hooktoggle@specific@to@cs{#1@forinserting@all}%
4380       \add@hookarg@specific@to@cs{#1@forinserting@all}%
4381       \csxappto%
4382         {#1@forinserting@all}%
4383       {%
4384         \keep@this@crossref@forinserting%

```

```

4385 \unexpanded{%
4386 \ifcsempy{Xsymlinenumber@#1}%
4387 {\csuse{Xparafootsep@#1}}%
4388 {}%
4389 #3{#1}{#2}%
4390 }%
4391 }%
4392 }%
4393 \listcsgadd{#1@forinserting}{all}%
4394 }%
4395 %

```

`\this@crossref@forinserting` `\keep@this@crossref@forinserting` is called by `\prepare@Xgroupbyline`. It just ensures that `\@this@crossref@start` and `\this@crossref@end` value are kept in memory.

```

4396 \newcommand{\keep@this@crossref@forinserting}{%
4397 \unexpanded{\gdef\@this@crossref@start}{\@this@crossref@start}%
4398 \unexpanded{\gdef\@this@crossref@end}{\@this@crossref@end}%
4399 }%
4400 %

```

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` `\multfootsep` 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`.

```

4401 \providecommand*\multiplefootnotemarker{3sp}
4402 \providecommand*\multfootsep{\textsuperscript{\normalfont,}}
4403
4404 %

```

`\m@mmf@prepare` A pair of self-cancelling kerns. This may have been defined in the `memoir` class.

```

4405 \providecommand*\m@mmf@prepare}{%
4406 \kern-\multiplefootnotemarker
4407 \kern\multiplefootnotemarker\relax}
4408 %

```

\m@mmf@check This may have been defined in the memoir class. If it recognises the last kern as \multiplefootnotemarker it typesets \multfootsep.

```

4409 \providecommand*\m@mmf@check}{%
4410 \ifdim\lastkern=\multiplefootnotemarker\relax
4411 \edef\@x@sf{\the\spacefactor}%
4412 \unkern
4413 \multfootsep
4414 \spacefactor\@x@sf\relax
4415 \fi}
4416 %
4417 %

```

We have to modify \@footnotetext and \@footnotemark. However, if memoir is used the modifications have already been made.

```

4418 \@ifclassloaded{memoir}{}{%
4419 %

```

\@footnotetext Add \m@mmf@prepare at the end of \@footnotetext.

```

4420 \apptocmd{\@footnotetext}{\m@mmf@prepare}{}{}
4421 %

```

\@footnotemark Modify \@footnotemark to cater for adjacent \footnotes.

```

4422 \patchcmd{\@footnotemark}
4423 {\nobreak}
4424 {\m@mmf@check
4425 \nobreak
4426 }
4427 {}{}
4428 \patchcmd{\@footnotemark}
4429 {\@makefnmark}
4430 {\@makefnmark
4431 \m@mmf@prepare
4432 }
4433 {}{}
4434 %
4435 %

```

Finished the modifications for the non-memoir case.

```

4436 }
4437 %
4438 %

```

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.

```

4439 \pretocmd{\@footnotetext}{%
4440   \ifnumberedpar@
4441   \edtext{}\l@dbfnote{#1}}%
4442 \else
4443   {}{}
4444 \apptocmd{\@footnotetext}{\fi}{}{}%
4445 %

```

`\l@dbfnote` `\l@dbfnote` adds the footnote to the insert list, and `\vl@dbfnote` calls the original
`\vl@dbfnote` `\@footnotetext`. We also patch `\footnote` in order to get the correct footnote
`\vl@dbfnote` numbers when typesetting parallel texts. This is moved into a `\get@fnmark` command.

```

\footnote
\get@fnmark
\get@thisfootnote
4446
4447 \patchcmd%
4448   {\footnote}%
4449   {\stepcounter\@mpfn}%
4450   {%
4451   \ifboolexpr{bool{l@dpairing} or bool{l@dprintingpages} or bool{
l@dprintingcolumns}}{%
4452     \global\advance\footnote@reading by \@ne%
4453     \get@thisfootnote%
4454     \get@fnmark{\thisc@footnote}%
4455     \ifcsdef{footnotereading\the\footnote@reading=typeset}%
4456       {\setcounter{\@mpfn}{\csuse{footnotereading\the\footnote@reading=
typeset}}}%
4457       {\setcounter{\@mpfn}{\footnote@reading}}%
4458     }{%
4459       \stepcounter\@mpfn%
4460     }%
4461   }%
4462   {}
4463   {}
4464
4465 \newcommand{\get@thisfootnote}{%
4466   \ifboolexpr{bool{l@dpairing} or bool{l@dprintingpages} or bool{
l@dprintingcolumns}}{%
4467     \protected@xdef\thisc@footnote{\the\footnote@reading}%
4468   }{%
4469     \protected@xdef\thisc@footnote{\the\c@footnote}%
4470   }%
4471 }%
4472
4473 \newcommand{\l@dbfnote}[1]{%

```

```

4474 \get@thisfootnote%
4475 \gdef\@tag{#1\relax}%
4476 \ifledRcol%
4477 \xright@appenditem{%
4478 \ifdefined\Hy@footnote@currentHref%
4479 \noexpand\def\noexpand\Hy@footnote@currentHref{\
Hy@footnote@currentHref}%
4480 \fi%
4481 \noexpand\vl@dbfnote{\expandonce\@tag}}{\thisc@footnote}%
4482 }%
4483 \to\inserts@listR
4484 \global\advance\insert@countR \@ne%
4485 \else%
4486 \xright@appenditem{%
4487 \ifdefined\Hy@footnote@currentHref%
4488 \noexpand\def\noexpand\Hy@footnote@currentHref{\
Hy@footnote@currentHref}%
4489 \fi%
4490 \noexpand\vl@dbfnote{\expandonce\@tag}}{\thisc@footnote}%
4491 }%
4492 \to\inserts@list
4493 \global\advance\insert@count \@ne%
4494 \fi
4495 \ignorespaces%
4496 }%
4497
4498 \newcommand{\get@fnmark}[1]{%
4499 \ifboolexpr{bool{l@dpairing} or bool{l@dprintingpages} or bool{
l@dprintingcolumns}}%
4500 {%
4501 \stepcounter{footnote@typeset}%
4502 \setcounter{footnote}{\c@footnote@typeset}%
4503 \immediate\write\@mainaux{%
4504 \csgdef{footnotereading#1=typeset}{\the\c@footnote@typeset}%
4505 }%
4506 \def\@thefnmark{\thefootnote}%
4507 }%
4508 {%
4509 \setcounter{footnote}{#1}%
4510 \def\@thefnmark{\thefootnote}%
4511 }%
4512 }%
4513
4514 \newcommand{\vl@dbfnote}[2]{%
4515 \get@fnmark{#2}%
4516 \@footnotetext{#1}%
4517 }%
4518 %

```


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`

```

4519 \newcommand*{\prebodyfootmark}{%
4520   \leavevmode
4521   \ifhmode
4522     \edef\x@sf{\the\spacefactor}%
4523     \m@mmf@check
4524     \nobreak
4525   \fi}
4526 \newcommand{\postbodyfootmark}{%
4527   \m@mmf@prepare
4528   \ifhmode\spacefactor\x@sf\fi\relax}
4529
4530 %

```

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.

```

4531 \newcommandx{\arrangementX}[2][1,usedefault]{%
4532   \def\do##1{%
4533     \csname arrangementX@#2\endcsname{##1}%
4534   }%
4535   \ifstrempy{#1}%
4536     {%
4537       \dolistloop{\@series}%
4538     }%
4539     {
4540       \docsvlist{#1}%
4541     }%
4542   }%
4543 %

```

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.

```

4544 \newcommand*{\normal@footnotemarkX}[1]{%
4545   \prebodyfootmark
4546   \wrapped@bodyfootmarkX{#1}%
4547   \postbodyfootmark}

```

```
4548
4549 %
```

`\normalbodyfootmarkX` The `\normalbodyfootmarkX{⟨series⟩}` really typesets the in-text marker. The style is the normal superscript.

```
4550 \newcommand*{\normalbodyfootmarkX}[1]{%
4551   \hbox{\textsuperscript{\normalfont\@nameuse{@thefnmark#1}}}%
4552 %}
```

`\normalvfootnoteX` `\normalvfootnoteX{⟨series⟩}{⟨text⟩}` does the `\insert` for the `⟨series⟩` and calls the series' `\footfmt... to format the ⟨text⟩`.

```
4553 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\normalvfootnoteX}[2]{%
4554   \beforeinsertionX{#1}%
4555   \insert\@nameuse{footins#1}\bgroup
4556     \fontseries{seriesdefault}%
4557     \fontshape{shapedefault}%
4558     \hsize=\expandafter\dimexpr\csuse{widthX@#1}\relax%
4559     \noindent\csuse{bhooknoteX@#1}%
4560     \csuse{notefontsizeX@#1}%
4561     \footplitskips
4562     \ifl@dpairing\ifl@dpageing\else%
4563       \setnoteswidthliketwocolumnsX@{#1}%
4564     \fi\fi%
4565     \setnotesXpositionliketwocolumns@{#1}%
4566     \spaceskip=\z@skip \xspaceskip=\z@skip
4567     \csuse{\csuse{footnote@dir}}\@nameuse{footfmt#1}{#1}{#2}\egroup}
4568
4569 %}
```

`\mpnormalvfootnoteX` The minipage version.

```
4570 \newcommand*{\mpnormalvfootnoteX}[3]{%
4571   \get@thisfootnoteX{#1}%
4572   \get@fnmarkX{#1}{\thisc@footnote}{#3}%
4573   \ifstrempy{#3}{%
4574     \edef\this@footnoteX@reading{\the\c@name footnote#1@reading\endc@name}%
4575   }{%
4576     \edef\this@footnoteX@reading{###3}%
4577   }%
4578   \global\setbox\@nameuse{mpfootins#1}\vbox{%
4579     \unvbox\@nameuse{mpfootins#1}
4580     \noindent\csuse{bhooknoteX@#1}%
4581     \csuse{notefontsizeX@#1}%
4582     \hsize\columnwidth
4583     \@parboxrestore
4584     \color@begingroup
4585     \@nameuse{footfmt#1}{#1}{#2}\color@endgroup}}
4586
4587 %}
```

`\normalfootfmtX` `\normalfootfmtX{<series>}{<text>}` typesets the footnote text, prepended by the marker.

```

4588 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\normalfootfmtX}[2]{%
4589 \ifluatex%
4590 \textdir\footnote@luatextextdir%
4591 \pardir\footnote@luatexpardir%
4592 \fi%
4593 \protected@edef\@currentlabel{%
4594 \@nameuse{@thefnmark#1}%
4595 }%
4596 \ledsetnormalparstuffX{#1}%
4597 \sethangindentX{#1}%
4598 \rule\z@\splittopskip%
4599 {{\csuse{notenunfontX@#1}\wrapped@footfootmarkX{#1}}%
4600 \csuse{wrapcontentX@#1}{#2}%
4601 \strut\par}}
4602 %
4603 %

```

`\normalfootfootmarkX` `\normalfootfootmarkX{<series>}` is called by `\normalfootfmtX` to typeset the footnote marker in the footer before the footnote text.

```

4604 \newcommand*{\normalfootfootmarkX}[1]{%
4605 \textsuperscript{\@nameuse{@thefnmark#1}}}
4606 %
4607 %

```

`\normalfootstartX` `\normalfootstartX{<series>}` is the `<series>` footnote starting macro used in the output routine.

```

4608 \newcommand*{\normalfootstartX}[1]{%
4609 \ifdimequal{Opt}{\prenotesX@}{}%
4610 {%
4611 \iftoggle{prenotesX@}{%
4612 \togglefalse{prenotesX@}%
4613 \skip\csname footins#1\endcsname=%
4614 \glueexpr\csuse{prenotesX@}+\csuse{afterruleX@#1}\relax%
4615 }%
4616 }%
4617 }%
4618 \vskip\skip\csname footins#1\endcsname%
4619 \leftskip=\z@
4620 \rightskip=\z@
4621 \ifl@dpairing\else%
4622 \hsize=\old@hsiz%
4623 \fi%
4624 \setnoteswidthliketwocolumnsX@{#1}%
4625 \setnotesXpositionliketwocolumns@{#1}%
4626 \print@footnoteXrule{#1}%
4627 }%

```

```
4628
4629 %
```

\normalfootnoteruleX The rule drawn before the footnote series group.

```
4630 \let\normalfootnoteruleX=\footnoterule
4631
4632 %
```

\normalfootgroupX `\normalfootgroupX{<series>}` sends the contents of the `<series>` insert box to the output page without alteration.

```
4633 \newcommand*\normalfootgroupX[1]{%
4634   \csuse{bhookgroupX@#1}%
4635   \unvbox\@nameuse{footins#1}%
4636   \hsize=\old@hsize%
4637 }%
4638
4639 %
```

\mpnormalfootgroupX The minipage version.

```
4640 \newcommand*\mpnormalfootgroupX[1]{%
4641   \vskip\skip\@nameuse{mpfootins#1}
4642   \ifl@dpairing\ifparledgroup%
4643     \leavevmode\marks\parledgroup@{begin}%
4644     \marks\parledgroup@series{#1}%
4645     \marks\parledgroup@type{footnoteX}%
4646     \fi\fi\normalcolor
4647   \ifparledgroup%
4648     \ifl@dpairing%
4649     \else%
4650       \setnoteswidthliketwocolumnsX@{#1}%
4651       \setnotesXpositionliketwocolumns@{#1}%
4652       \print@footnoteXrule{#1}%
4653     \fi%
4654   \else%
4655     \setnoteswidthliketwocolumnsX@{#1}%
4656     \setnotesXpositionliketwocolumns@{#1}%
4657     \print@footnoteXrule{#1}%
4658   \fi%
4659   \csuse{bhookgroupX@#1}%
4660   \unvbox\@nameuse{mpfootins#1}}
4661
4662 %
```

\normalbfnoteX #1 = footnote series, #2 = footnote content, #3 manual footnote number

```
4663
4664 \newcommand{\normalbfnoteX}[3]{%
```

```

4665 \get@thisfootnoteX{#1}%
4666 \ifledRcol%
4667 \ifluatex
4668 \footnotelang@lua[R]%
4669 \fi
4670 \@ifundefined{xpg@main@language}%if polyglossia
4671 {}%
4672 {\footnotelang@poly[R]}%
4673 \xright@appenditem{%
4674 \noexpand\led@set@index@fornote{#1}%
4675 \noexpand\prepare@edindex@fornote{\led@nums}%
4676 \unexpanded{\gdef\@currentnotetype{familiar}}%
4677 \unexpanded{\def\this@footnoteX@reading}{\the\cscname footnote#1
@reading\endcscname}%
4678 \noexpand\vbfnoteX{#1}{#2}{\thisc@footnote}{#3}%
4679 \noexpand\led@reinit@index@fornote%
4680 \unexpanded{\advance\@edindex@fornote@m@ne}%
4681 }%
4682 \to\inserts@listR
4683 \global\advance\insert@countR \@ne%
4684 \else%
4685 \ifluatex
4686 \footnotelang@lua%
4687 \fi
4688 \@ifundefined{xpg@main@language}%if polyglossia
4689 {}%
4690 {\footnotelang@poly}%
4691 \xright@appenditem{%
4692 \noexpand\led@set@index@fornote{#1}%
4693 \noexpand\prepare@edindex@fornote{\led@nums}%
4694 \unexpanded{\gdef\@currentnotetype{familiar}}%
4695 \unexpanded{\def\this@footnoteX@reading}{\the\cscname footnote#1
@reading\endcscname}%
4696 \noexpand\vbfnoteX{#1}{#2}{\thisc@footnote}{#3}%
4697 \noexpand\led@reinit@index@fornote%
4698 \unexpanded{\advance\@edindex@fornote@m@ne}%
4699 }%
4700 \to\inserts@list
4701 \global\advance\insert@count \@ne%
4702 \fi
4703 \ignorespaces}
4704
4705 %

```

`\get@thisfootnoteX` The macro `\get@thisfootnote` command just saves the footnote number in the `\thisfootnote` macro, depending on the use of pairing environments.

```

4706 \newcommand{\get@thisfootnoteX}[1]{%
4707 \ifboolexpr{bool{l@dpairing} or bool{l@dprintingpages} or bool{
l@dprintingcolumns}}{%

```

```

4708 \protected@xdef\thisc@footnote{\the\cename footnote#1@reading\
endcsname}%
4709 }{%
4710 \protected@xdef\thisc@footnote{\the\cename c@footnote#1\endcsname}%
4711 }%
4712 }%
4713 %

```

\vbfnoteX This command calls the correct footnote-inserting commands. #1 footnote series, #2 footnote content, #3 footnote counter, #4 manual footnote mark

```

4714 \newcommand{\vbfnoteX}[4]{%
4715 \get@fnmarkX{#1}{#3}{#4}\relax%
4716 \@nameuse{regvfootnote#1}{#1}{#2}%
4717 }%
4718 %

```

\get@fnmarkX This command gets the correct footnote number when typesetting parallel texts. #1 footnote series, #2 footnote counter, #3 manual footnote number

```

4719 \newcommand{\get@fnmarkX}[3]{%
4720 \ifstrempy{#3}{%
4721 \ifboolexpr{bool{l@dpairing} or bool{l@dprintingpages} or bool{
l@dprintingcolumns}}{%
4722 {%
4723 \stepcounter{footnote#1@typeset}%
4724 \setcounter{footnote#1}{\value{footnote#1@typeset}}%
4725 \@namedef{@thefnmark#1}{\csuse{thefootnote#1}}%
4726 \immediate\write\@mainaux{%
4727 \csgdef{footnote#1reading#2=typeset}{\the\cename c@footnote#1
@typeset\endcsname}%
4728 }%
4729 }%
4730 {%
4731 \setcounter{footnote#1}{#2}%
4732 \@namedef{@thefnmark#1}{\csuse{thefootnote#1}}%
4733 }%
4734 }%
4735 {%
4736 \csdef{@thefnmark#1}{#3}%
4737 }%
4738 }%
4739 %
4740 %

```

\vnumfootnoteX #1 footnote series, #2 footnote content, #3 manual footnote mark

```

4741 \newcommand{\vnumfootnoteX}[3]{%
4742 \ifnumberedpar@
4743

```

```

4744 \edtext{}{\normalbfnoteX{#1}{#2}{#3}}%
4745 \else
4746 \def\this@footnoteX@reading{\the\csname footnote#1@reading\endcsname}%
4747 \get@thisfootnoteX{#1}%
4748 \get@fnmarkX{#1}{\expandonce\thisc@footnote}{#3}%
4749 \@nameuse{regvfootnote#1}{#1}{#2}%
4750 \fi}
4751 %

```

`arrangementX@normal` `\arrangementX@normal{<series>}` initialises the settings for the `<series>` footnotes. This should always be called for each series.

```

4752 \newcommand*{\arrangementX@normal}[1]{%
4753 \csgdef{series@displayX#1}{normal}
4754 \expandafter\let\csname footstart#1\endcsname=\normalfootstartX
4755 \@namedef{@footnotemark#1}{\normal@footnotemarkX{#1}}
4756 \@namedef{bodyfootmark#1}{\normalbodyfootmarkX{#1}}
4757 \expandafter\let\csname regvfootnote#1\endcsname=\normalvfootnoteX
4758 \expandafter\let\csname vfootnote#1\endcsname=\vnumfootnoteX
4759 \expandafter\let\csname footfmt#1\endcsname=\normalfootfmtX
4760 \@namedef{footfootmark#1}{\normalfootfootmarkX{#1}}
4761 \expandafter\let\csname footgroup#1\endcsname=\normalfootgroupX
4762 \expandafter\let\csname footnoterule#1\endcsname=\normalfootnoteruleX
4763 \count\csname footins#1\endcsname=1000
4764 \csxdef{default@footins#1}{1000}%Use to have note only for one side
4765 \dimen\csname footins#1\endcsname=\csuse{maxhnotesX@#1}
4766 \skip\csname footins#1\endcsname=\csuse{beforenotesX@#1}%
4767 \advance\skip\csname footins#1\endcsname by\csuse{afterruleX@#1}%
4768 %

```

Additions for minipages.

```

4769 \ifnoledgroup@else%
4770 \expandafter\let\csname mpvfootnote#1\endcsname=\mpnormalvfootnoteX
4771 \expandafter\let\csname mpfootgroup#1\endcsname=\mpnormalfootgroupX
4772 \count\csname mpfootins#1\endcsname=1000
4773 \dimen\csname mpfootins#1\endcsname=\csuse{maxhnotesX@#1}
4774 \skip\csname mpfootins#1\endcsname=\csuse{beforenotesX@#1}%
4775 \advance\skip\csname mpfootins#1\endcsname by\csuse{afterruleX@#1}%
4776 \fi
4777 }
4778
4779 %

```

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 4780 \newcommand*{\arrangementX@twocol}[1]{%
4781   \csgdef{series@displayX#1}{twocol}
4782   \expandafter\let\csname regvfootnote#1\endcsname=\twocolvfootnoteX
4783   \expandafter\let\csname footfmt#1\endcsname=\twocolfootfmtX
4784   \expandafter\let\csname footgroup#1\endcsname=\twocolfootgroupX
4785   \dimen\csname footins#1\endcsname=\csuse{maxhnotesX@#1}%
4786   \skip\csname footins#1\endcsname=\csuse{beforenotesX@#1}%
4787   \advance\skip\csname footins#1\endcsname by \csuse{afterruleX@#1}\relax%
4788   \twocolfootsetupX{#1}
4789   \ifnoledgroup@else%
4790     \expandafter\let\csname mpvfootnote#1\endcsname=\mpnormalvfootnoteX
4791     \expandafter\let\csname mpfootgroup#1\endcsname=\mptwocolfootgroupX
4792     \skip\csname mpfootins#1\endcsname=\csuse{beforenotesX@#1}%
4793     \advance\skip\csname mpfootins#1\endcsname by \csuse{afterruleX@#1}
4794     \mptwocolfootsetupX{#1}
4795   \fi%
4796 }
4797
4798 %

```

```

\twocolfootsetupX \twocolfootsetupX{<series>}
\mptwocolfootsetupX
4799 \newcommand*{\twocolfootsetupX}[1]{%
4800   \count\csname footins#1\endcsname 500
4801   \csxdef{default@footins#1}{500}%Use this to confine the notes to one
side only
4802   \multiply\dimen\csname footins#1\endcsname by \tw@}
4803 \newcommand*{\mptwocolfootsetupX}[1]{%
4804   \count\csname mpfootins#1\endcsname 500
4805   \multiply\dimen\csname mpfootins#1\endcsname by \tw@}
4806
4807 %

```

```

\twocolvfootnoteX \twocolvfootnoteX{<series>}
4808 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\twocolvfootnoteX}[2]{%
4809   \beforeinsertion@X{#1}%
4810   \let\bidir@RTL@everypar\relax%
4811   \insert\csname footins#1\endcsname\bgroup%
4812     \hsize=\expandafter\dimexpr\csuse{widthX@#1}\relax%
4813     \noindent\csuse{bhooknoteX@#1}%
4814     \csuse{notefontsizeX@#1}%
4815     \footsplitskips%
4816     \spaceskip=\z@skip \xspaceskip=\z@skip%
4817     \@nameuse{footfmt#1}{#1}{#2}\egroup}
4818
4819 %

```

```

\twocolfootfmtX \twocolfootfmtX{<series>}

```



```

4820 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\twocolfootfmtX}[2]{%
4821   \protected@edef\@currentlabel{%
4822     \@nameuse{@thefnmark#1}%
4823   }%
4824   \normal@pars%
4825   \sethangindentX{#1}%
4826   \hsize \csuse{hsizetwocolX@#1}%
4827   \setparindentX{#1}%
4828   \tolerance=5000\relax%
4829   \par%
4830   \@tempdima=\parindent%
4831   \csuse{colalignX@#1}%
4832   \parindent=\@tempdima%
4833   {\hspace{\parindent}%
4834     \csuse{notenumfontX@#1}\wrapped@footfootmarkX{#1}\strut%
4835     \csuse{wrapcontentX@#1}{#2}%
4836     \strut\par}%
4837   \allowbreak%
4838 }%
4839 %
4840 %

```

```

\twocolfootgroupX \twocolfootgroupX{<series>}
\mptwocolfootgroupX
4841 \newcommand*{\twocolfootgroupX}[1]{\csuse{bhookgroupX@#1}\csuse{
notefontsizeX@#1}
4842 \splittopskip=\ht\strutbox
4843 \expandafter
4844 \rigidbalanceX\csname footins#1\endcsname \tw@ \splittopskip}}
4845
4846 \newcommand*{\mptwocolfootgroupX}[1]{%
4847   \vskip\skip\@nameuse{mpfootins#1}
4848   \ifl@dpairing\ifparledgroup%
4849     \leavevmode\marks\parledgroup@{begin}%
4850     \marks\parledgroup@series{#1}%
4851     \marks\parledgroup@type{footnoteX}%
4852   \fi\fi\normalcolor
4853   \ifparledgroup%
4854     \ifl@dpairing%
4855     \else%
4856       \setnoteswidthliketwocolumnsX@{#1}%
4857       \setnotesXpositionliketwocolumns@{#1}%
4858       \print@footnoteXrule{#1}%
4859     \fi%
4860   \else%
4861     \setnoteswidthliketwocolumnsX@{#1}%
4862     \setnotesXpositionliketwocolumns@{#1}%
4863     \print@footnoteXrule{#1}%
4864   \fi%
4865   \csuse{bhookgroupX@#1}%

```

```

4866 \splittopskip=\ht\strutbox
4867 \expandafter
4868 \rigidbalanceX\csname mpfootins#1\endcsname \tw@ \splittopskip}}
4869
4870 %

```

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 4871 \newcommand*\arrangementX@threecol}[1]{%
4872   \csgdef{series@displayX#1}{threecol}
4873   \expandafter\let\csname regvfootnote#1\endcsname=\threecolvfootnoteX
4874   \expandafter\let\csname footfmt#1\endcsname=\threecolfootfmtX
4875   \expandafter\let\csname footgroup#1\endcsname=\threecolfootgroupX
4876   \dimen\csname footins#1\endcsname=\csuse{maxhnotesX@#1}%
4877   \skip\csname footins#1\endcsname=\csuse{beforenotesX@#1}%
4878   \advance\skip\csname footins#1\endcsname by \csuse{afterruleX@#1}\relax%
4879   \threecolfootsetupX{#1}
4880   \ifnoledgroup@else%
4881     \expandafter\let\csname mpvfootnote#1\endcsname=\mpnormalvfootnoteX
4882     \expandafter\let\csname mpfootgroup#1\endcsname=\mpthreecolfootgroupX
4883     \skip\csname mpfootins#1\endcsname=\csuse{beforenotesX@#1}%
4884     \advance\skip\csname mpfootins#1\endcsname by \csuse{afterruleX@#1}
4885     \mpthreecolfootsetupX{#1}
4886   \fi%
4887 }
4888
4889 %

```

```

\threecolfootsetupX \threecolfootsetupX{<series>}
\mpthreecolfootsetupX
4890 \newcommand*\threecolfootsetupX}[1]{%
4891   \count\csname footins#1\endcsname 333
4892   \csxdef{default@footins#1}{333}%Use this to confine the notes to one
side only
4893   \multiply\dimen\csname footins#1\endcsname by \thr@@}
4894 \newcommand*\mpthreecolfootsetupX}[1]{%
4895   \count\csname mpfootins#1\endcsname 333
4896   \multiply\dimen\csname mpfootins#1\endcsname by \thr@@}
4897
4898 %

```

```

\threecolvfootnoteX \threecolvfootnoteX{<series>}{<text>}
4899 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\threecolvfootnoteX}[2]{
%
4900 \let\bidir@RTL@everypar\relax%

```

```

4901 \beforeinsertionX{#1}%
4902 \insert\csname footins#1\endcsname\bgroup%
4903   \hsize=\expandafter\dimexpr\csuse{widthX@#1}\relax%
4904   \noindent\csuse{bhooknoteX@#1}%
4905   \csuse{notefontsizeX@#1}%
4906   \footsplitskips%
4907   \@nameuse{footfmt#1}{#1}{#2}\egroup}
4908
4909 %

```

`\threecolfootfmtX` `\threecolfootfmtX{<series>}`

```

4910 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\threecolfootfmtX}[2]{%
4911   \protected@edef\@currentlabel{%
4912     \@nameuse{@thefnmark#1}%
4913   }%
4914   \sethangindentX{#1}%
4915   \normal@pars%
4916   \hsize \csuse{hsizethreecolX@#1}%
4917   \setparindentX{#1}%
4918   \tolerance=5000\relax%
4919   \@tempdima=\parindent%
4920   \csuse{colalignX@#1}%
4921   \parindent=\@tempdima%
4922   {\hspace{\parindent}%
4923     \csuse{notenumberX@#1}\wrapped@footfootmarkX{#1}\strut%
4924     \csuse{wrapcontentX@#1}{#2}%
4925     \strut\par}\allowbreak}
4926
4927 %

```

`\threecolfootgroupX` `\threecolfootgroupX{<series>}`
`\mpthreecolfootgroupX`

```

4928 \newcommand*{\threecolfootgroupX}[1]{\csuse{bhookgroupX@#1}\csuse{
notefontsizeX@#1}
4929   \splittopskip=\ht\strutbox
4930   \expandafter
4931   \rigidbalanceX\csname footins#1\endcsname \thr@@ \splittopskip}}
4932
4933 \newcommand*{\mpthreecolfootgroupX}[1]{%
4934   \vskip\skip\@nameuse{mpfootins#1}
4935   \ifl@dpairing\ifparledgroup
4936     \leavevmode\marks\parledgroup@{begin}%
4937     \marks\parledgroup@series{#1}%
4938     \marks\parledgroup@type{footnoteX}%
4939   \fi\fi\normalcolor
4940   \ifparledgroup%
4941     \ifl@dpairing%
4942     \else%
4943       \setnoteswidthliketwocolumnsX@{#1}%

```

```

4944 \setnotesXpositionliketwocolumns@{#1}%
4945 \print@footnoteXrule{#1}%
4946 \fi%
4947 \else%
4948 \setnoteswidthliketwocolumnsX@{#1}%
4949 \setnotesXpositionliketwocolumns@{#1}%
4950 \print@footnoteXrule{#1}%
4951 \fi%
4952 \csuse{bhookgroupX@#1}%
4953 \splittopskip=\ht\strutbox
4954 \expandafter
4955 \rigidbalanceX\csname mpfootins#1\endcsname \thr@@ \splittopskip}}
4956
4957 %

```

XIV.4.5 Paragraphed footnotes

The following macros set footnotes as one paragraph.

`\arrangementX@paragraph` `\footparagraphX{<series>}`

```

4958 \newcommand*{\arrangementX@paragraph}[1]{%
4959 \csgdef{series@displayX#1}{paragraph}%
4960 \expandafter\let\csname footstart#1\endcsname=\parafootstartX
4961 \expandafter\let\csname regvfootnote#1\endcsname=\para@vfootnoteX
4962 \expandafter\let\csname footfmt#1\endcsname=\parafootfmtX
4963 \expandafter\let\csname footgroup#1\endcsname=\para@footgroupX
4964 \expandafter\let\csname footnoterule#1\endcsname=\normalfootnoteruleX
4965 \count\csname footins#1\endcsname=1000
4966 \csxdef{default@footins#1}{1000}%Use this to confine the notes to one
side only
4967 \dimen\csname footins#1\endcsname=\csuse{maxhnotesX@#1}
4968 \skip\csname footins#1\endcsname=\csuse{beforenotesX@#1}%
4969 \advance\skip\csname footins#1\endcsname by\csuse{afterruleX@#1}%
4970 \para@footsetupX{#1}
4971 \ifnoledgroup@else
4972 \expandafter\let\csname mpvfootnote#1\endcsname=\mppara@vfootnoteX
4973 \expandafter\let\csname mpfootgroup#1\endcsname=\mppara@footgroupX
4974 \count\csname mpfootins#1\endcsname=1000
4975 \dimen\csname mpfootins#1\endcsname=\csuse{maxhnotesX@#1}
4976 \skip\csname mpfootins#1\endcsname=\csuse{beforenotesX@#1}%
4977 \advance\skip\csname mpfootins#1\endcsname by\csuse{afterruleX@#1}%
4978 \fi
4979 }
4980
4981 %

```

`\para@footsetupX` `\para@footsetupX{<series>}`

```

4982 \newcommand*{\para@footsetupX}[1]{\csuse{bhookgroupX@#1}\csuse{
notefontsizeX@#1}
4983 \setnoteswidthliketwocolumnsX@{#1}%
4984 \ifcsempy{widthX@#1}%
4985 {}%
4986 {\columnwidth=\expandafter\dimexpr\csuse{widthX@#1}\relax}%
4987 \dimen0=\baselineskip
4988 \multiply\dimen0 by 1024
4989 \divide\dimen0 by \columnwidth \multiply\dimen0 by \footfudgefiddle\relax
4990 %
4991 \expandafter
4992 \xdef\csname footfudgefactor#1\endcsname{%
\expandafter\strip@pt\dimen0 }}
4993 %
4994 %

```

\parafootstartX \parafootstartX{<series>}

```

4995 \newcommand*{\parafootstartX}[1]{%
4996 \ifdimequal{0pt}{\prenotesX@}{}%
4997 {%
4998 \iftoggle{prenotesX@}{%
4999 \togglefalse{prenotesX@}%
5000 \skip\csname footins#1\endcsname=%
5001 \glueexpr\csuse{prenotesX@}+\csuse{afterterruleX@#1}\relax%
5002 }%
5003 }%
5004 }%
5005 \leftskip=\z@
5006 \rightskip=\z@
5007 \setparindentX{#1}%
5008 \vskip\skip\@nameuse{footins#1}%
5009 \setnoteswidthliketwocolumnsX@{#1}%
5010 \setnotesXpositionliketwocolumns@{#1}%
5011 \print@footnoteXrule{#1}%
5012 }
5013 %
5014 %

```

\para@vfootnoteX \para@vfootnoteX{<series>}{<text>}
\mppara@vfootnoteX

```

5015 \newcommand*{\para@vfootnoteX}[2]{%
5016 \csuse{beforeinsertingX@#1}%
5017 \insert\csname footins#1\endcsname%
5018 \bgroup
5019 \csuse{notefontsizeX@#1}
5020 \footsplitskips
5021 \setbox0=\vbox{\hsize=\maxdimen%
5022 \let\bidir@RTL@everypar\@empty%
5023 \insert@txtbeforenotesX{#1}%

```

```

5024 \noindent\csuse{bhooknoteX@#1}%
5025 \@nameuse{footfmt#1}{#1}{#2}}%
5026 \setbox0=\hbox{\unvxhX{0}{#1}}%
5027 \dp0=\z@
5028 \ht0=\csname footfudgefactor#1\endcsname\wd0
5029 \box0
5030 \penalty0
5031 \egroup}
5032 \newcommand*{\mppara@vfootnoteX}[3]{%
5033 \get@thisfootnoteX{#1}%
5034 \get@fnmarkX{#1}{\thisc@footnote}{#3}%
5035 \edef\this@footnoteX@reading{\the\csname footnote#1@reading\endcsname}%
5036 \global\setbox\@nameuse{mpfootins#1}\vbox{%
5037 \unvbox\@nameuse{mpfootins#1}
5038 \csuse{notefontsizeX@#1}
5039 \footsplitskips
5040 \setbox0=\vbox{\hsize=\maxdimen%
5041 \let\bidir@RTL@everypar\@empty%
5042 \noindent\color@begingroup%
5043 \csuse{bhooknoteX@#1}%
5044 \@nameuse{footfmt#1}{#1}{#2}\color@endgroup}%
5045 \setbox0=\hbox{\unvxhX{0}{#1}}%
5046 \dp0=\z@
5047 \ht0=\csname footfudgefactor#1\endcsname\wd0
5048 \box0
5049 \penalty0}}
5050
5051 %

```

```

\unvxhX52 \newcommand*{\unvxhX}[2]{% 2th is optional for retro-compatibility
5053 \setbox0=\vbox{\unvbox#1%
5054 \global\setbox1=\lastbox}%
5055 \unhbox1
5056 \unskip % remove \rightskip,
5057 \unskip % remove \parfillskip,
5058 \unpenalty % remove \penalty of 10000,
5059 \hskip\csuse{afternoteX@#2}%
5060 \relax}% but add the glue to go between the notes
5061
5062 %

```

\parafootfmtX \parafootfmtX{<series>}

```

5063 \newcommand*{\parafootfmtX}[2]{%
5064 \protected@edef\@currentlabel{%
5065 \@nameuse{@thefnmark#1}%
5066 }%
5067 \insertparafootsepX{#1}%
5068 \ledsetnormalparstuff@common%

```

```

5069 {\csuse{notenumfontX@#1}%
5070 \csuse{notenumfontX@#1}%
5071 \wrapped@footfootmarkX{#1}%
5072 \strut%
5073 \csuse{wrapcontentX@#1}{#2}%
5074 \penalty-10}}
5075
5076 %

```

```

\para@footgroupX \para@footgroupX{<series>}
\mppara@footgroupX
5077 \newcommand*{\para@footgroupX}[1]{%
5078 \hsize=\expandafter\dimexpr\csuse{widthX@#1}\relax%
5079 \unvbox\csname footins#1\endcsname
5080 \ifcsstring{raggedX@#1}{L}{\RaggedLeft}{}%
5081 \ifcsstring{raggedX@#1}{R}{\RaggedRight}{}%
5082 \makehboxofhboxes
5083 \setbox0=\hbox{\unhbox0 \removehboxes}%
5084 \csuse{bhookgroupX@#1}
5085 \csuse{notefontsizeX@#1}
5086 \unhbox0\par}
5087
5088 \newcommand*{\mppara@footgroupX}[1]{%
5089 \setnoteswidthliketwocolumnsX@{#1}%
5090 \vskip\skip\@nameuse{mpfootins#1}
5091 \ifl@dpairing\ifparledgroup
5092 \leavevmode%
5093 \leavevmode\marks\parledgroup@{begin}%
5094 \marks\parledgroup@series{#1}%
5095 \marks\parledgroup@type{footnoteX}%
5096 \fi\fi\normalcolor
5097 \ifparledgroup%
5098 \ifl@dpairing%
5099 \else%
5100 \setnoteswidthliketwocolumnsX@{#1}%
5101 \setnotesXpositionliketwocolumns@{#1}%
5102 \print@footnoteXrule{#1}%
5103 \fi%
5104 \else%
5105 \setnoteswidthliketwocolumnsX@{#1}%
5106 \setnotesXpositionliketwocolumns@{#1}%
5107 \print@footnoteXrule{#1}%
5108 \fi%
5109 \unvbox\csname mpfootins#1\endcsname
5110 \ifcsstring{raggedX@#1}{L}{\RaggedLeft}{}%
5111 \ifcsstring{raggedX@#1}{R}{\RaggedRight}{}%
5112 \makehboxofhboxes
5113 \setbox0=\hbox{\unhbox0 \removehboxes}%
5114 \csuse{bhookgroupX@#1}%
5115 \csuse{notefontsizeX@#1}%

```

```

5116 \setparindentX{#1}%
5117 \unhbox0\par}}
5118
5119 %

```

Insertion of the footnotes separator The command `\insertparafootsepX{<series>}` must be called at the beginning of `\parafootftmX`.

```

\insertparafootsepX20 \newcommand{\insertparafootsepX}[1]{%
5121   \ifledRcol{%
5122     \ifnumequal{\csuse{prevpage#1@numR}}{\page@numR}%
5123     {\csuse{parafootsepX@#1}}%
5124     {}%
5125     \global\csname prevpage#1@numR\endcsname=\page@numR%
5126   \else%
5127     \ifboolexpr{bool{!@dprintingpages} or bool{!@dprintingcolumns} or bool
\ifnumequal{\csuse{prevpage#1@num}}{\page@num}%
5128     {\csuse{parafootsepX@#1}}%
5129     {}%
5130     \global\csname prevpage#1@num\endcsname=\page@num%
5131   }%We can't use directly the page counter, see https://tex.
stackexchange.com/a/239457/7712
5132   \global\expandafter\advance\csname footnote#1@parafootsepX\endcsname
by 1%
5133   \edef\@tmpa{parafootsepX@#1@the\csname footnote#1@parafootsepX\
endcsname}%
5134   \expandafter\edlabel\expandafter{\@tmpa}%
5135   \edef\@tmpb{\expandafter\xpagerref\expandafter{\@tmpa}}%
5136   \IfStrEq{\csuse{prevpage#1@num@outnumbering}}{\@tmpb}%
5137   {\csuse{parafootsepX@#1}}%
5138   {}%
5139   \global\cslet{prevpage#1@num@outnumbering}{\@tmpb}%
5140 }%
5141 \fi%
5142 }
5143 %
5144 %

```

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.

```

5145 \newcommand{\wrapped@footfootmarkX}[1]{%
5146   \ifdefined\hypertarget%
5147   \hyperlink%
5148     {@bodyfootmark#1@this@footnoteX@reading}%
5149     {@nameuse{footfootmark#1}}%

```



```

5150 \Hy@raisedlink{%
5151 \hypertarget%
5152 {@footnotemark#1@\this@footnoteX@reading}%
5153 {}%
5154 }%
5155 \else%
5156 \@nameuse{footfootmark#1}%
5157 \fi%
5158 }%
5159 %

```

`\wrapped@bodyfootmarkX` `\wrapped@bodyfootmarkX` prints the footnote mark of the text body, wrapped in hyperref package’s commands, if needed.

```

5160 \newcommand{\wrapped@bodyfootmarkX}[1]{%
5161 \ifdefined\hypertarget%
5162 \hyperlink%
5163 {@footnotemark#1@\expandafter\the\csname footnote#1@reading\
5164 endcsname}%
5165 {\@nameuse{bodyfootmark#1}}%
5166 \Hy@raisedlink{%
5167 \hypertarget%
5168 {@bodyfootmark#1@\expandafter\the\csname footnote#1@reading\
5169 endcsname}%
5170 {}%
5171 }%
5172 \else%
5173 \@nameuse{bodyfootmark#1}%
5174 \fi%
5175 }%
5176 %

```

XV Code common to both critical and familiar footnote in normal arrangement

`\par` should always be redefined to `\endgraf` within the format macro (this is what `\normal@pars` does), to override tricky material in the main text to get the lines numbered automatically (as set up by `\autopar`, for example).

In the case of footnote arranged in a “normal” way, we also must set some setting for paragraph indent and text direction when using Lua \TeX .

That why we have defined `\ledsetnormalparstuff@common` in order to make this setting for both familiar and critical notes. This command is called by command to make specific setting to critical or familiar footnote.

```

\ledsetnormalparstuff@common75 \newcommand*{\ledsetnormalparstuff@common}{%
\Xledsetnormalparstuff76 \ifluatex%
\ledsetnormalparstuffX77 \ifdefstring{\footnote@luatextextdir}{TLT}{}%

```

```

5178   {\textdir\footnote@luatexttextdir}%
5179   \pardir\footnote@luatexpardir%
5180   \fi%
5181   \csuse{\csuse{footnote@dir}}}%
5182   \normal@pars%
5183   \parfillskip \z@ \@plus 1fil}%
5184
5185 \newcommand*{\Xledsetnormalparstuff}[1]{%
5186   \ledsetnormalparstuff@common%
5187   \Xsetparindent{#1}%
5188   \iftoggle{Xparindent@#1}{\hspace{\parindent}}{}%
5189 }%
5190
5191 \newcommand*{\ledsetnormalparstuffX}[1]{%
5192   \ledsetnormalparstuff@common%
5193   \setparindentX{#1}%
5194   \iftoggle{parindentX@#1}{\hspace{\parindent}}{}%
5195 }%
5196 %

```

XVI Footnotes' width for two columns

We define here some commands which make sense only with `reledpar`, but must be called when defining notes parameters. These commands change the width of block notes to allow them to have the same size than two parallel columns.

`\old@hsize` These two commands are called at the beginning of critical or familiar notes groups. They set, if the option is enabled, the `\hsize`. They are also called at the on the setup for paragraphed notes.

`\setXnoteswidthliketwocolumns@`
`\setnoteswidthliketwocolumnsX@`

```

5197
5198 \newdimen\old@hsize%
5199 \AtBeginDocument{\old@hsize=\hsize}%
5200
5201 \newcommand{\setXnoteswidthliketwocolumns@}[1]{%
5202   \global\let\hsize@fornote=\hsize%
5203   \global\old@hsize=\hsize%
5204   \let\old@columnwidth=\columnwidth%
5205   \iftoggle{Xnoteswidthliketwocolumns@#1}%
5206     {%
5207       \setwidthliketwocolumns%
5208       \global\let\hsize@fornote=\hsize%
5209     }%
5210   {}%
5211   \let\hsize=\hsize@fornote%
5212   \let\columnwidth=\old@columnwidth%
5213 }%
5214
5215 \newcommand{\setnoteswidthliketwocolumnsX@}[1]{%

```

```

5216 \global\let\hsize@fornote=\hsize%
5217 \global\old@hsize=\hsize%
5218 \let\old@columnwidth=\columnwidth%
5219 \iftoggle{noteswidthliketwocolumnsX@#1}{%
5220   {%
5221     \setwidthliketwocolumns%
5222     \global\let\hsize@fornote=\hsize%
5223   }%
5224   {}%
5225   \let\hsize=\hsize@fornote%
5226   \let\columnwidth=\old@columnwidth%
5227 }%
5228
5229 %

```

`\setnotespositionliketwocolumns@` These two commands set the position of the critical / familiar footnotes, depending on
`\setXnotespositionliketwocolumns@` the hooks `Xnoteswidthliketwocolumns` and `noteswidthliketwocolumnsX`. They
 call commands which are defined only in `reledpar`, because this feature has no sens
 without `reledpar`.

```

5230 \newcommand{\setXnotespositionliketwocolumns@}[1]{%
5231   \iftoggle{Xnoteswidthliketwocolumns@#1}{%
5232     \csuse{setnotespositionliketwocolumns@\columns@position}%
5233   }{}%
5234 }%
5235
5236 \newcommand{\setnotesXpositionliketwocolumns@}[1]{%
5237   \iftoggle{noteswidthliketwocolumnsX@#1}{%
5238     \csuse{setnotespositionliketwocolumns@\columns@position}%
5239   }{}%
5240 }%
5241
5242 %

```

XVII Footnotes' order

`\fnpos` The `\fnpos` and `\mpfnpos` simply place their arguments in `\@fnpos` and `\@mpfnpos`,
`\mpfnpos` which will be used later in the output routine.

```

5243 \def\@fnpos{familiar-critical}
5244 \def\@mpfnpos{critical-familiar}
5245 \newcommand{\fnpos}[1]{\xdef\@fnpos{#1}}
5246 \newcommand{\mpfnpos}[1]{\xdef\@mpfnpos{#1}}
5247 %

```

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@Xfootnoterule%48 \newcommand{\print@Xfootnoterule}[1]{%
\print@footnotexrule%49 \vskip-\csuse{Xafterrule@#1}%Because count in \dimen\csuse{#1footins}
5250 \nointerlineskip%
5251 \moveleft-\leftskip\vbox{\csuse{#1footnoterule}}%
5252 \nointerlineskip%
5253 \vskip\csuse{Xafterrule@#1}%
5254 }%
5255
5256 \newcommand{\print@footnotexrule}[1]{%
5257 \vskip-\csuse{afterruleX@#1}%Because count in \dimen\csuse{footins#1}
5258 \nointerlineskip%
5259 \moveleft-\leftskip\vbox{\csuse{footnoterule#1}}%
5260 \nointerlineskip%
5261 \vskip\csuse{afterruleX@#1}%
5262 }%
5263
5264 %

```

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.

```

5265 \newtoggle{Xprenotes@}%
5266 \toggletrue{Xprenotes@}%
5267 \newcommand{\Xprenotes@}{0pt}%
5268 \newcommand*{\Xprenotes}[1]{\renewcommand{\Xprenotes@}{#1}}%
5269 \newcommand{\preXnotes}[1]{\led@warning@preXnotes@deprecated\Xprenotes{#1}}
    %For compatibility
5270 %

```

The same, but for familiar footnotes.

```

\Xprenotes 71 \newtoggle{prenotesX@}
\Xprenotes 72 \toggletrue{prenotesX@}
5273 \newcommand{\prenotesX@}{0pt}
5274 \newcommand*{\prenotesX}[1]{\renewcommand{\prenotesX@}{#1}}
5275 %

```

XIX.3 Internal commands

```

firstXseries@ 76 \gdef\firstXseries@{}
prepare@Xprenotes 77 \newcommand{\prepare@Xprenotes}[1]{%
5278   \ifdimequal{0pt}{\Xprenotes@}%
5279   }%
5280   {%
5281     \IfStrEq{\firstXseries@}{}{%
5282       \global\skip\csuse{#1footins}=\Xprenotes@%
5283       \global\advance\skip\csname #1footins\endcsname by\csuse{Xafterrule@
#1}%
5284       \gdef\firstXseries@{#1}%
5285     }%
5286     {%
5287       \ifseriesbefore{#1}{\firstXseries@}%
5288       {%
5289         \global\skip\csuse{#1footins}=\csuse{Xbeforenotes@\firstXseries@}%

```

```

5290 \global\advance\skip\csname #1footins\endcsname by\csuse{Xafterrule@
#1}%
5291 \gdef\firstXseries@{#1}%
5292 }%
5293 {}%
5294 }%
5295 }%
5296 }
5297 %

```

The same thing is required for familiar notes and `\prenotesX`.

```

firstseriesX@98 \gdef\firstseriesX@{}
prepare@prenotesX99 \newcommand{\prepare@prenotesX}[1]{%
5300 \ifdimequal{0pt}{\prenotesX@}%
5301 {}%
5302 {%
5303 \IfStrEq{\firstseriesX@}{}{%
5304 \global\skip\csuse{footins#1}=\prenotesX@%
5305 \global\advance\skip\csname footins#1\endcsname by\csuse{afterruleX@
#1}%
5306 \gdef\firstseriesX@{#1}%
5307 }%
5308 {%
5309 \ifseriesbefore{#1}{\firstseriesX@}%
5310 {%
5311 \global\skip\csuse{footins#1}=\csuse{beforenotesX@firstseriesX@}%
5312 \global\advance\skip\csname footins#1\endcsname by\csuse{afterruleX@
#1}%
5313 \gdef\firstXseries@{#1}%
5314 }%
5315 {}%
5316 }%
5317 }%
5318 }
5319 %

```

XX Endnotes

First, check the `noend` option.

```

5320 \ifbool{noend@}{}{%Used instead of \ifnoend@ to prevent expansion problem
5321 %

```

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 num-

bers 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.

```

5322 \newcommand{\l@dend@open}[1]{%
5323   \global\booltrue{l@dend@#1}%
5324   \expandafter\immediate%
5325   \expandafter\openout%
5326   \csname l@d@#1end\endcsname%
5327   =\l@auxdir\jobname.#1end\relax%
5328 }%
5329 \newcommand{\l@dend@close}[1]{%
5330   \global\boolfalse{l@dend@#1}%
5331   \expandafter\immediate%
5332   \expandafter\closeout\csname l@d@#1end\endcsname%
5333 }%
5334 %
5335 %

```

\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.

```

5336 \newcommand{\l@dend@stuff}{%
5337   \def\do##1{%
5338     \ifbool{l@dend@##1}{}%
5339     {\l@dend@open{##1}}%
5340     \expandafter\immediate\expandafter\write\csname l@d@##1end\endcsname{\
string\l@d@section{\the\section@num}\@percentchar}%
5341   }%
5342   \dolistloop{\@series}%
5343 }%
5344 %
5345 %

```

\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.

```

5346 \global\newbool{parapparatus@#4}\def\endprint#1#2#3#4#5#6#7{%
5347 \iftoggle{Xendfirstnote@#4}{%
5348 \global\togglefalse{Xendfirstnote@#4}%
5349 \csuse{Xendtxtbeforenotes@#4}%
5350 }{%
5351 \csuse{Xendbhooknote@#4}%
5352 \csuse{Xendnotefontsize@#4}%
5353 \hangindent=\csuse{Xendhangindent@#4}%
5354 \Xendstorelineinfo{#1}{#4}%
5355 \ifXendinsertsep%
5356 \hskip\csuse{Xendafternote@#4}\relax%
5357 \ifcsdef{prevendline#4}%
5358 {\ifcsequal{prevendline#4}{lineinfo@}%
5359 {\ifcsequal{Xendsymnum@#4}{\csuse{Xendsep@#4}}}%
5360 {\csuse{Xendsep@#4}}}%
5361 }%
5362 {\csuse{Xendsep@#4}}%
5363 \else%
5364 \iftoggle{Xendparagraph@#4}%
5365 {\global\Xendinsertsep@true}%
5366 }%
5367 \fi%
5368 \xdef\@currentseries{#4}%
5369 \def\do##1{%
5370 \setkeys{mac}{truefootnoteoption}{##1}%
5371 }%
5372 \notblank{#5}{\docsvlist{#5}}}%
5373 \IfStrEq{#6}{R}{\ledRcol@true}{}%
5374 \def\@this@crossref@start{#7:start}%
5375 \def\@this@crossref@end{#7:end}%
5376 \printlineendnote{#1}{#4}%
5377 \IfStrEq{#6}{R}{\ledRcol@false}{}%
5378 \nottoggle{Xendlemmadisablefontselection@#4}%
5379 {\select@lemmafont#1}%
5380 }%
5381 \bgroup%
5382 \csuse{Xendlemmafont@#4}%
5383 \csuse{Xendwraplemma@#4}{#2}%
5384 \egroup%
5385 \ifboolexpr{%
5386 tog1 {nosep@}%
5387 or test{\ifcsequal{Xendlemmaseparator@#4}}}%
5388 }%
5389 {\hskip\csuse{Xendinplaceoflemmaseparator@#4}\relax}%
5390 {\nobreak%
5391 \hskip\csuse{Xendbeforelemmaseparator@#4}%

```



```

5392 \csuse{Xendlemmaseparator@#4}%
5393 \hskip\csuse{Xendafterlemmaseparator@#4}%
5394 \relax%
5395 }%
5396 \csuse{Xendwrapcontent@#4}{#3}%
5397 \nottoggle{Xendparagraph@#4}{\par}{}%
5398 \def\do##1{%
5399 \setkeys[mac]{falsefootnoteoption}{##1}%
5400 }%
5401 \notblank{#5}{\docsvlist{#5}}{}%
5402 }}%
5403
5404 \let\l@d@section=\@gobble
5405
5406 %

```

\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.

```

5407 \newcommand{\Xendstorelineinfo}[2]{%
5408 \l@dp@rsefootspec#1|%
5409 \iftoggle{Xendnumberonlyfirstintwolines@#2}{%
5410 \xdef\lineinfo@{\l@dparsedstartpage - \l@dparsedstartline - \
5411 \l@dparsedstartsub - \l@dparsedendpage - \l@dparsedendline - \
5412 \l@dparsedendsub}%
5413 }%
5414 {%
5415 \xdef\lineinfo@{\l@dparsedstartpage - \l@dparsedstartline - \
5416 \l@dparsedstartsub}%
5417 }%
5418 }%
5419 %

```

\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.

```

5417 \newcommand{\printlineendnote}[2]{%
5418 \ifboolexpr{%
5419 togl {nonum@}%
5420 or togl {Xendnonumber@#2}%
5421 }%
5422 {\hspace{\csuse{Xendinplaceofnumber@#2}}}%
5423 {%
5424 \iftoggle{Xendnumberonlyfirstinline@#2}%
5425 {\ifcsdef{prevendline#2}%
5426 {\ifcsequal{prevendline#2}{lineinfo@}%
5427 }%

```

```

5428 \csuse{Xendbhookinplaceofnumber@#2}%
5429 \ifcsempy{Xendsymmlinenumber@#2}%
5430 {\hspace{\csuse{Xendinplaceofnumber@#2}}}%
5431 {\printsymmlineendnotearea{#2}}%
5432 \csuse{Xendahookinplaceofnumber@2}%
5433 }%
5434 {\printlineendnotearea{#1}{#2}}}%
5435 {\printlineendnotearea{#1}{#2}}}%
5436 }%
5437 {\printlineendnotearea{#1}{#2}}}%We keep every time line
5438 \csxdef{preendlinenum@#2}{\lineinfo@}%
5439 }%
5440 }%
5441 %

```

```

\printsymmlineendnotearea 42 \newcommand{\printsymmlineendnotearea}[1]{%
5443 \hspace{\csuse{Xendbeforelinenum@#1}}%
5444 \csuse{Xendnotenumfont@#1}%
5445 \ifdimequal{\csuse{Xendboxsymmlinenumber@#1}}{\z@}%
5446 {\csuse{Xendsymmlinenumber@#1}}%
5447 {\hbox to \csuse{Xendboxsymmlinenumber@#1}%
5448 {\csuse{Xendsymmlinenumber@#1}\hfill}%
5449 }%
5450 \hspace{\csuse{Xendaftersymmlinenumber@#1}}%
5451 }%
5452 %

```

\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.)

```

5453 \newcommand{\printlineendnotearea}[2]{%
5454 \csuse{Xendbhooklinenum@#2}%
5455 \hspace{\csuse{Xendbeforelinenum@#2}}%
5456 \bgroup%
5457 \csuse{Xendnotenumfont@#2}%
5458 \ifdimequal{\csuse{Xendboxlinenum@#2}}{\Opt}%
5459 {\printendlines#1||\ifledRcol@{\Rlineflag\fi}%
5460 {\leavevmode%
5461 \hbox to \csuse{Xendboxlinenum@#2}%
5462 {\%
5463 \IfSubStr{RC}{\csuse{Xendboxlinenumalign@#2}}{\hfill}{\fi}%
5464 \printendlines#1||\ifledRcol@{\Rlineflag\fi}%
5465 \IfSubStr{LC}{\csuse{Xendboxlinenumalign@#2}}{\hfill}{\fi}%
5466 }%
5467 \egroup%
5468 \hspace{\csuse{Xendafterlinenum@#2}}%
5469 \csuse{Xendahooklinenum@#2}%

```

```

5470 }%
5471 %

```

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 72 \newcommandx{\toendnotes}[2][1,usedefault]{%
\toendnotes*73 \ifboolexpr{bool{numbering} or bool{numberingR}}{%
5474 \def\do##1{%
5475 \expandafter\immediate\expandafter\write\csname l@d@##1end\endcsname%
5476 {\unexpanded{#2}\@percentchar}%
5477 }%
5478 \ifstrempy{#1}%
5479 {\dolistloop{\@series}}%
5480 {\docsvlist{#1}}%
5481 }{\led@err@toendnotes@outsidenumbering}%
5482 }%
5483 \WithSuffix\newcommandx\toendnotes*[2][1,usedefault]{%
5484 \ifboolexpr{bool{numbering} or bool{numberingR}}{%
5485 \def\do##1{%
5486 \expandafter\immediate\expandafter\write\csname l@d@##1end\endcsname%
5487 {#2\@percentchar}%
5488 }%
5489 \ifstrempy{#1}%
5490 {\dolistloop{\@series}}%
5491 {\docsvlist{#1}}%
5492 }{\led@err@toendnotes@outsidenumbering}%
5493 }%
5494 %

```

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.

```

5495 \newif\ifXendinsertsep@
5496 \newcommand*\doendnotes[1]{%
5497 \l@dend@close{#1}%
5498 \begingroup
5499 \csxdef{prevpagenum@#1}{}%
5500 \csxdef{prevpagerange@#1}{%

```

```

5501 \global\toggletrue{Xendfirstnote@#1}%
5502 \makeatletter
5503 \expandafter\let\csname #1end\endcsname=\endprint
5504 \input\l@auxdir\jobname.#1end%
5505 \global\Xendinsertsep@false%
5506 \endgroup}
5507 %

```

`\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.

```

5508 \newcommand*{\doendnotesbysection}[1]{%
5509 \l@dend@close{#1}%
5510 \csxdef{prevpagenum@#1}{}%
5511 \csxdef{prevpagerange@#1}{}%
5512 \global\expandafter\advance\csname #1end@bysection\endcsname by 1%
5513 \begingroup%
5514 \makeatletter%
5515 \def\l@d@section##1{%
5516 \global\toggletrue{Xendfirstnote@#1}%
5517 \ifnumequal{##1}{\csname #1end@bysection\endcsname}%
5518 {\cslet{#1end}{\endprint}}%
5519 {\cslet{#1end}{\@gobbleseven}}%
5520 }%
5521 \input\l@auxdir\jobname.#1end%
5522 \global\Xendinsertsep@false%
5523 \endgroup%
5524 }%
5525 %

```

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.

```

5526 }%
5527 %

```

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.

```

\setprintendlines28 \newif\ifl@d@Xendlinenumber%
\ifl@d@Xendlinenumber29 \newcommand*{\setprintendlines}[6]{%
5530   \let\@annot@start@print\relax%
5531   \let\@annot@end@print\relax%
5532   \l@d@pnumfalse%
5533   \l@d@dashfalse%
5534   \l@d@elinfalse%
5535   %

```

Should we print the line number? It depends on hooks and annotations.

```

5536   \l@d@Xendlinenumbertrue%
5537   \ifboolexpr{togl{Xendnolinenumber@\@currentseries}}%
5538     or (%
5539       togl{Xendnolinenumberifannotation@\@currentseries}%
5540       and not test{\ifdefvoid{\annot@start}}}%
5541       and not test{\ifdefvoid{\annot@end}}}%
5542     )%
5543   }{%
5544     \l@d@Xendlinenumberfalse%
5545   }{}%
5546   %

```

We print the second page number only if the ending page number is different

```

5547   \ifnum#4=#1 \else
5548     \l@d@pnumtrue
5549     \l@d@dashtrue
5550   \fi
5551   %

```

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.

```

5552   \ifboolexpr{%
5553     bool{\l@d@pnum}%
5554     or not test{\ifnumequal{#2}{#5}}}%
5555     and bool{\l@d@Xendlinenumber}%
5556   }{%
5557     \l@d@elintrue%
5558     \l@d@dashtrue%
5559     \l@d@elintrue
5560     \l@d@dashtrue
5561     \unless\ifx\relax\annot@end%
5562     \def\@annot@end@print{%

```

```

5563 \l@wrapcs@ifnotemptybox{Xendwraplinenumannotation@ \@currentseries
5564 }{\annot@end}%
5565 }%
5566 \fi%
5567 }%
5568 {}%
5569 %

```

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.

```

5569 \ifl@d@elin%
5570 \def \@annot@start@print{%
5571 \l@wrapcs@ifnotemptybox{Xendwraplinenumannotation@ \@currentseries}{\
5572 annot@start}%
5573 }%
5574 \else%
5575 \ifx \@annot@start\annot@end%
5576 \unless\ifx \@annot@start\relax%
5577 \def \@annot@start@print{%
5578 \l@wrapcs@ifnotemptybox{Xendwraplinenumannotation@ \@currentseries
5579 }{\annot@start}%
5580 }%
5581 \fi%
5582 \else%
5583 \ifx \@annot@end@print\relax%
5584 \def \@annot@start@print{%
5585 \l@wrapcs@ifnotemptybox{Xendwraplinenumannotation@ \@currentseries
5586 }{%
5587 \ifx \@annot@start\empty\else%
5588 \annot@start%
5589 \ifdefined\linangesep%
5590 \linangesep%
5591 \else%
5592 \csuse{Xendlinangeseparator@ \@currentseries}%
5593 \fi%
5594 \fi%
5595 \annot@end%
5596 }%
5597 \else%
5598 \let \@annot@start@print \@annot@end@print%
5599 \let \@annot@end@print \relax%
5600 \fi%
5601 \fi%
5602 %

```

We print the starting sub-line if it is nonzero.

```

5602 \l@d@ssubfalse

```

```

5603 \ifnum#3=0 \else
5604     \l@d@ssubtrue
5605 \fi
5606 %

```

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.

```

5607 \l@d@eslfalse
5608 \ifnum#6=0 \else
5609     \ifnum#6=#3
5610         \ifl@d@elin \l@d@esltrue \else \l@d@eslfalse \fi
5611     \else
5612         \l@d@esltrue
5613         \l@d@dashtrue
5614     \fi
5615 \fi%
5616 %

```

```

5617 \ifl@d@dash%
5618     \ifboolexpr{togl{fulllines@} or test{\ifcsemt{Xendtwolines@}\
@currentseries}}}%
5619     {%
5620     {%
5621     \setistwofollowinglines{#1}{#2}{#4}{#5}%
5622     \ifboolexpr{%
5623         (%
5624             togl {Xendtwolinesbutnotmore@\@currentseries}%
5625             and not%
5626             (%
5627                 bool {istwofollowinglines@}%
5628             )%
5629         )%
5630         or%
5631         (%
5632             (not test{\ifnumequal{#1}{#4}})%
5633             and togl{Xendtwolinesonlyinsamepage@\@currentseries}%
5634         )%
5635     }%
5636     {%
5637     {%
5638     \l@d@dashfalse%
5639     \l@d@Xtwolinestrue%
5640     \l@d@elinfalse%
5641     \l@d@eslfalse%
5642     \ifcsemt{Xendmorethantwolines@\@currentseries}%
5643     {%
5644     {\ifistwofollowinglines@\else%
5645         \l@d@Xmorethantwolinestrue%
5646     \fi%
5647     }%

```

```

5648     }%
5649   }%
5650 \fi%
5651 %

```

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

```

5652 \iftoggle{Xendnoidenticallinenumannotation@ \@currentseries}{%
5653   \ifx\annot@start\annot@end%
5654     \let\@annot@end@print\relax%
5655     \ifboolexpr{%Do not print the dash if we do not print the line number
5656       test{\ifdefempty{\linenumrep}}%
5657       or not bool {l@d@Xendlinenum}}%
5658     }{%
5659       \l@d@dashfalse%
5660     }{}%
5661   \fi%
5662 }{}%
5663 %

```

Finally, we check for `\Xendlinenumannotationonlyfirst` and `\Xendlinenumannotationonlyfirsti` and we redefine, if required, `\@annot@start@print` and `\annot@end@print`. We also store the current line number annotations.

```

5664 \iftoggle{Xendlinenumannotationonlyfirst@ \@currentseries}{%
5665   \ifboolexpr{%
5666     (%
5667       tog1{Xendlinenumannotationonlyfirstintwo@ \@currentseries}%
5668       and test{\ifcsequal{annot@start}{prevannot@start@ \@currentseries}}%
5669       and test{\ifcsequal{annot@end}{prevannot@end@ \@currentseries}}%
5670     )%
5671     or%
5672     (%
5673       not tog1{Xendlinenumannotationonlyfirstintwo@ \@currentseries}%
5674       test{\ifcsequal{annot@start}{prevannot@start@ \@currentseries}}%
5675     )%
5676   }{%
5677     \def\@annot@start@print{%
5678       \l@wrapcs@ifnotemptybox{Xendwraplinenumannotation@ \@currentseries
5679 }{\csuse{Xendsym\linenumannotation@ \@currentseries}}%
5680     }%
5681     \let\@annot@end@print\relax%
5682     \ifboolexpr{%Do not print the dash if do not print the line number
5683       test{\ifdefempty{\linenumrep}}%
5684       or not bool {l@d@Xendlinenum}}%
5685     }{%
5686       \l@d@dashfalse%
5687     }{}%
5688   }{}%

```



```

5688 \global\cslet{prevannot@start@\@currentseries}{\annot@start}%
5689 \global\cslet{prevannot@end@\@currentseries}{\annot@end}%
5690 }%
5691 {}%
5692 %

End of \setprintendlines.

5693 }%
5694 %

```

\printendlines Now we are ready to print it all.

```

5695 \def\printendlines#1|#2|#3|#4|#5|#6|#7|#8|{%
5696 \begingroup
5697 \setprintendlines{#1}{#2}{#3}{#4}{#5}{#6}%
5698 %

```

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.

```

5699 \ifdimequal{\csuse{Xendboxstartlinenum@\@currentseries}}{0pt}%
5700 {\bgroup}%
5701 {\leavevmode\hbox to \csuse{Xendboxstartlinenum@\@currentseries}\bgroup
\hfill}%
5702 %

```

Then, print the starting page number-

```

5703 \ifbool{expr}{%
5704 (%
5705 test{\ifcsstring{prevpagenum@\@currentseries}{#1}}%
5706 and not%
5707 (togl{Xendpagenumberonlyfirstifsingle@\@currentseries} and bool{
1@d@pnum})}%
5708 )%
5709 or%
5710 (%
5711 test {\ifcsstring{prevpagerange@\@currentseries}{#1-#4}}%
5712 )%
5713 }%
5714 {%
5715 \ifcsempy{Xendsympagenum@\@currentseries}%
5716 {\hspace{\csuse{Xendinplaceofpagenumber@\@currentseries}}}%
5717 {\csuse{Xendsympagenum@\@currentseries}}%
5718 }%
5719 {%
5720 \wrap@edcrossref{\@this@crossref@start}{\printnpnum{#1}}%
5721 }%
5722 %

```

Then, determine what must be printed before the start line.

```

5723 \ifl@d@dash%
5724 \ifl@d@pnum%
5725 \csuse{Xendlineprefixsingle@\@currentseries}%
5726 \else%
5727 \ifcseempty{Xendlineprefixmore@\@currentseries}%
5728 {\csuse{Xendlineprefixsingle@\@currentseries}}%
5729 {\csuse{Xendlineprefixmore@\@currentseries}}%
5730 \fi%
5731 \else%
5732 \csuse{Xendlineprefixsingle@\@currentseries}%
5733 \fi%
5734 %

```

Then print the starting line, followed, if needed, by the side flag and the starting sub line number, then the line number annotation.

```

5735 \ifcsstring{Xendlinenumberannotationposition@\@currentseries}{before}%
5736 {\@annot@start@print}%
5737 {}%
5738 \ifl@d@Xendlinenumber%
5739 \wrap@edcrossref{\@this@crossref@start}{%
5740 \ifledRcol%
5741 \linenumrepR{#2}%
5742 \else%
5743 \linenumrep{#2}%
5744 \fi%
5745 }%
5746 \fi%
5747 \iftoggle{Xendlineflag@\@currentseries}{\ifledRcol@\@Rlineflag\fi}{}%
5748 \ifl@d@Xendlinenumber%
5749 \ifl@d@ssub%
5750 \csuse{Xendsublinesep@\@currentseries}%
5751 \wrap@edcrossref{\@this@crossref@start}{%
5752 \ifledRcol%
5753 \sublinenumrepR{#3}%
5754 \else%
5755 \sublinenumrep{#3}%
5756 \fi%
5757 }%
5758 \fi%
5759 \fi%
5760 \ifcsstring{Xendlinenumberannotationposition@\@currentseries}{after}%
5761 {\@annot@start@print}%
5762 {}%
5763 %

```

Close the box.

```

5764 \egroup%
5765 %

```

Open the box for the ending line number.

```

5766 \ifdimequal{\csuse{Xendboxendlinenum@\@currentseries}}{Opt}%
5767 {\bgroup}%
5768 {\hbox to \csuse{Xendboxendlinenum@\@currentseries}\bgroup}%
5769 %

```

Print the dash + the ending line number, or the line number range symbol.

```

5770 \ifl@d@Xtwolines%
5771 \ifl@d@Xendlinenum%
5772 \ifl@d@Xmorethantwolines%
5773 \csuse{Xendmorethantwolines@\@currentseries}%
5774 \else%
5775 \csuse{Xendtwolines@\@currentseries}%
5776 \fi%
5777 \fi%
5778 \else%
5779 \ifl@d@dash%
5780 \ifdefined\linangesep%
5781 \linangesep%
5782 \else%
5783 \csuse{Xendlinangeseparator@\@currentseries}%
5784 \fi%
5785 \fi%
5786 %

```

Print the ending page number.

```

5787 \ifl@d@pnum%
5788 \ifcsstring{prevpagerange@\@currentseries}{#1-#4}%
5789 {%
5790 \ifcempty{Xendsympagenum@\@currentseries}%
5791 {\hspace{\csuse{Xendinplaceofpagenumber@\@currentseries}}}%
5792 {\csuse{Xendsympagenum@\@currentseries}}%
5793 }%
5794 {%
5795 \wrap@edcrossref{\@this@crossref@end}{\printnpnum{#4}}%
5796 }%
5797 \fi%
5798 %

```

Print the ending line number, with, if needed, the line prefix, and followed by the side flag, the subtitle number, and line number annotation.

```

5799 \ifcsstring{Xendlinenumannotationposition@\@currentseries}{before}%
5800 {\@annot@end@print}%
5801 {%
5802 \ifl@d@Xendlinenum%
5803 \ifl@d@elin%
5804 \ifl@d@pnum\csuse{Xendlineprefixsingle@\@currentseries}\fi%
5805 \wrap@edcrossref{\@this@crossref@end}{%
5806 \ifledRcol%

```

```

5807         \linenumrepR{#5}%
5808         \else%
5809         \linenumrep{#5}%
5810         \fi%
5811     }%
5812     \iftoggle{Xendlineflag@ \@currentseries}{\ifledRcol@ \@Rlineflag\fi
}{\}%
5813     \fi%
5814     \ifl@d@esl%
5815     \ifl@d@elin%
5816     \csuse{Xendsublinesep@ \@currentseries}%
5817     \fi%
5818     \wrap@edcrossref{\@this@crossref@end}{%
5819     \ifledRcol@%
5820     \sublinenumrepR{#6}%
5821     \else%
5822     \sublinenumrep{#6}%
5823     \fi%
5824     }%
5825     \fi%
5826     \fi%
5827     \ifcsstring{Xendlinenumannotationposition@ \@currentseries}{after}%
5828     {\@annot@end@print}%
5829     }%
5830     \fi%
5831 %

```

Close the ending line box.

```

5832     \ifdimequal{\csuse{Xendboxendlinenum@ \@currentseries}}{0pt}%
5833     {}%
5834     {\hfill}%Prevent underfull hbox
5835     \egroup%
5836 %

```

And, finally, save, if needed, the current page number for the Xendpagenumberonlyfirst hooks.

```

5837     \iftoggle{Xendpagenumberonlyfirst@ \@currentseries}%
5838     {\iftoggle{Xendpagenumberonlyfirstintwo@ \@currentseries}%
5839     {\csxdef{prevpagerange@ \@currentseries}{#1-#4}}%
5840     {\csxdef{prevpagenum@ \@currentseries}{#4}}%
5841     }%
5842     {}%
5843 %

```

Now, the end of \printendlines macro.

```

5844     \endgroup%
5845 }%
5846
5847 %

```

\printnpnum A macro to print a page number in an endnote. Should not be override anymore

```
5848 \newcommand*{\printnpnum}[1]{\csuse{Xendbeforepagenumber@\@currentseries
5849 }#1\csuse{Xendafterpagenumber@\@currentseries}}
5850 %
```

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@.

```
5851 \newcommand{\newseries}[1]{%
5852   \def\do##1{\newseries@{##1}}%
5853   \docsvlist{#1}
5854 }
5855 %
```

\@series The \series@ macro is an etoolbox list, which contains the name of all series.

```
5856 \newcommand{\@series}{}
5857 %
```

The command \newseries@\series creates a new series of the footnote.

\newseries@ \newcommand{\newseries@}[1]{
%

XXI.1 Test if series is still existing

```
5860 \xifinlist{#1}{\@series}{\led@warn@SeriesStillExist{#1}}%
5861 {%
5862 %
```

XXI.2 Init specific to reledpar

When calling \newseries@ after having loaded reledpar, we need to load specific setting.

```
5863 \ifdefined\newseries@par%
5864   \newseries@par{#1}%
5865 \fi%
5866 %
```

XXI.3 For critical footnotes

Critical footnotes are those which start with letters. We look for the `\nocritical` option of `reledmac`.

```
5867 \unless\ifnocritical@
5868 %
```

XXI.3.1 Options

```
5869 \newtoggle{Xlineflag@#1}
5870 \newtoggle{Xparindent@#1}
5871 \newtoggle{Xlemmadisablefontselection@#1}
5872 \csgdef{Xwrapcontent@#1}{}%
5873 \csgdef{Xbeforeinserting@#1}{}%
5874 \csgdef{Xhangindent@#1}{Opt}%
5875 \csgdef{Xragged@#1}{}%
5876 \csgdef{Xhsizetwocol@#1}{0.45 \hsize}%
5877 \csgdef{Xhsizethreecol@#1}{.3 \hsize}%
5878 \csgdef{Xcolalign@#1}{\raggedright}%
5879 \csgdef{Xnotenumfont@#1}{\normalfont}%
5880 \csgdef{Xnotefontsize@#1}{\footnotesize}%
5881 \csgdef{Xbhooknote@#1}{}%
5882 \csgdef{Xbhookgroup@#1}{}%
5883
5884 \csgdef{Xboxlinenum@#1}{Opt}%
5885 \csgdef{Xboxlinenumalign@#1}{L}%
5886
5887 \csgdef{Xboxstartlinenum@#1}{Opt}%
5888 \csgdef{Xboxendlinenum@#1}{Opt}%
5889
5890 \csgdef{Xboxsymlinenum@#1}{Opt}%
5891 \newtoggle{Xgroupbyline@#1}%
5892 \newtoggle{Xgroupbylineseparetwolines@#1}%
5893
5894 \newtoggle{Xnumberonlyfirstinline@#1}%
5895 \newtoggle{Xnumberonlyfirstintwolines@#1}%
5896
5897 \newtoggle{Xlinenumannotationonlyfirst@#1}%
5898 \newtoggle{Xlinenumannotationonlyfirstintwo@#1}%
5899
5900 \csgdef{Xtwolines@#1}{}%
5901 \csgdef{Xmorethantwolines@#1}{}%
5902 \csgdef{Xsublinesep@#1}{\fullstop}%
5903 \csgdef{Xpagelinesep@#1}{\csname Xsublinesep@#1\endcsname}%for
backward compatibility, call Xsublinesep@#1
5904 \newtoggle{Xtwolinesbutnotmore@#1}%
5905 \newtoggle{Xtwolinesonlyinsamepage@#1}%
5906 \newtoggle{Xonlypstart@#1}%
5907 \newtoggle{Xpstarteverytime@#1}%
```

```

5908 \newtoggle{Xpstart@#1}%
5909 \newtoggle{Xpstartonlyfirst@#1}%
5910 \csdef{Xprevpstart@#1}{0}%Not a hook, but a tool for the hook
5911 Xpstartonlyfirst
5912 \csgdef{Xpstartseparator@#1}{}%
5913
5914 \newtoggle{Xstanza@#1}%
5915 \newtoggle{Xstanzaonlyfirst@#1}%
5916 \csdef{Xprevstanza@#1}{0}%Not a hook, but a tool for the hook
5917 Xstanzaonlyfirst
5918
5919 \csgdef{Xstanzaseparator@#1}{}%
5920
5921 \csgdef{Xsymlinenum@#1}{}%
5922 \csgdef{Xsymlinenumannotation@#1}{}%
5923
5924 \newtoggle{Xnonote@#1}%
5925
5926 \newtoggle{Xnonumber@#1}%
5927 \newtoggle{Xnolinenumber@#1}%
5928 \newtoggle{Xnolinenumberifannotation@#1}%
5929 \csgdef{Xbeforenumber@#1}{Opt}%
5930 \csgdef{Xtxtbeforenumber@#1}{}%
5931 \csgdef{Xafternumber@#1}{0.5em}%
5932 \newtoggle{Xnonbreakableafternumber@#1}%
5933 \csgdef{Xbeforesymlinenum@#1}{\csuse{Xbeforenumber@#1}}%
5934 \csgdef{Xaftersymlinenum@#1}{\csuse{Xafternumber@#1}}%
5935 \csgdef{Xinplaceofnumber@#1}{1em}%
5936 \global\cslet{Xlemmaseparator@#1}{\rbracket}%
5937 \csgdef{Xbeforelemmaseparator@#1}{0em}%
5938 \csgdef{Xafterlemmaseparator@#1}{0.5em}%
5939 \csgdef{Xinplaceoflemmaseparator@#1}{1em}%
5940 \csgdef{Xbeforenotes@#1}{1.2em \@plus .6em \@minus .6em}%
5941 \csgdef{Xafterrule@#1}{Opt}%
5942
5943 \csgdef{Xtxtbeforenotes@#1}{%
5944 \newtoggle{Xtxtbeforenotes@#1@typeset}}%Not directly used by user,
5945 but internal
5946 \newtoggle{Xtxtbeforenotesonlyonce@#1}%
5947
5948 \csgdef{Xmaxhnotes@#1}{0.8\vsizex}
5949 \newtoggle{Xnoteswidthliketwocolumns@#1}%
5950 \csgdef{Xparafootsep@#1}{}%
5951 \csgdef{Xafternote@#1}{1em plus.4em minus.4em}%
5952 \csgdef{Xlinerrangeseparator@#1}{\endashchar}%
5953
5954 \csgdef{Xlemmafont@#1}{}%
5955 \csgdef{Xwraplemma@#1}{%
5956 \csgdef{Xwidth@#1}{\hsizex}%

```

```

5955 \csgdef{Xwraplinenumannotation@#1}{\textsuperscript}%
5956 \csgdef{Xlinenumannotationposition@#1}{after}%
5957 \expandafter\csgdef{Xinnotemark@#1}##1{\ledinnotemark{##1}}%
5958 \newtoggle{Xnoidenticallylinenumannotation@#1}%
5959 %

```

XXI.3.2 Create inserts, needed to add notes in foot

As regards inserts, see chapter 15 of *The TeXbook* by D. Knuth.

```

5960 \expandafter\newinsert\csname #1footins\endcsname%
5961 \unless\ifnoledgroup%
5962 \expandafter\newinsert\csname mp#1footins\endcsname%
5963 \fi%
5964 %

```

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.

```

5965 \global\newbool{parapparatus@}{\expandafter\newcommand\expandafter
*}{\expandafter\newcommand}\csname #1footnote\endcsname[2] []{%
5966 \nottoggle{Xnonote@#1}{%
5967 \if@edtext@secondarg%
5968 \ifledRcol%
5969 \ifcsstring{Xonlyside@#1}{L}{\
led@error@note@called@onrightside{#1footnote}}}%
5970 \else%
5971 \ifcsstring{Xonlyside@#1}{R}{\
led@error@note@called@onleftside{#1footnote}}}%
5972 \fi%
5973 \begingroup%
5974 \newcommand{\content}{##2}%
5975 \ifnumberedpar%
5976 \ifledRcol%
5977 \ifluatex%
5978 \footnotelang@lua[R]%
5979 \fi%
5980 \@ifundefined{xpg@main@language}%if polyglossia
5981 {}%
5982 {\footnotelang@poly[R]}%
5983 \footnoteoptions@{R}{##1}{true}%
5984 \xright@appenditem{%
5985 \ifbool{indtl@innote}%
5986 {\unexpanded{\let\index\nindex}}%
5987 {}%
5988 \ifbool{indtl@notenumber}%
5989 {\unexpanded{\let\index\nindex}}%There is no note
number... so
5990 {}%

```



```

5991         \noexpand\Xnote@true%
5992         \unexpanded{\gdef\@currentnotetype{critical}}%
5993         \noexpand\prepare@Xprenotes{#1}%
5994         \noexpand\prepare@edindex@fornote{\l@d@nums}%
5995         \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.
5996         \noexpand\setcounter{stanzaR}{\the\c@stanzaR}%Save
stanzaR counter for footnote
5997         \unexpanded{\def\@this@crossref@start}{\theedtext:
start}%
5998         \unexpanded{\def\@this@crossref@end}{\theedtext:end}%
5999         \expandonce{\@beforeinsertofthisedtext}% Internal for
now, no reason to make it public
6000         \noexpand\parse@annot\l@current@annotR|%
6001         \noexpand\csuse{v#1footnote}{#1}%
6002         {\l@d@nums}{\expandonce\@tag}{\expandonce\content}}
%
6003         \noexpand\Xnote@false%
6004         \unexpanded{\advance\@edindex@fornote@\m@ne}%
6005         \ifbool{indtl@innote}%
6006             {\unexpanded{\let\index\orig@index}}%
6007             {}%
6008         \ifbool{indtl@notenumber}%
6009             {\unexpanded{\let\index\orig@index}}%
6010             {}%
6011         }\to\inserts@listR
6012         \footnoteoptions@{R}{##1}{false}%
6013         \global\advance\insert@countR \@ne%
6014     \else%
6015         \ifluatex%
6016             \footnotelang@lua%
6017             \fi%
6018         \@ifundefined{xpg@main@language}%if polyglossia
6019             {}%
6020             {\footnotelang@poly}%
6021         \footnoteoptions@{L}{##1}{true}%
6022         \xright@appenditem{%
6023             \ifbool{indtl@innote}%
6024                 {\unexpanded{\let\index\nindex}}%
6025                 {}%
6026             \ifbool{indtl@notenumber}%
6027                 {\unexpanded{\let\index\nindex}}%There is no note
number... so
6028             {}%
6029         \noexpand\Xnote@true%
6030         \unexpanded{\gdef\@currentnotetype{critical}}%
6031         \noexpand\prepare@Xprenotes{#1}%
6032         \noexpand\prepare@edindex@fornote{\l@d@nums}%

```

```

6033 \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.
6034 \ifl@dpairing%
6035 \noexpand\setcounter{stanzaL}{\the\c@stanzaL}%Save
stanzaR counter for footnote
6036 \fi%
6037 \unexpanded{\def\@this@crossref@start}{\theedtext:
start}%
6038 \unexpanded{\def\@this@crossref@end}{\theedtext:end}%
6039 \expandonce{\@beforeinsertofthisedtext}%Internal for
now, no reason to make it public
6040 \noexpand\parse@annot\l@current@annot|%
6041 \noexpand\csuse{v#1footnote}%
6042 {#1}%
6043 {{\l@d@nums}{\expandonce\@tag}{\expandonce\content
}}%
6044 \noexpand\Xnote@false%
6045 \unexpanded{\advance\@edindex@fornote@m@one}%
6046 \ifbool{indtl@innote}%
6047 {\unexpanded{\let\index\orig@index}}%
6048 {}%
6049 \ifbool{indtl@notenumber}%
6050 {\unexpanded{\let\index\orig@index}}%
6051 {}%
6052 }\to\inserts@list
6053 \global\advance\insert@count \@ne%
6054 \footnoteoptions@{L}{#1}{false}%
6055 \fi
6056 \else
6057 \csuse{v#1footnote}{#1}{0|0|0|0|0|0|0|0}{#1}%
6058 \fi%
6059 \endgroup%
6060 \else%
6061 \led@err@FootnoteNotInSecondArgEdtext{#1}%
6062 \fi%
6063 }{\@noneed@Footnotetrue}% End of \nottoggle{Xnonote@#1}{%
6064 \ignorespaces%
6065 }
6066 %

```

Create counter used to determine on which page the previous note was called.

```

6067 \expandafter\newcount\csname #1prevpage@num\endcsname%
6068 \expandafter\newcount\csname #1prevpage@numR\endcsname%
6069 %

```

We need to be able to modify reledmac's footnote macros and restore their

```

6070 \global\csletcs{#1@footnote}{#1footnote}
6071 %

```

XXI.3.4 Set standard display

```

6072 \Xarrangement@normal{#1}%
6073 %

```

End of for critical footnotes.

```

6074 \fi
6075 %

```

XXI.4 For familiar footnotes

Familiar footnotes are those which end with letters. We look for the `nofamiliar` option of `reledmac`.

```

6076 \unless\ifnofamiliar@
6077 %

```

XXI.4.1 Options

```

6078 \newtoggle{parindentX@#1}
6079 \csgdef{wrapcontentX@#1}{}%
6080 \csgdef{hangindentX@#1}{Opt}%
6081 \csgdef{beforeinsertingX@#1}{}%
6082 \csgdef{raggedX@#1}{}%
6083 \csgdef{hsizetwocolX@#1}{0.45 \hsize}%
6084 \csgdef{hsizethreecolX@#1}{.3 \hsize}%
6085 \csgdef{colalignX@#1}{\raggedright}%
6086 \csgdef{notenumfontX@#1}{\normalfont}%
6087 \csgdef{notefontsizeX@#1}{\footnotesize}%
6088 \csgdef{bhooknoteX@#1}{}%
6089 \csgdef{bhookgroupX@#1}{}%
6090 \csgdef{afterruleX@#1}{Opt}
6091 \csgdef{beforenotesX@#1}{1.2em \@plus .6em \@minus .6em}
6092 \csgdef{maxhnotesX@#1}{0.8\vsizex}%
6093 \newtoggle{noteswidthliketwocolumnsX@#1}%
6094 \csgdef{parafootsepX@#1}{}%
6095 \csgdef{afternoteX@#1}{1em plus.4em minus.4em}
6096 \csgdef{widthX@#1}{\hsizex}%
6097 \expandafter\csgdef{innotemarkX@#1}##1{\ledinnotemark{##1}}%
6098 \csgdef{txtbeforenotesX@#1}{}%
6099 \newtoggle{nonoteX@#1}%
6100 \newtoggle{txtbeforenotesX@#1@typeset}%Not directly used by user,
but internal
6101 \newtoggle{txtbeforenotesonlyonceX@#1}%
6102 % End of for familiar footnotes.
6103 % \subsubsection{Create inserts, needed to add notes in foot}
6104 % As regards inserts, see chapter 15 of the TeXBook by D. Knuth.
6105 % \begin{macrocode}
6106 \expandafter\newinsert\csname footins#1\endcsname%
6107 \unless\ifnoledgroup@%

```

```

6108 \expandafter\newinsert\csname mpfootins#1\endcsname%
6109 \fi%
6110 %

```

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.

```

6111 \global\expandafter\newcommand\csname footnote#1\endcsname[2] []{%
6112 \nottoggle{nonoteX@#1}{%
6113 \begingroup%
6114 \prepare@prenotesX{#1}%
6115 \newcommand{\content}{##2}%
6116 %
6117 %

```

If we are using the \csquotes package, we reset the quotation level.

```

6118 \ifdefined\csq@qllevel%
6119 \csq@qllevel=0\relax%
6120 \fi%
6121 %

```

If we prepare parallel typesetting, we cannot just increase the footnote counter. Read reledpar's handbook about that (V.1.2 p. 55). If we have a manual footnote mark, use it.

```

6122 \global\expandafter\advance\csname footnote#1@reading\
6123 endcsname by \One%
6124 \ifstrempy{##1}{%
6125 \ifboolexpr{bool{l@dpairing} or bool{l@dprintingpages} or
6126 bool{l@dprintingcolumns}}{%
6127 \ifcsdef{footnote#1reading\the\csname footnote#1@reading
6128 \endcsname=typeset}%
6129 {\setcounter{footnote#1}{\csuse{footnote#1reading\the\
6130 csname footnote#1@reading\endcsname=typeset}}}%
6131 {\setcounter{footnote#1}{\the\csname footnote#1@reading\
6132 endcsname}}}%
6133 }{%
6134 \stepcounter{footnote#1}%
6135 }%
6136 }{}%
6137 %

```

We also have to check consistency with \onlysideX setting.

```

6133 \ifledRcol%
6134 \ifcsstring{onlysideX@#1}{L}{\
6135 led@error@note@called@onrightside{footnote#1}}{}%
6136 \else%
6137 \ifcsstring{onlysideX@#1}{R}{\
6138 led@error@note@called@onleftside{footnote#1}}{}%

```

```

6137         \fi%
6138 %

And now, the feature not depending on whether we prepare parallel typesetting

6139         \ifstreampy{##1}%
6140         {\protected@csxdef{@thefnmark#1}{\csuse{thefootnote#1}}}%
6141         {\protected@csxdef{@thefnmark#1}{##1}}%
6142         \nottoggle{nomk@}%Nomk is set to true when using \
footnoteXnomk with \parpackage
6143         {\csuse{@footnotemark#1}}%
6144         {}%
6145         \ifluatex%
6146         \xdef\footnote@luatextextdir{\the\textdir}%
6147         \xdef\footnote@luatexpardir{\the\pardir}%
6148         \fi%
6149         \if@ledgroup%
6150         \led@set@index@fornote{#1}%
6151         \fi%
6152         \csuse{vfootnote#1}{#1}{\expandonce\content}{##1}\
m@mmf@prepare%
6153         \ifbool{indtl@innote}%
6154         {\let\index\orig@@index}%
6155         {}%
6156         \ifbool{indtl@notenumber}%
6157         {\let\index\orig@@index}%
6158         {}%
6159         \endgroup%
6160         }{}%End of the notoggleX@#1
6161     }
6162 %

```

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. 55).

```

6163         \newcounter{footnote#1}
6164         \global\expandafter\renewcommand\csname thefootnote#1\endcsname{\
arabic{footnote#1}}
6165         \expandafter\newcount\csname footnote#1@reading\endcsname%
6166 %

```

Create counter used to determine on which page the previous note was called.

```

6167         \expandafter\newcount\csname prevpage#1@num\endcsname%
6168         \expandafter\newcount\csname prevpage#1@numR\endcsname%
6169         \expandafter\newcount\csname footnote#1@parafootsepX\endcsname%For
the case we use \parafootsepX outside of numbered section
6170 %

```

Add `\let\footnoteX@gobble to \no@expands`.

```

6171 \expandafter\gappto\expandafter\no@expands\expandafter{\expandafter\
let\csname footnote#1\endcsname\@gobble}%
6172 %

```

And now, define `\footnoteXmark` and `\footnoteXtext`, equivalent to classical `\footnotemark` and `\footnotetext`.

```

6173 \expandafter\newcommand\csname footnote#1mark\endcsname[1] []{%
6174 \begingroup%
6175 \prepare@prenotesX{#1}%
6176 \ifstrepty{##1}{%
6177 \stepcounter{footnote#1}%
6178 }{%
6179 \setcounter{footnote#1}{##1}%
6180 }%
6181 \protected@csxdef{@thefnmark#1}{\csuse{thefootnote#1}}%
6182 \csuse{@footnotemark#1}%
6183 \m@mmlf@prepare%
6184 \endgroup%
6185 }%
6186 \expandafter\newcommand\csname footnote#1text\endcsname[2] []{%
6187 \begingroup%
6188 \csuse{vfootnote#1}{#1}{\expandonce{##2}}{##1}%
6189 \endgroup%
6190 }%
6191 %

```

Do not forget to initialize the series.

```

6192 \arrangementX@normal{#1}%
6193 \fi
6194 %

```

XXI.5 The endnotes

Endnotes are commands like `\Xendnote`, where `X` is a series letter. First, we check for the `noend` options.

```

6195 \unless\ifnoend@
6196 %

```

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`

```

6197 \expandafter\newwrite\csname l@d@#1end\endcsname%
6198 \expandafter\newif\csname ifl@dend@#1\endcsname%
6199 %

```

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.

```

6200 \global\expandafter\newcommand\csname #1endnote\endcsname[2][1,
6201 usedefault]{%
6202     \bgroup%
6203     \newlinechar='40%
6204     \global\@noneed@Footnotetrue%
6205     \newcommand{\content}{##2}%
6206     \stepcounter{labidx}%
6207     \expandafter\immediate\expandafter\write\csname l@d@#1end\
endcsname{%
6208         \unexpanded{\def\sw@list@inedtext}%
6209         {\expandafter\unexpanded\expandafter{\sw@inthisedtext}}%
6210         \@percentchar\space%Explicit space, to add a linebreak in the
output file
6211         \noexpand\parse@annot\l@current@annot|\@percentchar\space%
6212         \expandafter\string\csname #1end\endcsname%
6213         {\ifnumberedpar@\l@d@nums\fi}%
6214         {\ifnumberedpar@\expandonce\@tag\fi}%
6215         {\expandonce\content}%
6216         {\#1}%
6217         {\unexpanded{##1}}%
6218         {\ifledRcol R\else L\fi}%
6219         {\theedtext}%
6220         \@percentchar%
6221     }%
6222     \egroup%
6223     \ignorespaces%
6224 }%
6225 %

```

XXI.5.3 Tools

The `\Xtoendnotes` command inserts any arbitrary content into the endnote file. It is an alias of the more generalist `\addtoendnotes`

```

6226 \global\expandafter\newcommand\csname #1toendnotes\endcsname[1]{%
6227     \toendnotes[#1]{##1}%
6228 }%
6229
6230
6231 \expandafter\WithSuffix\expandafter\newcommand\csname #1toendnotes\
endcsname*[1]{%
6232     \toendnotes*[#1]{##1}%

```

```

6233 }%
6234
6235 %

```

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.

```

6236
6237 \global\cslet{#1end}{\@gobbleseven}
6238 %

```

We need to store the number of times \doendnotesbysection is called for one series.

```

6239 \global\expandafter\newcount\csname #1end@bysection\endcsname%
6240 %

```

XXI.5.5 The options

```

6241 \csgdef{Xendwraplemma@#1}{%
6242 \csgdef{Xendwrapcontent@#1}{}%
6243 \csgdef{Xendtwolines@#1}{}%
6244 \csgdef{Xendmorethantwolines@#1}{}%
6245 \newtoggle{Xendtwolinesbutnotmore@#1}{}%
6246 \newtoggle{Xendtwolinesonlyinsamepage@#1}{}%
6247 \newtoggle{Xendlemmadisablefontselection@#1}{}%
6248 \csgdef{Xendnotenumfont@#1}{\normalfont}%
6249 \csgdef{Xendnotefontsize@#1}{\footnotesize}%
6250 \csgdef{Xendbhooknote@#1}{}%
6251
6252 \csgdef{Xendsublinesep@#1}{\fullstop}%
6253
6254 \csgdef{Xendbeforenumber@#1}{Opt}
6255 \csgdef{Xendafternumber@#1}{0.5em}
6256
6257 \csgdef{Xendboxlinenum@#1}{Opt}%
6258 \csgdef{Xendboxlinenumalign@#1}{L}%
6259
6260 \csgdef{Xendboxstartlinenum@#1}{Opt}%
6261 \csgdef{Xendboxendlinenum@#1}{Opt}%
6262
6263 \csgdef{Xendlemmaseparator@#1}{}%
6264 \csgdef{Xendbeforelemmaseparator@#1}{0em}%
6265 \csgdef{Xendafterlemmaseparator@#1}{0.5em}%
6266 \csgdef{Xendinplaceoflemmaseparator@#1}{0.5em}%

```



```

6267 \newtoggle{Xendparagraph@#1}%
6268 \csgdef{Xendafternote@#1}{1em plus.4em minus.4em}%
6269 \csgdef{Xendsep@#1}{}%
6270
6271
6272 \csgdef{Xendinplaceofnumber@#1}{Opt}%
6273 \newtoggle{Xendnonumber@#1}%
6274 \newtoggle{Xendnolinenumber@#1}%
6275 \newtoggle{Xendnolinenumberifannotation@#1}%
6276
6277 \csgdef{Xendhangindent@#1}{Opt}%
6278
6279 \newtoggle{Xendnumberonlyfirstinline@#1}%
6280 \newtoggle{Xendnumberonlyfirstintwolines@#1}%
6281
6282 \newtoggle{Xendlinenumannotationonlyfirst@#1}%
6283 \newtoggle{Xendlinenumannotationonlyfirstintwo@#1}%
6284
6285 \csgdef{Xendbeforesymmlinenum@#1}{\csuse{Xendbeforenumber@#1}}%
6286 \csgdef{Xendaftersymmlinenum@#1}{\csuse{Xendafternumber@#1}}%
6287
6288 \csgdef{Xendsymmlinenum@#1}{}%
6289 \csgdef{Xendsymmlinenumannotation@#1}{}%
6290
6291 \csgdef{Xendboxsymmlinenum@#1}{Opt}%
6292
6293 \csgdef{Xendbhooklinenumber@#1}{}%
6294 \csgdef{Xendehooklinenumber@#1}{}%
6295 \csgdef{Xendbhookinplaceofnumber@#1}{}%
6296 \csgdef{Xendehookinplaceofnumber@#1}{}%
6297
6298 \csgdef{Xendlinerangeseparator@#1}{\endashchar}%
6299
6300 \csgdef{Xendbeforepagenumber@#1}{p.}%
6301 \csgdef{Xendafterpagenumber@#1}{) }%
6302 \csgdef{Xendlineprefixsingle@#1}{}%
6303 \csgdef{Xendlineprefixmore@#1}{}%
6304
6305 \newtoggle{Xendlineflag@#1}
6306
6307 \csgdef{Xendlemmafont@#1}{}%
6308
6309 \csgdef{Xendlinenumannotationposition@#1}{after}%
6310 \csgdef{Xendwraplinenumannotation@#1}{\textsuperscript}%
6311 \newtoggle{Xendnoidenticallinenumannotation@#1}%
6312
6313 \newtoggle{Xendpagenumberonlyfirst@#1}%
6314 \newtoggle{Xendpagenumberonlyfirstifsingle@#1}%
6315 \newtoggle{Xendpagenumberonlyfirstintwo@#1}%
6316 \csgdef{Xendsympagenum@#1}{}%

```

```

6317 \csgdef{Xendinplaceofpagenumber@#1}{0pt}%
6318
6319 \csgdef{Xendtxtbeforenotes@#1}{}%
6320 \newtoggle{Xendfirstnote@#1}%Not a hook, but used to apply
Xendtxtbeforenotes
6321
6322 %
End of endnotes declaration
6323 \fi%
6324 %
Dump series in \@series
6325 \listxadd{\@series}{#1}
6326 }
6327 }% End of \newseries
6328 %

```

XXI.6 Init standards series (A,B,C,D,E)

```

6329 \expandafter\newseries\expandafter{\default@series}
6330 %

```

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.

```

6331 \newcommand{\seriesatbegin}[1]{%
6332 \StrDel{\@series}{#1}[\@series]%
6333 \edef\@new{}%
6334 \listead{\@new}{#1}%
6335 \listead{\@new}{\@series}%
6336 \xdef\@series{\@new}%
6337 }
6338 %

```

\seriesatend And `\seriesatend` moves the series to the end of the list.

```

6339 \newcommand{\seriesatend}[1]{%
6340 \StrDel{\@series}{#1}[\@series]%
6341 \edef\@new{}%
6342 \listead{\@new}{\@series}%
6343 \listead{\@new}{#1}%
6344 \xdef\@series{\@new}%
6345 }
6346 %

```

XXII.2 Test series order

`\ifseriesbefore` `\ifseriesbefore{<seriesA>}{<seriesB>}{<true>}{<false>}` expands to `<true>` if `<seriesA>` is printed before `<seriesB>`, or to `<false>` otherwise.

```

6347 \newcommand{\ifseriesbefore}[4]{%
6348   \StrPosition{\@series}{#1}[\@first]%
6349   \StrPosition{\@series}{#2}[\@second]%
6350   \ifnumgreater{\@second}{\@first}{#3}{#4}%
6351 }
6352 %

```

XXII.2.1 Get the first series

In some specific case, we need to know the first series of the list of series.

```

\@getfirstseries53 \newcommand{\@getfirstseries}{%
6354   \ifdefempty{\@series}%
6355   {\xdef\@firstseries{}}%
6356   {\StrChar{\@series}{1}[\@firstseries]}%
6357 }%
6358 %

```

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.

```

6359 \newcommandx{\settoggle@series}[5][4,5,usedefault]{%
6360   \def\do##1{%
6361     \ifcsdef{etb@tgl@#2@##1}{%
6362       \global\settoggle{#2@##1}{#3}%
6363     }{%
6364       \led@warn@BadHookSeries{##1}{#2}%
6365     }%
6366     \ifstrequal{#4}{critical}{
6367       \csuse{Xarrangement@}\csuse{series@display##1}{##1}%
6368     }{}
6369     \ifstrequal{#4}{familiar}{%
6370       \csuse{arrangementX@}\csuse{series@displayX##1}{##1}%
6371     }{}%
6372   }%
6373   \ifstreempty{#1}{%
6374     \dolistloop{\@series}%
6375     \ifstreempty{#5}{%
6376       \docsvlist{#5}%
6377     }
6378   }%
6379   {%
6380     \docsvlist{#1}%
6381   }%
6382 }
6383 %

```

`\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.

```

6384 \newcommandx{\setcommand@series}[6][4,5,6=0,usedefault]{%
6385   \def\do##1{
6386     \ifcsdef{#2@##1}{%
6387       \expandafter\global\expandafter\renewcommand\csname#2@##1\endcsname
6388       [#6]{#3}%
6389     }{%
6390       \led@warn@BadHookSeries{##1}{#2}%
6391     }%
6392     \ifstrequal{#4}{critical}{%
6393       \csuse{Xarrangement@\csuse{series@display##1}}{##1}%
6394     }{%
6395       \ifstrequal{#4}{familiar}{%
6396         \csuse{arrangementX@\csuse{series@displayX##1}}{##1}%
6397       }{%
6398         \ifstrequal{#1}{%
6399           \dolistloop{\@series}%
6400           \ifstreempty{#5}{%
6401             \docsvlist{#5}
6402           }
6403         }%
6404         {%
6405           \docsvlist{#1}%
6406         }%
6407       }%
6408     }%

```

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

```

6409 \newcommandx{\newhookcommand@series}[3][2,3=0,usedefault]{%
6410   \global\expandafter\newcommand\expandafter*\csname #1\endcsname[2][]{%
6411     \setcommand@series{##1}{#1}{##2}[][#2][#3]%
6412   }%
6413   \ifstreempty{#2}{%
6414     \def\do##1{%
6415       \global\expandafter\newcommand\expandafter*\csname #1##1\endcsname
6416       [1]{%

```

```

6416     \csuse{#1}[##1]{####1}%
6417   }%
6418 }%
6419 \docsvlist{#2}%
6420 }%
6421 }
6422 %

```

\newhooktoggle@series `\newhooktoggle@series` command names is a generic command to add new commands for a new toggle hook, like `\Xnumberonlyfirstinline`. The second argument is also used to create commands named `\<hookname><pseudoseris>`, like `\Xtwolinesbutnotmoreappref`.

```

6423 \newcommandx{\newhooktoggle@series}[2][2,usedefault]{%
6424   \global\expandafter\newcommandx\expandafter*\csname #1\endcsname[2][1,2={
true},usedefault]{%
6425     \settoggle@series{##1}{#1}{##2}[][#2]%
6426   }%
6427   \ifstrempy{#2}{-}{%
6428     \def\do##1{%
6429       \global\expandafter\newcommand\expandafter*\csname #1##1\endcsname{%
6430         \csuse{#1}[##1]%
6431       }%
6432     }%
6433     \docsvlist{#2}%
6434   }%
6435 }
6436 %

```

\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

```

6437 \newcommand{\newhooktoggle@series@reload}[2]{%
6438   \global\expandafter\newcommandx\expandafter*\csname #1\endcsname[2][1,2={
true},usedefault]{%
6439     \settoggle@series{##1}{#1}{##2}[#2]%
6440   }%
6441 }%
6442 %

```

\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.

```

6443 \newcommand{\newhookcommand@series@reload}[2]{%
6444   \global\expandafter\newcommand\expandafter*\csname #1\endcsname[2][]{%
6445     \setcommand@series{##1}{#1}{##2}[#2]%
6446   }%
6447 }
6448 %

```

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.

```

6449 \unless\ifnocritical@
6450   \newhookcommand@series{Xwrapcontent}%
6451   \newhookcommand@series{Xbeforeinserting}%
6452   \newhookcommand@series{Xlemmafont}%
6453   \newhookcommand@series{Xwraplemma}%
6454   \newhooktoggle@series{Xparindent}
6455   \newhookcommand@series{Xhangindent}
6456   \newhookcommand@series{Xragged}
6457   \newhookcommand@series{Xhsizetwocol}
6458   \newhookcommand@series{Xhsizethreecol}
6459   \newhookcommand@series{Xcolalign}%
6460   \newhookcommand@series{Xnotenumfont}
6461   \newhookcommand@series{Xbhooknote}
6462   \newhookcommand@series@reload{Xbhookgroup}{critical}
6463   \newhookcommand@series{Xboxsymlinenum}%
6464   \newhookcommand@series{Xsymlinenum}
6465   \newhookcommand@series{Xsymlinenumannotation}%
6466   \newhookcommand@series{Xbeforenumber}
6467   \newhookcommand@series{Xtxtbeforenumber}
6468   \newhookcommand@series{Xafternumber}
6469   \newhookcommand@series{Xbeforesymlinenum}
6470   \newhookcommand@series{Xaftersymlinenum}
6471   \newhookcommand@series{Xinplaceofnumber}
6472   \newhookcommand@series{Xlemmaseparator}
6473   \newhookcommand@series{Xbeforelemmaseparator}
6474   \newhookcommand@series{Xafterlemmaseparator}
6475   \newhookcommand@series{Xinplaceoflemmaseparator}
6476   \newhookcommand@series{Xtxtbeforenotes}
6477   \newhooktoggle@series{Xtxtbeforenotesonlyonce}%
6478   \newhookcommand@series@reload{Xafterrule}{critical}
6479   \newhooktoggle@series{Xnumberonlyfirstinline}
6480   \newhooktoggle@series{Xnumberonlyfirstintwolines}
6481   \newhooktoggle@series{Xlinenumannotationonlyfirst}%
6482   \newhooktoggle@series{Xlinenumannotationonlyfirstintwo}%
6483   \newhooktoggle@series{Xgroupbyline}%
6484   \newhooktoggle@series{Xgroupbylineseparatetwolines}%
6485   \newhooktoggle@series{Xnonumber}
6486   \newhooktoggle@series{Xnolinenum}%
6487   \newhooktoggle@series{Xnolinenumifannotation}%
6488
6489   \newhooktoggle@series{Xpstart}
6490   \newhooktoggle@series{Xpstartonlyfirst}%
6491   \newhooktoggle@series{Xpstarteverytime}%
6492   \newhookcommand@series{Xpstartseparator}%

```

```

6493 \newhooktoggle@series{Xstanza}%
6494 \newhooktoggle@series{Xstanzaonlyfirst}%
6495 \newhookcommand@series{Xstanzaseparator}%
6496
6497 \newhooktoggle@series{Xnonote}%
6498
6499 \newhooktoggle@series{Xonlypstart}
6500 \newhooktoggle@series{Xnonbreakableafternumber}
6501 \newhooktoggle@series{Xlemmadisablefontselection}
6502 \newhookcommand@series@reload{Xmaxhnotes}{critical}
6503 \newhookcommand@series@reload{Xbeforenotes}{critical}
6504 \newhooktoggle@series@reload{Xnoteswidthliketwocolumns}{critical}%
6505 \newhookcommand@series@reload{Xnotefontsize}{critical}
6506
6507
6508 \newhookcommand@series{Xboxlinenum}%
6509 \newhookcommand@series{Xboxlinenumalign}%
6510
6511 \newhookcommand@series{Xboxstartlinenum}%
6512 \newhookcommand@series{Xboxendlinenum}%
6513
6514 \newhookcommand@series{Xafternote}%
6515 \newhookcommand@series{Xparafootsep}
6516 \newhookcommand@series@reload{Xwidth}{critical}%
6517
6518 \newhookcommand@series{Xinnotemark}[] [1]%
6519
6520 \ifundef{\Xhsize}%
6521 {
6522   \newcommandx{\Xhsize}[2] [1,usedefault]{
6523     \led@warning@Xhsize@deprecated%
6524     \Xwidth[#1]{#2}%
6525   }%
6526 }%
6527 {}%
6528 \fi
6529 %

```

Now, hooks for critical notes which also apply to crossreferencing and line numbering at the sides of the page.

```

6530 \newhooktoggle@series{Xlineflag}[appref,SEref]
6531 \newhookcommand@series{Xtwolines}[appref,SEref]
6532 \newhookcommand@series{Xmorethantwolines}[appref,SEref]
6533 \newhookcommand@series{Xsublinesep}[appref,SEref,side]%
6534 \newhookcommand@series{Xpagelinesep}[appref,SEref,side]%
6535 \newhooktoggle@series{Xtwolinesbutnotmore}[appref,SEref]
6536 \newhooktoggle@series{Xtwolinesonlyinsamepage}[appref,SEref]
6537 \newhookcommand@series{Xlinerrangeseparator}[appref,SEref]
6538 \newhookcommand@series{Xlinenumannotationposition}[side,appref,SEref]%
6539 \newhookcommand@series{Xwraplinenumannotation}[side,ref,appref,SEref]%

```



```

6540 \newhooktoggle@series{Xnoidenticallinenumannotation}[side,ref,appref,Seref]
6541 %

```

XXII.3.5 Options for familiar notes

Before generating the optional commands for familiar notes, we check the `\nofamiliar` option.

```

6542 \unless\ifnofamiliar@
6543 \newhookcommand@series{wrapcontentX}%
6544 \newhookcommand@series{beforeinsertingX}%
6545 \newhooktoggle@series{parindentX}
6546 \newhookcommand@series{hangindentX}
6547 \newhookcommand@series{raggedX}
6548 \newhookcommand@series{hsizetwocolX}
6549 \newhookcommand@series{hsizethreecolX}
6550 \newhookcommand@series{colalignX}%
6551 \newhookcommand@series{notenumfontX}
6552 \newhookcommand@series{bhooknoteX}
6553 \newhookcommand@series@reload{bhookgroupX}{familiar}
6554 \newhookcommand@series@reload{beforenotesX}{familiar}
6555 \newhookcommand@series@reload{maxhnotesX}{familiar}
6556 \newhooktoggle@series@reload{noteswidthliketwocolumnsX}{familiar}%
6557 \newhookcommand@series@reload{afterruleX}{familiar}
6558 \newhookcommand@series@reload{notefontsizeX}{familiar}
6559 \newhookcommand@series{afternoteX}
6560 \newhookcommand@series{parafootsepX}
6561 \newhookcommand@series{txtbeforenotesX}%
6562 \newhooktoggle@series{txtbeforenotesonlyonceX}%
6563 \newhookcommand@series@reload{widthX}{familiar}%
6564 \newhookcommand@series{innotemarkX}[] [1]%
6565 \newhooktoggle@series{nonoteX}%
6566 \ifundef{\hsizeX}%
6567 {%
6568   \newcommandx{\hsizeX}[2][1,usedefault]{%
6569     \led@warning@hsizeX@deprecated%
6570     \widthX[#1]{#2}%
6571   }%
6572 }%
6573 {}%
6574 \fi
6575 %

```

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.

```

6576 \unless\ifnoend@
6577 \newhookcommand@series{Xendwraplemma}
6578 \newhookcommand@series{Xendwrapcontent}
6579 \newhookcommand@series{Xendnotenumfont}
6580 \newhookcommand@series{Xendlemmafont}%
6581 \newhookcommand@series{Xendbhooknote}
6582
6583 \newhookcommand@series{Xendboxlinenum}%
6584 \newhookcommand@series{Xendboxlinenumalign}%
6585
6586 \newhookcommand@series{Xendboxstartlinenum}%
6587 \newhookcommand@series{Xendboxendlinenum}%
6588
6589 \newhookcommand@series{Xendnotefontsize}
6590 \newhooktoggle@series{Xendlemmadisablefontselection}
6591 \newhookcommand@series{Xendlemmaseparator}
6592 \newhookcommand@series{Xendbeforelemmaseparator}
6593 \newhookcommand@series{Xendafterlemmaseparator}
6594 \newhookcommand@series{Xendinplaceoflemmaseparator}
6595
6596 \newhookcommand@series{Xendbeforenumber}%
6597 \newhookcommand@series{Xendafternumber}%
6598
6599 \newhooktoggle@series{Xendparagraph}
6600 \newhookcommand@series{Xendafternote}
6601 \newhookcommand@series{Xendsep}
6602
6603 \newhookcommand@series{Xendinplaceofnumber}%
6604 \newhooktoggle@series{Xendnonumber}%
6605 \newhooktoggle@series{Xendnolinenum}%
6606 \newhooktoggle@series{Xendnolinenumifannotation}%
6607
6608 \newhooktoggle@series{Xendnumberonlyfirstinline}%
6609 \newhooktoggle@series{Xendnumberonlyfirstintwolines}%
6610
6611 \newhooktoggle@series{Xendlinenumannotationonlyfirst}%
6612 \newhooktoggle@series{Xendlinenumannotationonlyfirstintwo}%
6613
6614 \newhookcommand@series{Xendsymmlinenum}%
6615 \newhookcommand@series{Xendbeforesymmlinenum}%
6616 \newhookcommand@series{Xendaftersymmlinenum}%
6617 \newhookcommand@series{Xendboxsymmlinenum}%
6618
6619 \newhookcommand@series{Xendsymmlinenumannotation}%
6620
6621 \newhookcommand@series{Xendbhooklinenum}%
6622 \newhookcommand@series{Xendahooklinenum}%
6623 \newhookcommand@series{Xendbhookinplaceofnumber}%
6624 \newhookcommand@series{Xendahookinplaceofnumber}%
6625

```

```

6626 \newhookcommand@series{Xendhangindent}%
6627
6628 \newhooktoggle@series{Xendpagenumberonlyfirst}%
6629 \newhooktoggle@series{Xendpagenumberonlyfirstifsingle}%
6630 \newhooktoggle@series{Xendpagenumberonlyfirstintwo}%
6631 \newhookcommand@series{Xendsympagenum}%
6632 \newhookcommand@series{Xendinplaceofpagenumber}%
6633
6634 \newhookcommand@series{Xendtxtbeforenotes}%
6635
6636 \fi
6637 \newhooktoggle@series{Xendlineflag}[apprefwithpage,Serefwithpage]
6638 \newhookcommand@series{Xendt看olines}[apprefwithpage,Serefwithpage]
6639 \newhookcommand@series{Xendmorethantwoines}[apprefwithpage,Serefwithpage]
6640 \newhooktoggle@series{Xendt看olinesbutnotmore}[apprefwithpage,Serefwithpage]
6641 \newhooktoggle@series{Xendt看olinesonlyinsamepage}[apprefwithpage,
6642 Serefwithpage]
6643 \newhookcommand@series{Xendlinrangeseparator}[apprefwithpage,Serefwithpage
6644 ]
6645 \newhookcommand@series{Xendbeforepagenumber}[apprefwithpage,Serefwithpage,
6646 Serefonlypage]
6647 \newhookcommand@series{Xendafterpagenumber}[apprefwithpage,Serefwithpage]
6648 \newhookcommand@series{Xendlineprefixsingle}[apprefwithpage,Serefwithpage]
6649 \newhookcommand@series{Xendlineprefixmore}[apprefwithpage,Serefwithpage]
6650 \newhookcommand@series{Xendsublinesep}[apprefwithpage,Serefwithpage]
6651
6652 \newhookcommand@series{Xendlinenumannotationposition}[apprefwithpage,
6653 Serefwithpage]%
6654 \newhookcommand@series{Xendwraplinenumannotation}[apprefwithpage,
6655 Serefwithpage]%
6656 \newhooktoggle@series{Xendnoidenticallylinenumannotation}[apprefwithpage,
6657 Serefwithpage]%
6658 %
6659 %

```

XXII.4 Hooks for a particular footnote

`\newhooktoggle@specific` `\newhooktoggle@specific` is a generic command to create boolean hook specific to a note.

```

6653 \newcommand{\newhooktoggle@specific}[1]{%
6654 \newtoggle{#1}%
6655 \listgadd{\hooktoggle@specific}{#1}%
6656 \define@key{mac}{truefootnoteoption}{#1}[]{\global\settoggle{#1}{true}}%
When enabling footnote option
6657 \define@key{mac}{falsefootnoteoption}{#1}[]{\global\settoggle{#1}{false
}}
6658 }
6659 %

```

`\newhookarg@specific` `\newhookarg@specific` is a generic command to create argumen hook specific to a note.

```

6660 \newcommand{\newhookarg@specific}[1]{%
6661   \listgadd{\hookarg@specific}{#1}%
6662   \define@key[mac]{truefootnoteoption}{#1}{\csgdef{#1@}{##1}}%When enabling
        footnote option
6663   \define@key[mac]{falsefootnoteoption}{#1}{\global\csundef{#1@}}%When
        disabling footnote option
6664 }
6665 %

```

`\hooktoggle@specific` 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.

```

6666 \def\hooktoggle@specific{%
6667 \newcommand{\add@hooktoggle@specific@to@cs}[1]{%
6668   \def\do##1{%
6669     \iftoggle{##1@}{%
6670       \ifcsdef{#1}{%
6671         \csgappto{#1}{\toggletrue{##1@}}%
6672       }{%
6673         \csgdef{#1}{\toggletrue{##1@}}%
6674       }%
6675     }{%
6676       \ifcsdef{#1}{%
6677         \csgappto{#1}{\togglefalse{##1@}}%
6678       }{%
6679         \csgdef{#1}{\togglefalse{##1@}}%
6680       }%
6681     }%
6682   }%
6683   \dolistloop{\hooktoggle@specific}%
6684 }%
6685 %

```

`\hookarg@specific` The same, but for optional argument of critical footnotes with assigned value.

```

\add@hookarg@specific@to@cs
6686 \def\hookarg@specific{%
6687 \newcommand{\add@hookarg@specific@to@cs}[1]{%
6688   \def\do##1{%
6689     \ifcsvoid{##1@}{%
6690       \ifcsdef{#1}{%
6691         \csxappto{#1}{%
6692           \noexpand\csdef%
6693             {##1@}{\csname##1@\endcsname}%
6694         }%
6695       }{%

```

```

6696 \csxdef{#1}{%
6697 \noexpand\csdef%
6698 {##1@}{\csname##1@\endcsname}%
6699 }%
6700 }%
6701 }%
6702 }%
6703 \dolistloop{\hookarg@specific}%
6704 }%
6705 %

```

And now, we define some hooks specific to a note.

```

6706 \newhooktoggle@specific{fulllines}%
6707 \newhooktoggle@specific{nonum}
6708 \newhooktoggle@specific{nosep}
6709 \newhooktoggle@specific{noprefix}%
6710 \newhooktoggle@specific{prefixmore}%
6711 \newhookarg@specific{linangesep}
6712 %

```

linangesep@ `\linangesep@` is defined by the option `linangesep` 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.

```

6713 %

```

\nomk@ `\nomk@` toggle is used by `reledpar` to remove the footnote mark in the text when using `\footnoteXmk`. Read `reledpar` handbook.

```

6714 \newtoggle{nomk@}%
6715 %

```

XXII.5 Alias

\Xnolemmaseparator `\Xnolemmaseparator[⟨series⟩]` is just an alias for `\Xlemmaseparator[⟨series⟩]{}`.

```

6716 \newcommandx*{\Xnolemmaseparator}[1][1]{\Xlemmaseparator[#1]}
6717 %

```

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.

```

6718 \newcommand*{\l@ddoxtrafeet}{%
6719   \IfStrEq{familiar-critical}{\@fnpos}
6720     {\do@feetX\do@Xfeet}%
6721     {%
6722       \IfStrEq{critical-familiar}{\@fnpos}%
6723         {\do@Xfeet\do@feetX}%
6724         {%
6725           \setbox\@outputbox \vbox{%
6726             \unvbox\@outputbox%
6727             \do@feet@custom@order}{\@fnpos}%
6728           }%
6729         }%
6730     }%
6731 }%
6732
6733 %

```

`\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, \LaTeX could perform an infinite run.

```

6734 \newcommand{\do@feet@custom@order}[2]{%
6735   \def\do##1{%
6736     \edef\@notesseries{\@firstoftwo##1}%
6737     \edef\@notetype{\@secondoftwo##1}%
6738     \ifdefstring{\@notetype}{critical}%
6739       {\csuse{#1append@Xnotes}{\@notesseries}}%
6740       {\ifdefstring{\@notetype}{familiar}%
6741         {\csuse{#1append@notesX}{\@notesseries}}%
6742         {}%
6743       }%
6744   }%
6745   \expandafter\docsvlist\expandafter{#2}%
6746 }%
6747 %

```

`\do@Xfeet` `\do@Xfeet` is the code extending `\@makecol` to cater to the extra critical feet.

```

6748 \newcommand*{\do@Xfeet}{%

```

```

6749 \setbox\@outputbox \vbox{%
6750 \unvbox\@outputbox
6751 \@opXfeet}}
6752 %

```

`\@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`

```

6753 \newcommand{\append@Xnotes}[1]{%
6754 \ifvoid\csuse{#1footins}\else%
6755 \global\skip\csuse{#1footins}=\csuse{Xbeforenotes@#1}%
6756 \global\advance\skip\csuse{#1footins} by\csuse{Xafterrule@#1}%
6757 \print@Xnotes{#1}%
6758 \fi%
6759 }%
6760 %

```

The normal way to add one series, `\print@Xnotes`, is replaced by `reledpar` when using `\Pages`.

```

6761 \newcommand\print@Xnotes[1]{%
6762 \xdef\@currentseries{#1}%
6763 \csuse{#1footstart}{#1}%
6764 \csuse{#1footgroup}{#1}%%
6765 }%
6766 %

```

We print all series of notes by looping on them. We check before printing them that they are not voided.

```

6767 \newcommand*\@opXfeet}{%
6768 \unless\ifnocritical@%
6769 \gdef\firstXseries@{}%
6770 \def\do##1{%
6771 \append@Xnotes{##1}%
6772 }%
6773 \dolistloop{\@series}%
6774 \fi%
6775 }%
6776 %

```

`\l@ddodoreinextrafeet` `\l@ddodoreinextrafeet` is the code catering for the extra footnotes within `\@reinserts`. We use the same category and ordering as in `\l@ddoxtrafeet`.

```

6777 \newcommand*\l@ddodoreinextrafeet}{%
6778 \IfStrEq{familiar-critical}{\@fnpos}
6779 {\@doreinfeetX\@doreinfeet}%
6780 {%
6781 \IfStrEq{critical-familiar}{\@fnpos}%
6782 {\X@doreinfeet\@doreinfeetX}%
6783 {\@doreinfeetX\X@doreinfeet}%

```

```

6784 }%
6785 }
6786
6787 %

```

`\X@doreinfeet` `\X@doreinfeet` is the code for catering for the extra critical footnotes within `\@reinserts`.

```

6788 \newcommand*{\X@doreinfeet}{%
6789   \unless\ifnocritical@%
6790     \def\do##1{%
6791       \ifvoid\csuse{##1footins}\else%
6792         \insert\csuse{##1footins}{\unvbox\csuse{##1footins}}%
6793       \fi}%
6794     \dolistloop{\@series}
6795   \fi%
6796 }
6797
6798 %

```

`\print@notesX` We have to add all the new kinds of familiar footnotes to the output routine. A macro
`\append@notesX` which appends the familiar footnotes of one series onto the output routine, also adding
`\do@feetX` vertical skip before notes.

```

\@doreinfeetX
6799 \newcommand{\append@notesX}[1]{%
6800   \ifvoid\csuse{footins#1}\else%
6801     \global\skip\csuse{footins#1}=\csuse{beforenotesX@#1}%
6802     \global\advance\skip\csuse{footins#1} by\csuse{afterruleX@#1}%
6803     \print@notesX{#1}%
6804   \fi%
6805 }%
6806 %

```

The normal way to print one series of notes. `\print@Xnotes` is replaced by `reledpar` when using `\Pages`.

```

6807 \newcommand\print@notesX[1]{%
6808   \xdef\@currentseries{#1}%
6809   \csuse{footstart#1}{#1}%
6810   \csuse{footgroup#1}{#1}%
6811 }%
6812 %

```

We print all the series of notes by looping over them. We check before printing them that they are not voided.

```

6813 \newcommand*{\do@feetX}{%
6814   \unless\ifnofamiliar@%
6815     \gdef\firstseriesX@{}%
6816     \setbox\@outputbox \vbox{%
6817       \unvbox\@outputbox%
6818       \def\do##1{%

```



```

6819     \append@notesX{##1}%
6820   }%
6821   \dolistloop{\@series}}%
6822 \fi%
6823 }%
6824
6825 \newcommand{\@doreinfeetX}{%
6826   \unless\ifnofamiliar@%
6827   \def\do##1{%
6828     \ifvoid\csuse{footins##1}\else
6829     \insert%
6830       \csuse{footins##1}
6831       {\unvbox\csuse{footins##1}}%
6832     \fi%
6833   }%
6834   \dolistloop{\@series}%
6835 \fi%
6836 }%
6837
6838 %

```

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.)

```

6839 \@ifclassloaded{memoir}{%
6840 %
6841 memoir is loaded so we use memoir's built in hooks.
6842
6843 \g@addto@macro{\m@doextrafeet}{\l@ddoxtrafeet}%
6844 \g@addto@macro{\m@mdodoreinextrafeet}{\l@ddodoreinextrafeet}%
6845 }{%
6846 %
6847

```

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`.

```

6845 \ifboolexpr{%
6846   test{\@ifpackageloaded{fancyhdr}}%
6847   and test {\ifdef{\latex@makecol}}%
6848 }{%
6849   \patchcmd%
6850     {\latex@makecol}%

```

```

6851     {\xdef\@freelist{\@freelist\@midlist}}}%
6852     {\xdef\@freelist{\@freelist\@midlist}\l@ddoxtrafeet}%
6853     }%
6854     {\led@error@fail@patch@@makecol}%
6855   }{%
6856   \patchcmd%
6857     {\@makecol}%
6858     {\xdef\@freelist{\@freelist\@midlist}}}%
6859     {\xdef\@freelist{\@freelist\@midlist}\l@ddoxtrafeet}%
6860     }%
6861     {\led@error@fail@patch@@makecol}%
6862   }%
6863
6864   \patchcmd%
6865     {\@reinserts}%
6866     {\ifvbox}%
6867     {\l@ddodoreinxtrafeet\ifvbox}%
6868     }%
6869     {\led@error@fail@patch@@reinserts}%
6870 }
6871
6872 %

```

It turns out that `\@doclearpage` also needs modifying.

`\if@led@nofoot` We have to check if there are any leftover feet.

```

6873 \newif\if@led@nofoot
6874
6875 %

```

```

6876 \@ifclassloaded{memoir}{%
6877 %

```

If the memoir class is loaded, we hook into its modified `\@doclearpage`.

```

\@mem@extranofeet78 \g@addto@macro{\@mem@extranofeet}{%%
6879   \def\do#1{%
6880     \unless\ifnocritical@%
6881       \ifvoid\csuse{#1footins}\else\@mem@nofootfalse\fi%
6882     \fi%
6883     \unless\ifnofamiliar@%
6884       \ifvoid\csuse{footins#1}\else\@mem@nofootfalse\fi%
6885     \fi%
6886   }
6887   \dolistloop{\@series}%
6888 }%
6889 }{%
6890 %

```

As memoir is not loaded we have patch \@doclearpage.

```

\@led@testifnofoot%91 \newcommand*{\@led@testifnofoot}{%
\@doclearpage%92   \@led@nofoottrue%
6893   \ifvoid\footins\else%
6894     \@led@nofootfalse%
6895   \fi%
6896   \def\do##1{%
6897     \unless\ifnocritical@%
6898       \ifvoid\csuse{##1footins}\else%
6899         \@led@nofootfalse%
6900       \fi%
6901     \fi%
6902     \unless\ifnofamiliar@%
6903       \ifvoid\csuse{footins##1}\else%
6904         \@led@nofootfalse%
6905       \fi%
6906     \fi%
6907   }%
6908   \dolistloop{\@series}%
6909 }%
6910
6911 \pretocmd%
6912 {\@doclearpage}%
6913 {\@led@testifnofoot}%
6914 {}%
6915 {\led@error@fail@patch@@doclearpage}%
6916
6917 \patchcmd%
6918 {\@doclearpage}%
6919 {\ifvoid\footins}%
6920 {\if@led@nofoot}%
6921 {}%
6922 {\led@error@fail@patch@@doclearpage}%
6923
6924 }
6925
6926 %

```

XXIV Page numbering in parallel typesetting

The `reledpar` package has two options which change the way page numbering works. We need to implement these options on `reledmac` and not on `reledpar` because they have some consequences for the `reledmac` auxiliary files (numbered file; see V.12 p. 134). The `sameparallelpagenunder` 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.

```

6927 \newcommand{\par@patch@thepage}{%
6928   \ifboolexpr{%
6929     bool{sameparallelpagenumber}%
6930     or bool{prevpgnotnumbered}%
6931   }%
6932   {%
6933     \patchcmd{\the page}%
6934       {page}{\par@page}%
6935       {}%
6936       {\led@error@fail@patch@thepage}%
6937   }{}%
6938 }%
6939
6940 \newcommand{\par@patch@pagenumbering}{%
6941   \setcounter{\par@page}{1}%
6942   \par@patch@thepage%
6943 }%
6944
6945 \ifl@dmemoir%
6946   \apptocmd{\@mempnum}%
6947     {\par@patch@pagenumbering}%
6948     {}%
6949     {\led@error@fail@patch@@mempnum}%
6950
6951 \else%
6952   \apptocmd{\pagenumbering}%
6953     {\par@patch@pagenumbering}%
6954     {}%
6955     {\led@error@fail@patch@pagenumbering}%
6956 \fi%
6957
```

```

6958
6959 \AtBeginDocument{\par@patch@thepage}%
6960 %

```

\@outputpage As its name says, \@outputpage is a \LaTeX macro called in the output routine. It is this macro which increases the page counter. We patch it in order to increase, conditionally, the par@page counter, and to set correctly the page number sent to the auxiliary files (.1, .2, etc.).

```

6961 \AtBeginDocument{%
6962   \set@this@c@page%
6963   \apptocmd{\@outputpage}{%
6964     \ifsameparallelpagenumber%
6965       \ifl@dprintingpages%
6966         \ifodd\c@page\else%
6967           \stepcounter{par@page}%
6968         \fi%
6969       \else%
6970         \stepcounter{par@page}%
6971       \fi%
6972     \else%
6973       \stepcounter{par@page}%
6974     \fi%
6975   }%
6976   {}%
6977   {\led@error@fail@patch@outputpage}%
6978 }%
6979 \catcode`\#=12%
6980 \AtBeginDocument{%
6981   \apptocmd{\setcounter}{%
6982     \IfStrEq{#1}{page}{\set@this@c@page}{}%
6983   }%
6984   {}%
6985   {\led@error@fail@patch@setcounter}%
6986   \apptocmd{\addtocounter}{%
6987     \IfStrEq{#1}{page}{\set@this@c@page}{}%
6988   }%
6989   {}%
6990   {\led@error@fail@patch@addtocounter}%
6991   \ifpackageloaded{calc}{%
6992     \apptocmd{\stepcounter}{%Because calc.sty does not use addtocounter in
stepcounter
6993       \IfStrEq{#1}{page}{\set@this@c@page}{}%
6994     }%
6995     {}%
6996     {\led@error@fail@patch@steptocounter}%
6997   }{}%
6998 }%
6999 \catcode`\#=6%
7000 %

```

`\thepar@page` The par@page counter.

```
7001 \newcounter{par@page}%
7002 \setcounter{par@page}{1}%
7003 %
```

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.

```
7004 \list@create{\labelref@list}
7005 %
```

`\zz@@@` Two convenience macros to zero three / four labeling counters in one go.

```
\zz@@@
7006 \newcommand*{\zz@@@}{000|000|000}% Set three counters to zero in one go
7007 \newcommand*{\zz@@@@}{000|000|000|000}% Set four counters to zero in one go
7008
7009 %
```

`\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.

```
7010 \newcommand*{\edlabel}[1]{%
7011   \leavevmode%
7012   \@bsphack%
```

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.

```

7013 \ifbool{expr{bool{ledRcol} or bool{ledRcol@}}}{%
7014   \ifXnote{%
7015     \protected@write\@auxout{%
7016       {\string\l@dmake@labelsR\space\thepage|\l@dparsestartline|\
7017       l@dparsestartsub||\annot@start|\the\c@pstartR|{#1}}}%
7018       \ifdef{\hypertarget}%
7019         {\Hy@raisedlink{\hypertarget{#1}{}}}%
7020         {}%
7021       \else%
7022         \write\linenum@outR{\string\@lab}%
7023         \ifx\labelref@listR\empty%
7024           \xdef\label@refs{\zz@@@}%
7025         \else%
7026           \gl@p\labelref@listR\to\label@refs%
7027         \fi%
7028         \ifvmode%
7029           \advancelabel@refs%
7030         \fi%
7031       }%

```

Use code from the kernel `\label` command to write the correct page number. Also define an `hypertarget` if `hyperref` package is loaded.

```

7031   \protected@write\@auxout{%
7032     {\string\l@dmake@labelsR\space\thepage|\label@refs|\the\c@pstartR
7033     |{#1}}}%
7034   \ifdef{\hypertarget}%
7035     {\Hy@raisedlink{\hypertarget{#1}{}}}%
7036     {}%
7037   \fi%
7038 }{%
7039   \ifl@dpairing%pstart or pstartL?
7040   \protected@write\@auxout{%
7041     {\string\l@dmake@labelsR\space\thepage|\l@dparsestartline|\
7042     l@dparsestartsub||\annot@start|\the\c@pstartL|{#1}}}%
7043     \ifdef{\hypertarget}%
7044       {\Hy@raisedlink{\hypertarget{#1}{}}}%
7045       {}%
7046     \else%
7047       \protected@write\@auxout{%
7048         {\string\l@dmake@labelsR\space\thepage|\l@dparsestartline|\
7049         l@dparsestartsub||\annot@start|\the\c@pstart|{#1}}}%
7050         \ifdef{\hypertarget}%
7051           {\Hy@raisedlink{\hypertarget{#1}{}}}%
7052           {}%
7053         \fi%
7054       \else%
7055         \write\linenum@out{\string\@lab}%
7056         \ifx\labelref@list\empty%
7057           \xdef\label@refs{\zz@@@}%

```

```

7056     \else%
7057         \gl@p\labelref@list\to\label@refs%
7058     \fi%
7059     \ifvmode%
7060         \advancelabel@refs%
7061     \fi%
7062     \ifl@dpairing%Pstart or PstartL?
7063         \protected@write\@auxout{}%
7064             {\string\l@dmake@labels\space\thepage|\label@refs|\the\c@pstartL
7065 |{#1}}%
7066         \ifdef{\hypertarget}%
7067             {\Hy@raisedlink{\hypertarget{#1}{}}}%
7068             {}%
7069         \else%
7070             \protected@write\@auxout{}%
7071                 {\string\l@dmake@labels\space\thepage|\label@refs|\the\c@pstart
7072 |{#1}}%
7073             \ifdef{\hypertarget}%
7074                 {\Hy@raisedlink{\hypertarget{#1}{}}}%
7075                 {}%
7076             \fi%
7077         \fi%
7078     }%
7079     \@esphack}%
7080 %

```

`\advancelabel@refs` In cases where `\edlabel` is the first element in a paragraph, we have a problem with line counts, because line counts change only at the first horizontal box of the paragraph. Hence, we need to test `\edlabel` if it occurs at the start of a paragraph. To do so, we use `\ifvmode`. If the test is true, we must advance by one unit the amount of text we write into the .aux file. We do so with the `\advancelabel@refs` command.

```

7080 \newcounter{line}%
7081 \newcounter{subline}%
7082 \newcounter{absline}%
7083 \newcommand{\advancelabel@refs}{%
7084     \setcounter{line}{\expandafter\labelrefsparseline\label@refs}%
7085     \stepcounter{line}%
7086     \setcounter{absline}{\expandafter\labelrefsparseabsline\label@refs}%
7087     \stepcounter{absline}%
7088     \ifsublines@%
7089         \setcounter{subline}{\expandafter\labelrefsparsesubline\label@refs}
7090 %
7091         \stepcounter{subline}{1}%
7092         \def\label@refs{\theline|\thesubline|\theabsline}%
7093     \else%
7094         \def\label@refs{\theline|0|\theabsline}%
7095     \fi%
7096 }

```



```

7096 \def\labelrefsparseline#1|#2|#3{#1}%
7097 \def\labelrefsparsesubline#1|#2|#3{#2}%
7098 \def\labelrefsparseabsline#1|#2|#3{#3}%
7099 %

```

\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.

```

7100 \newcommand*{\l@dmake@labels}{%
7101 \def\l@dmake@labels#1|#2|#3|#4|#5|#6|#7{%
7102   \expandafter\ifx\csname the@label\csuse{XR@prefix}#7\endcsname%
7103     \relax%
7104   \else%
7105     \led@warn@DuplicateLabel{\csuse{XR@prefix}#7}%
7106   \fi
7107   \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
7108   \csgdef{the@label\csuse{XR@prefix}#7}{#1|#2|#3|#4|#5|#6|\relax}%
7109   \ignorespaces}
7110 %
7111 %

```

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.

```

7112 \AtBeginDocument{%
7113   \def\l@dmake@labels#1|#2|#3|#4|#5|#6|#7{%
7114   }
7115 %
7116 %

```

\@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`.

```

7117 %
7118 \newcommand*{\@lab}{%

```

```

7119 \ifledRcol
7120 \xright@appenditem{\linenumr@p{\line@numR}}|{%
7121 \ifsublines@ \sublinenumr@p{\subline@numR}\else 0\fi|{%
7122 \the\absline@numR|{%
7123 \current@annot%
7124 }%
7125 \to\labelref@listR
7126 \else
7127 \xright@appenditem{%
7128 \linenumr@p{\line@num}}|{%
7129 \ifsublines@ \sublinenumr@p{\subline@num}\else 0\fi|{%
7130 \the\absline@num|{%
7131 \current@annot%
7132 }%
7133 \to\labelref@list
7134 \fi}
7135 %

```

`\applabel` `\applabel`, if called in `\edtext` will insert automatically both a starting and an ending label for the current edtext lines.

```

7136 \newcommand*{\applabel}[1]{%
7137 \if@edtext@secondarg%
7138 %

```

Label should not be already defined.

```

7139 \ifcsundef{the@label#1}{%
7140 \csdef{the@label#1}{applabel}%
7141 }%
7142 {%
7143 \led@warn@DuplicateLabel{#1 (applabel)}%
7144 }%
7145 %

```

Parse the `\edtext` line numbers and annotations.

```

7146 \expandafter\l@dp@rsefootspec\l@d@nums|{%
7147 \expandafter\parse@annot\l@current@annot|{%
7148 %

```

Use the \TeX standard hack for label.

```

7149 \@bsphack%
7150 %

```

And now, write the data in the auxiliary file.

```

7151 \ifledRcol%
7152 \protected@write\@auxout{%
7153 {\string\l@dmake@labelsR\space\l@dparsedstartpage|\l@dparsedstartline|\l@dparsedstartsub||\annot@start|\the\c@pstartR|{#1:start}}}%

```

```

7154 \ifdef{\hypertarget}%
7155   {\Hy@raisedlink{\hypertarget{#1:start}}}%
7156   {}%
7157   \protected@write\@auxout{%
7158     {\string\l@dmake@labelsR\space\l@dparsedendpage|\l@dparsedendline
7159     ||\l@dparsedendsub||\annot@end|\the\c@pstartR|{#1:end}}%
7159   \else%
7160     \ifl@dpairing%pstart or pstartL?
7161     \protected@write\@auxout{%
7162       {\string\l@dmake@labels\space\l@dparsedstartpage|\
7163       l@dparsedstartline|\l@dparsedstartsub||\annot@start|\the\c@pstartL|{#1:
7164       start}}%
7165     \ifdef{\hypertarget}%
7166       {\Hy@raisedlink{\hypertarget{#1:start}}}%
7167       {}%
7168       \protected@write\@auxout{%
7169         {\string\l@dmake@labels\space\l@dparsedendpage|\
7170         l@dparsedendline|\l@dparsedendsub||\annot@end|\the\c@pstartL|{#1:end}}%
7171       \else%
7172         \protected@write\@auxout{%
7173           {\string\l@dmake@labels\space\l@dparsedstartpage|\
7174           l@dparsedstartline|\l@dparsedstartsub||\annot@start|\the\c@pstart|{#1:start
7175           }}%
7176     \fi%
7177   \fi%
7178 %

```

Use the \TeX standard hack for label.

```

7179 \esphack%
7180 %

```

Warning if `\applabel` is called outside of `\edtext`.

```

7181 \else%
7182   \led@warn@AppLabelOutSecondArgEdtext{#1}%
7183   \fi%
7184 %

```

End of `\applabel`

```

7185 }%
7186 %

```

`\edlabels` `\edlabelS` and `\edlabelE` are just used to mark the beginning and the end of a passage.
`\edlabelE`
`\edlabelSE`

```

7187 \newcommand{\edlabelS}[1]{%
7188   \edlabel{#1:start}%
7189 }
7190 \newcommand{\edlabelE}[1]{%
7191   \edlabel{#1:end}%
7192 }
7193 \newcommand{\edlabelSE}[1]{%
7194   \edlabelS{#1}%
7195   \edlabelE{#1}%
7196 }
7197 %

```

\wrap@edcrossref `\wrap@edcrossref` is called around all `reledmac` crossref commands, except those which start with `x`. It adds the hyperlink.

```

7198 \newrobustcmd{\wrap@edcrossref}[2]{%
7199   \ifdef{\hyperlink}%
7200     {\hyperlink{#1}{#2}}%
7201     {#2}%
7202 }
7203 %

```

\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.

TeX already defines a `\pageref`, so changing the name to `\edpageref`.

```

7204 \newcommand*\edpageref[1]{\l@dref@undefined{#1}\wrap@edcrossref{#1}{\l@
7205   dgetref@num{1}{#1}}}
7206 \newcommand*\xpageref[1]{\l@dgetref@num{1}{#1}}
7207 %

```

\edlineref If the specified label exists, `\edlineref` gives its line number.

\xlineref

```

7208 \newcommand*\edlineref[1]{%
7209   \l@dref@undefined{#1}%
7210   \wrap@edcrossref{#1}{%
7211     \providetoggle{label@#1@ledRcol}%Required for the first run, when the
7212     label has not yet been parsed on the .aux file
7213     \iftoggle{label@#1@ledRcol}%
7214       {\linenumrepR{\l@dgetref@num{2}{#1}}}%
7215       {\linenumrep{\l@dgetref@num{2}{#1}}}%
7216     \xflagref{#1}%
7217   }%

```

```

7217 }%
7218 \newcommand*{\xlineref}[1]{\l@getref@num{2}{#1}}%
7219
7220 %

```

\sublineref If the specified label exists, \sublineref gives its sub-line number.

```

\sublineref
7221 \newcommand*{\sublineref}[1]{%
7222   \l@dref@undefined{#1}%
7223   \wrap@edcrossref{#1}{%
7224     \providetoggle{label@#1@dRcol}%Required for the first run, when the
label has not yet been parsed on the .aux file
7225     \iftoggle{label@#1@dRcol}%
7226       {\sublinenumrepR{\l@getref@num{3}{#1}}}%
7227       {\sublinenumrep{\l@getref@num{3}{#1}}}%
7228   }%
7229 }%
7230 \newcommand*{\xsublineref}[1]{\l@getref@num{3}{#1}}
7231
7232 %

```

\xabslineref If the specified label exists, \xabslineref gives its absolute line number. This is generally used only by reledmac internal macros.

```

7233 \newcommand*{\xabslineref}[1]{\l@getref@num{4}{#1}}%
7234 %

```

\annotationref If the specified label exists, \edannotationref gives the line number annotation with which it is associated.

```

7235 \newcommand*{\annotationref}[1]{%
7236   \l@dref@undefined{#1}%
7237   \wrap@edcrossref{#1}{%
7238     \l@wrap@ifnotemptybox{\Xwraplinenumannotation@ref}{%
7239       \l@getref@num{5}{#1}%
7240     }%
7241   }%
7242 }%
7243 \newcommand*{\xannotationref}[1]{%
7244   \l@getref@num{5}{#1}%
7245 }%
7246 %

```

\pstartref If the specified label exists, \pstartref gives its pstart number.

```

\pstartref
7247 \newcommand*{\pstartref}[1]{\l@dref@undefined{#1}\wrap@edcrossref{#1}{\l@getref@num{6}{#1}}}%
7248 \newcommand*{\xpstartref}[1]{\l@getref@num{6}{#1}}%
7249
7250 %

```

\xflagref \xflagref finds the side flag of any ref defined with \edlabel.

```
7251 \newcommand*\xflagref}[1]{\l@getref@num{7}{#1}}%
7252 %
```

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.

```
7253 \newcommand*\l@dref@undefined}[1]{%
7254   \expandafter\ifx\csname the@label#1\endcsname\relax
7255     \led@warn@RefUndefined{#1}%
7256   \fi}
7257
7258 %
```

\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 \l@lab macro above is defined to be a string of the type 123|456|789.

```
7259 \newcommand*\l@dgetref@num}[2]{%
7260   \expandafter
7261   \ifx\csname the@label#2\endcsname \relax
7262     000%
7263   \else
7264     \expandafter\expandafter\expandafter
7265     \l@dlabel@parse\csname the@label#2\endcsname|#1%
7266   \fi}
7267
7268 %
```

\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.)

```
7269 \newcommand*\l@dlabel@parse}{%
7270 \def\l@dlabel@parse#1|#2|#3|#4|#5|#6|#7|#8{%
7271   \ifcase #8%
7272     \or #1%
```

```

7273 \or #2%
7274 \or #3%
7275 \or #4%
7276 \or #5%
7277 \or #6%
7278 \or #7%
7279 \fi}
7280 %

```

\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.

```

7281 \newcommand*{\xxref}[2]{%
7282   {%
7283     \expandafter\ifx\csname the@label#1\endcsname\relax%
7284       \let\@tempa\zz@@%
7285       \def\@tempc{%
7286         \else%
7287         \def\@tempa{%
7288           \l@getref@num{1}{#1}|%
7289           \l@getref@num{2}{#1}|%
7290           \l@getref@num{3}{#1}%
7291         }%
7292         \edef\@tempc{%
7293           \l@getref@num{5}{#1}%
7294         }%
7295       \fi%
7296     \expandafter\ifx\csname the@label#2\endcsname\relax%
7297       \let\@tempb\zz@@%
7298       \def\@tempd{%
7299         \else%
7300         \def\@tempb{%
7301           \l@getref@num{1}{#2}|%
7302           \l@getref@num{2}{#2}|%
7303           \l@getref@num{3}{#2}%
7304         }%
7305         \edef\@tempd{%
7306           \l@getref@num{5}{#2}%
7307         }%
7308       \fi%
7309     \global\appto\@beforeinsertofthisedtext{%
7310       \def\@this@crossref@start{#1}%
7311     }%

```

```

7312 \global\appto\@beforeinsertofthisedtext{%
7313 \def\@this@crossref@end{#2}%
7314 }%
7315 \linenum{\@tempa|\@tempb}%
7316 \edef\@tempe{\@tempc|\@tempd}%
7317 \expandafter\lineannot\expandafter{\@tempe}%
7318 }%
7319 }%
7320
7321
7322 %

```

`\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`.

```

7323 \def\Xtwolines@appref{%
7324 \def\Xtwolines@SEref{%
7325
7326 \def\Xmorethantwolines@appref{%
7327 \def\Xmorethantwolines@SEref{%
7328
7329 \def\Xlinerangeseparator@appref{\endashchar}%
7330 \def\Xlinerangeseparator@SEref{\endashchar}%
7331
7332 \def\Xsublinesep@appref{\fullstop}%
7333 \def\Xsublinesep@SEref{\fullstop}%
7334
7335 \def\Xpagelinesep@appref{\fullstop}%
7336 \def\Xpagelinesep@SEref{\fullstop}%
7337
7338
7339 \newtoggle{Xtwolinesbutnotmore@appref}%
7340 \newtoggle{Xtwolinesbutnotmore@SEref}%
7341
7342 \newtoggle{Xtwolinesonlyinsamepage@appref}%
7343
7344 \newtoggle{Xtwolinesonlyinsamepage@SEref}%
7345
7346 \newtoggle{Xlineflag@appref}%
7347 \toggletrue{Xlineflag@appref}%Here exception
7348 \newtoggle{Xlineflag@SEref}%
7349 \toggletrue{Xlineflag@SEref}%%Here exception
7350
7351 \newtoggle{Xlinenumannotationonlyfirst@SEref}%Never changed, just for the
test in \setprintlines

```



```

7352 \newtoggle{Xlinenumannotationonlyfirst@appref}%Idem
7353 \newtoggle{Xnolinenumber@SEref}%Idem
7354 \newtoggle{Xnolinenumber@appref}%Idem
7355 \newtoggle{Xnolinenumberifannotation@SEref}%Idem
7356 \newtoggle{Xnolinenumberifannotation@appref}%Idem
7357
7358
7359 \def\Xlinenumannotationposition@appref{after}%
7360 \def\Xlinenumannotationposition@SEref{after}%
7361
7362 \def\Xwraplinenumannotation@appref{\textsuperscript}%
7363 \def\Xwraplinenumannotation@SEref{\textsuperscript}%
7364 \newtoggle{Xnoidenticallinenumannotation@appref}%
7365 \newtoggle{Xnoidenticallinenumannotation@SEref}%
7366
7367 \def\Xendtwolines@apprefwithpage{}%
7368 \def\Xendtwolines@SErefwithpage{}%
7369
7370 \def\Xendmorethantwolines@apprefwithpage{}%
7371 \def\Xendmorethantwolines@SErefwithpage{}%
7372
7373 \def\Xendlinerangeseparator@apprefwithpage{\endashchar}
7374 \def\Xendlinerangeseparator@SErefwithpage{\endashchar}
7375 \def\Xendlinerangeseparator@SErefonlypage{\endashchar}
7376
7377 \def\Xendbeforepagenumber@apprefwithpage{p.}%
7378 \def\Xendbeforepagenumber@SErefwithpage{p.}%
7379 \def\Xendbeforepagenumber@SEonlypage{p.}%
7380
7381 \def\Xendafterpagenumber@apprefwithpage{} }%
7382 \def\Xendafterpagenumber@SErefwithpage{} }%
7383
7384
7385 \def\Xendlineprefixsingle@apprefwithpage{}%
7386 \def\Xendlineprefixsingle@SErefwithpage{}%
7387
7388 \def\Xendlineprefixmore@apprefwithpage{}%
7389 \def\Xendlineprefixmore@SErefwithpage{}%
7390
7391 \newtoggle{Xendtwolinesbutnotmore@apprefwithpage}%
7392 \newtoggle{Xendtwolinesbutnotmore@SErefwithpage}%
7393
7394 \def\Xendsublinesep@apprefwithpage{\fullstop}%
7395 \def\Xendsublinesep@SErefwithpage{\fullstop}%
7396
7397 \newtoggle{Xendtwolinesonlyinsamepage@apprefwithpage}%
7398 \newtoggle{Xendtwolinesonlyinsamepage@SErefwithpage}%
7399
7400 \newtoggle{Xendlineflag@apprefwithpage}
7401 \toggletrue{Xendlineflag@apprefwithpage}%Here, exception

```

```

7402 \newtoggle{Xendlineflag@SErefwithpage}
7403 \toggletrue{Xendlineflag@SErefwithpage}%Here, exception
7404
7405
7406 \def\Xendlinenumannotationposition@apprefwithpage{after}%
7407 \def\Xendlinenumannotationposition@SErefwithpage{after}%
7408
7409 \def\Xendlinenumannotationposition@apprefwithpage{after}%
7410 \def\Xendlinenumannotationposition@apprefwithpage@SErefwithpage{after}%
7411
7412 \def\Xendwraplinenumannotation@apprefwithpage{\textsuperscript}%
7413 \def\Xendwraplinenumannotation@SErefwithpage{\textsuperscript}%
7414
7415 \newtoggle{Xendnoidenticallinenumannotation@apprefwithpage}%
7416 \newtoggle{Xendnoidenticallinenumannotation@SErefwithpage}%
7417
7418
7419 \newtoggle{Xendnolinenum@SErefwithpage}%Never changed, only for test in
    \setprintendlines
7420 \newtoggle{Xendnolinenum@apprefwithpage}%Idem
7421 \newtoggle{Xendnolinenum@ifannotation@SErefwithpage}%Idem
7422 \newtoggle{Xendnolinenum@ifannotation@apprefwithpage}%Idem
7423
7424 %

```

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.

```

7425
7426 \newtoggle{Xendlinenumannotationonlyfirst@SEref}%Never changed, just for
    the test in \setprintendlines
7427 \newtoggle{Xendlinenumannotationonlyfirst@SErefwithpage}%Idem
7428 \newtoggle{Xendlinenumannotationonlyfirst@appref}%Idem
7429 \newtoggle{Xendlinenumannotationonlyfirst@apprefwithpage}%Idem
7430
7431 \gdef\Xboxstartlinenum@appref{Opt}
7432 \gdef\Xboxstartlinenum@SEref{Opt}
7433
7434 \gdef\Xboxendlinenum@appref{Opt}
7435 \gdef\Xboxendlinenum@SEref{Opt}
7436
7437 \gdef\Xendboxstartlinenum@apprefwithpage{Opt}
7438 \gdef\Xendboxstartlinenum@SErefwithpage{Opt}
7439
7440 \gdef\Xendboxendlinenum@apprefwithpage{Opt}
7441 \gdef\Xendboxendlinenum@SErefwithpage{Opt}
7442
7443 \newtoggle{Xendpagenumberonlyfirst@apprefwithpage}
7444 \newtoggle{Xendpagenumberonlyfirst@SErefwithpage}
7445

```

```

7446 \newtoggle{Xendpagenumberonlyfirstifsingle@apprefwithpage}
7447 \newtoggle{Xendpagenumberonlyfirstifsingle@Serefwithpage}
7448
7449 \newtoggle{Xendpagenumberonlyfirstintwo@apprefwithpage}
7450 \newtoggle{Xendpagenumberonlyfirstintwo@Serefwithpage}
7451
7452 \gdef\Xendsympagenum@apprefwithpage{}
7453 \gdef\Xendsympagenum@Serefwithpage{}
7454
7455 \gdef\Xendinplaceofpagenumber@apprefwithpage{}
7456 \gdef\Xendinplaceofpagenumber@Serefwithpage{}
7457
7458 %

```

Now, declare the default values of \@apprefprefixsingle and \@apprefprefixmore, \@Serefprefix, \@Serefprefixmore and the commands which defines them.

```

7459 \newcommand\@apprefprefixsingle{}%
7460 \newcommand\@Serefprefixsingle{}%
7461
7462 \newcommand\@apprefprefixmore{}%
7463 \newcommand\@Serefprefixmore{}%
7464
7465 \newcommand{\setapprefprefixsingle}[1]{%
7466   \gdef\@apprefprefixsingle{#1}%
7467 }
7468 \newcommand{\setSerefprefixsingle}[1]{%
7469   \gdef\@Serefprefixsingle{#1}%
7470 }
7471
7472 \newcommand{\setapprefprefixmore}[1]{%
7473   \gdef\@apprefprefixmore{#1}%
7474 }
7475 \newcommand{\setSerefprefixmore}[1]{%
7476   \gdef\@Serefprefixmore{#1}%
7477 }
7478
7479 %

```

And not \setSerefonlypageprefixsingle and \setSerefonlypageprefixmore.

```

7480 \newcommand{\setSerefonlypageprefixsingle}[1]{%
7481   \gdef\Serefonlypage@prefixsingle{#1}%
7482 }%
7483 \newcommand{\setSerefonlypageprefixmore}[1]{%
7484   \gdef\Serefonlypage@prefixmore{#1}%
7485 }%
7486 %

```

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.

```

7487
7488 \newcommandx{\appref}[2][1,usedefault]{\reformatted@{#1}{#2}{appref}}
7489 \newcommandx{\Seref}[2][1,usedefault]{\reformatted@{#1}{#2}{Seref}}
7490
7491 \newcommandx{\apprefwithpage}[2][1,usedefault]{\reformattedwithpage@
7492 {#1}{#2}{appref}}
7493 \newcommandx{\Serefwithpage}[2][1,usedefault]{\reformattedwithpage@
7494 {#1}{#2}{Seref}}
7495
7496 \newcommand{\reformatted@}[3]{%
7497   \def\do##1{%
7498     \setkeys[mac]{truefootnoteoption}{##1}%
7499   }%
7500   \notblank{#1}{\docsvlist{#1}}{%}%
7501   \xdef\@currentseries{#3}%
7502   \iftoggle{noprefix@}{}%
7503   {%
7504     \ifcempty{@#3prefixmore}%
7505     {\@apprefprefixsingle}%
7506     {%
7507       \IfEq{\xlineref{#2:start}}{\xlineref{#2:end}}%
7508       {\iftoggle{prefixmore@}%
7509        {\csuse{@#3prefixmore}}%
7510        {\csuse{@#3prefixsingle}}}%
7511     }%
7512     {\csuse{@#3prefixmore}}%
7513   }%
7514 }%
7515 \ifboolexpr{%
7516   test{\ifcsundef{the@label#2:start}}%
7517   or test{\ifcsundef{the@label#2:end}}%
7518 }%
7519 {\led@warn@pairRefUndefined{#2}\nfss@text{\reset@font\bfseries ??}}%
7520 {%
7521   \def\@this@crossref@start{#2:start}%
7522   \def\@this@crossref@end{#2:end}%
7523   \xdef\annot@start{\xannotationref{#2:start}}%
7524   \xdef\annot@end{\xannotationref{#2:end}}%
7525   \printlines\xpageref{#2:start}|\xlineref{#2:start}|\xsublineref{#2:
start}|\xpageref{#2:end}|\xlineref{#2:end}|\xsublineref{#2:end}|\relax|
xflagref{#2:start}}%
7526 }%
7527 \def\do##1{%
7528   \setkeys[mac]{falsefootnoteoption}{##1}%

```

```

7529 }%
7530 \notblank{#1}{\docsvlist{#1}}{}%
7531 }%
7532
7533 \newcommand{\reformattedwithpage@}[3]{%
7534   \def\do##1{%
7535     \setkeys[mac]{truefootnoteoption}{##1}%
7536   }%
7537   \notblank{#1}{\docsvlist{#1}}{}%
7538   \xdef\@currentseries{#3withpage}%
7539   \ifboolexpr{%
7540     test{\ifcsundef{the@label#2:start}}%
7541     or test{\ifcsundef{the@label#2:end}}%
7542   }%
7543   {\led@warn@pairRefUndefined{#2}\nfss@text{\reset@font\bfseries ??}}%
7544   {%
7545     \def\@this@crossref@start{#2:start}%
7546     \def\@this@crossref@end{#2:end}%
7547     \xdef\annot@start{\xannotationref{#2:start}}%
7548     \xdef\annot@end{\xannotationref{#2:end}}%
7549     \printendlines\xpageref{#2:start}|\xlineref{#2:start}|\xsublineref{#2:
start}|\xpageref{#2:end}|\xlineref{#2:end}|\xsublineref{#2:end}|\relax\
xflagref{#2:start}|\%
7550   }%
7551   \def\do##1{%
7552     \setkeys[mac]{falsefootnoteoption}{##1}%
7553   }%
7554   \notblank{#1}{\docsvlist{#1}}{}%
7555 }%
7556
7557 \newcommand{\reformattedonlypage@}[3]{%
7558   \def\do##1{%
7559     \setkeys[mac]{truefootnoteoption}{##1}%
7560   }%
7561   \notblank{#1}{\docsvlist{#1}}{}%
7562   \xdef\@currentseries{#3onlypage}%
7563   \ifboolexpr{%
7564     test{\ifcsundef{the@label#2:start}}%
7565     or test{\ifcsundef{the@label#2:end}}%
7566   }%
7567   {\led@warn@pairRefUndefined{#2}\nfss@text{\reset@font\bfseries ??}}%
7568   {\ifnumequal{\xpageref{#2:end}}{\xpageref{#2:start}}%
7569     {%
7570       \ifcsvoid{#3onlypage@prefixsingle}%
7571       {}%
7572       {\csletcs{Xendbeforepagenumber@#3onlypage}{#3onlypage@prefixsingle
}}%
7573       \printnpnum{%
7574         \wrap@edcrossref{#2:start}{\xpageref{#2:start}}%
7575       }%

```

```

7576 }%
7577 {%
7578 \ifcsvoid{#3onlypage@prefixmore}%
7579 {}%
7580 {\csletcs{Xendbeforepagenumber@#3onlypage}{#3onlypage@prefixmore}}%
7581 \ifdefined\linangesep@%
7582 \printnpnum{%
7583 \wrap@edcrossref{#2:start}{\xpageref{#2:start}}%
7584 \linangesep@%
7585 \wrap@edcrossref{#2:end}{\xpageref{#2:end}}%
7586 }%
7587 \else%
7588 \printnpnum{%
7589 \wrap@edcrossref{#2:start}{\xpageref{#2:start}}%
7590 \csuse{Xendlinangeseparator@}\currentseries}%
7591 \wrap@edcrossref{#2:end}{\xpageref{#2:end}}%
7592 }%
7593 \fi%
7594 }%
7595 }%
7596 \def\do##1{%
7597 \setkeys[mac]{falsefootnoteoption}{##1}%
7598 }%
7599 \notblank{#1}{\docsvlist{#1}}}%
7600 }%
7601 %

```

`\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`.

```

7602 \newcommand*{\edmakelabel}[2]{\expandafter\xdef\csname the@label#1\
7603 endcsname{#2}}
7604 %

```

(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. 150 and V.9 p. 113), 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`.

```

7605 \pretocmd{\XR@test}%
7606   {\XR@test@mac+++#1#2#3#4+++}%
7607   {}%
7608   {}%
7609   %

```

`\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`.

```

7610 \long\def\xR@test@mac+++#1+++{\XR@test@mac@test#1}
7611 %

```

`\XR@test@mac@test` And finally, `\XR@test@mac@test` does the job. This code is based on the `\XR@test` macro of the `xr` package. However, note that the `\XR@prefix` is not called here, but it is integrated directly in `\l@dmake@labels` and `\l@dmake@labelsR`.

```

7612 \long\def\xR@test@mac@test#1#2...{%The triple dots (NOT \ldots) are because
7613   of the line 22 of xr.sty v5.02 1994/05/28
7614   \ifx#1\l@dmake@labels%
7615     \l@dmake@labels#2%
7616   \else
7617     \ifx#1\l@dmake@labelsR%
7618       \l@dmake@labelsR #2%
7619     \fi%
7620   \fi%
7621 }%

```

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.

```

7622 \pretocmd{\@xympar}%
7623   {\ifnumberedpar@
7624     \led@warn@NoMarginpars
7625     \esphack
7626   \else}%
7627   {}%
7628   {}%
7629
7630 \apptocmd{\@xympar}%
7631   {\fi}%

```

```

7632 {}
7633 {}
7634
7635 %

```

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

```

7636 \newcount\sidenote@margin
7637 \newcommand*{\sidenotemargin}[1]{\%
7638   \l@dgetsidenote@margin{#1}%
7639   \ifnum\@l@dttempcntb>\m@ne
7640     \ifledRcol
7641       \global\sidenote@marginR=\@l@dttempcntb
7642     \else
7643       \global\sidenote@margin=\@l@dttempcntb
7644     \fi
7645   \fi}}
7646 \newcommand*{\l@dgetsidenote@margin}[1]{\%
7647   \def\@tempa{#1}\def\@tempb{left}%
7648   \ifx\@tempa\@tempb
7649     \@l@dttempcntb \z@
7650   \else
7651     \def\@tempb{right}%
7652     \ifx\@tempa\@tempb
7653       \@l@dttempcntb \@ne
7654     \else
7655       \def\@tempb{outer}%
7656       \ifx\@tempa\@tempb
7657         \@l@dttempcntb \tw@
7658       \else
7659         \def\@tempb{inner}%
7660         \ifx\@tempa\@tempb
7661           \@l@dttempcntb \thr@@
7662         \else
7663           \led@warn@BadSidenotemargin
7664           \@l@dttempcntb \m@ne
7665         \fi

```



```

7666 \fi
7667 \fi
7668 \fi}
7669 \sidenotemargin{right}
7670
7671 %

```

`\l@dlp@rbox` We need two boxes to store sidenote texts.

```

\l@drp@rbox
7672 \newbox\l@dlp@rbox
7673 \newbox\l@drp@rbox
7674
7675 %

```

`\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
7676 \newdimen\ledlsnotewidth \ledlsnotewidth=\marginparwidth
\ledrsnotesep
7677 \newdimen\ledrsnotewidth \ledrsnotewidth=\marginparwidth
\ledlsnotefontsetup
7678 \newdimen\ledlsnotesep \ledlsnotesep=\linenumsep
\ledrsnotefontsetup
7679 \newdimen\ledrsnotesep \ledrsnotesep=\linenumsep
7680 \newcommand*{\ledlsnotefontsetup}{\raggedleft\footnotesize}
7681 \newcommand*{\ledrsnotefontsetup}{\raggedright\footnotesize}
7682
7683 %

```

`\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
7684 \newcommand*{\ledleftnote}[1]{\edtext{}{\l@dlsnote{#1}}}
7685 \newcommand*{\ledrightnote}[1]{\edtext{}{\l@drsnote{#1}}}
7686 \newcommand*{\ledsidenote}[1]{\edtext{}{\l@dcsnote{#1}}}%
7687 \newcommand*{\ledinnernote}[1]{\edtext{}{\l@disnote{#1}}}%
7688 \newcommand*{\ledouternote}[1]{\edtext{}{\l@dosnote{#1}}}%
7689 %

```

`\l@dlsnote` . The “footnotes” for left, right, and moveable sidenotes. The whole scheme is reminiscent of the critical footnotes code.

```

\l@drsnote
\l@dcsnote
7690 \newif\ifrightrightnoteup
\l@desnote
7691 \rightnoteuptrue
\l@disnote
7692
7693 \newcommand*{\l@dlsnote}[1]{%
7694 \begin{group}%
7695 \newcommand{\content}{#1}%
7696 \ifnumberedpar@
7697 \ifledRcol%
7698 \xright@appenditem{\noexpand\l@dlsnote{\expandonce\content}}}%

```

```

7699             \to\inserts@listR
7700             \global\advance\insert@countR \@ne%
7701         \else%
7702             \xright@appenditem{\noexpand\vl@dlsnote{\expandonce\content}}}%
7703             \to\inserts@list
7704             \global\advance\insert@count \@ne%
7705         \fi
7706     \fi%
7707     \ignorespaces%
7708 \endgroup%
7709 }%
7710
7711 \newcommand*{\l@drsnote}[1]{%
7712     \begingroup%
7713     \newcommand{\content}{#1}%
7714     \ifnumberedpar@
7715         \ifledRcol%
7716             \xright@appenditem{\noexpand\vl@drsnote{\expandonce\content}}}%
7717             \to\inserts@listR
7718             \global\advance\insert@countR \@ne%
7719         \else%
7720             \xright@appenditem{\noexpand\vl@drsnote{\expandonce\content}}}%
7721             \to\inserts@list
7722             \global\advance\insert@count \@ne%
7723         \fi
7724     \fi\ignorespaces%
7725 \endgroup%
7726 }%
7727
7728 \newcommand*{\l@dcsnote}[1]{%
7729     \begingroup%
7730     \newcommand{\content}{#1}%
7731     \ifnumberedpar@
7732         \ifledRcol%
7733             \xright@appenditem{\noexpand\vl@dcsnote{\expandonce\content}}}%
7734             \to\inserts@listR
7735             \global\advance\insert@countR \@ne%
7736         \else%
7737             \xright@appenditem{\noexpand\vl@dcsnote{\expandonce\content}}}%
7738             \to\inserts@list
7739             \global\advance\insert@count \@ne%
7740         \fi
7741     \fi\ignorespaces%
7742 \endgroup%
7743 }%
7744
7745 \newcommand*{\l@disnote}[1]{%
7746     \begingroup%
7747     \newcommand{\content}{#1}%
7748     \ifnumberedpar@%

```

```

7749 \ifledRcol%
7750 \xright@appenditem{\noexpand\vl@disnote{\expandonce\content}}}%
7751 \to\inserts@listR%
7752 \global\advance\insert@countR \@ne%
7753 \else%
7754 \xright@appenditem{\noexpand\vl@disnote{\expandonce\content}}}%
7755 \to\inserts@list%
7756 \global\advance\insert@count \@ne%
7757 \fi%
7758 \fi\ignorespaces%
7759 \endgroup%
7760 }%
7761
7762 \newcommand*{\l@dosnote}[1]{%
7763 \begingroup%
7764 \newcommand{\content}{#1}%
7765 \ifnumberedpar%
7766 \ifledRcol%
7767 \xright@appenditem{\noexpand\vl@dosnote{\expandonce\content}}}%
7768 \to\inserts@listR%
7769 \global\advance\insert@countR \@ne%
7770 \else%
7771 \xright@appenditem{\noexpand\vl@dosnote{\expandonce\content}}}%
7772 \to\inserts@list%
7773 \global\advance\insert@count \@ne%
7774 \fi%
7775 \fi\ignorespaces%
7776 \endgroup%
7777 }%
7778
7779 %

```

\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.

```

7780 \newcommand*{\vl@dlsnote}[1]{%
7781   \ifledRcol{%
7782     \@l@dttempcntb=\sidenote@marginR%
7783     \ifnum\@l@dttempcntb>\@ne%
7784       \advance\@l@dttempcntb by\page@numR%
7785     \fi%
7786   \else%
7787     \@l@dttempcntb=\sidenote@margin%
7788     \ifnum\@l@dttempcntb>\@ne%
7789       \advance\@l@dttempcntb by\page@num%
7790     \fi%
7791   \fi%
7792   \ifodd\@l@dttempcntb%
7793     \listgadd{\l@dcsnotetext@l}{#1}%
7794   \else%
7795     \listgadd{\l@dcsnotetext}{#1}%
7796   \fi
7797 }
7798 \newcommand*{\vl@drsnote}[1]{%
7799   \ifledRcol{%
7800     \@l@dttempcntb=\sidenote@marginR%
7801     \ifnum\@l@dttempcntb>\@ne%
7802       \advance\@l@dttempcntb by\page@numR%
7803     \fi%
7804   \else%
7805     \@l@dttempcntb=\sidenote@margin%
7806     \ifnum\@l@dttempcntb>\@ne%
7807       \advance\@l@dttempcntb by\page@num%
7808     \fi%
7809   \fi%
7810   \ifodd\@l@dttempcntb%
7811     \listgadd{\l@dcsnotetext}{#1}%
7812   \else%
7813     \listgadd{\l@dcsnotetext@r}{#1}%
7814   \fi%
7815 }
7816 \newcommand*{\vl@dcsnote}[1]{\listgadd{\l@dcsnotetext}{#1}}
7817
7818 \newcommand{\vl@disnote}[1]{%
7819   \ifledRcol{%
7820     \@tempcnta=\page@numR%
7821   \else%
7822     \@tempcnta=\page@num%
7823   \fi%
7824   \ifodd\@tempcnta% ODD => right page => inner side = left side
7825     \vl@dlsnote{#1}%
7826   \else%

```

```

7827 \vl@drsnote{#1}%
7828 \fi%
7829 }%
7830
7831 \newcommand{\vl@dosnote}[1]{%
7832 \ifledRcol%
7833 \@tempcnta=\page@numR%
7834 \else%
7835 \@tempcnta=\page@num%
7836 \fi%
7837 \ifodd\@tempcnta% ODD => right page => outer side = right side
7838 \vl@drsnote{#1}%
7839 \else%
7840 \vl@dlsnote{#1}%
7841 \fi%
7842 }%
7843
7844 %

```

`\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.

```

7845 \newcommand*\setl@dlp@rbox}[1]{%
7846 \begingroup%
7847 \parindent\z@\hsize=\ledlsnotewidth%
7848 \ledlsnotefontsetup%We kept it outside of the vbox, because can affect
the ragging
7849 \global\setbox\l@dlp@rbox%
7850 \ifleftnoteup%
7851 =\vbox to\z@{\ledlsnotefontsetup\vss #1}}%We put \
ledlsnotefontsetup inside footnote because required for color command. Note
the {} to keep setting local.
7852 \else%
7853 =\vbox to 0.70\baselineskip{\ledlsnotefontsetup\strut#1\vss}}%
7854 \fi%
7855 \endgroup%
7856 }
7857
7858 \newcommand*\setl@drp@rbox}[1]{%
7859 \begingroup%
7860 \parindent\z@\hsize=\ledrsnotewidth%
7861 \ledrsnotefontsetup%We kept it outside of the vbox, because can affect
the ragging
7862 \global\setbox\l@drp@rbox%
7863 \ifrighnoteup%
7864 =\vbox to\z@{\ledrsnotefontsetup\vss#1}}%We put \ledrsnotefontsetup
inside footnote because required for color command. Note the {} to keep
setting local.
7865 \else%

```

```

7866      =\vbox to0.7\baselineskip{{\ledrsnotefontsetup\strut#1\vss}}}%
7867      \fi%
7868      \endgroup%
7869    }%
7870    \newif\ifleftnoteup
7871      \leftnoteuptrue
7872    %

```

\@sidenotesep This macro is used to separate sidenotes of the same line.

```

7873 \newcommand{\setsidenotesep}[1]{\gdef\@sidenotesep{#1}}
7874 \newcommand{\@sidenotesep}{, }
7875 %

```

\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.

```

7876 \newcommand*{\affixside@note}{%
7877   \prepare@edindex@for@note{\the\page@num|\the\line@num|\the\subline@num|\the\page@num|\the\line@num|\the\subline@num}%
7878   \def\sidenotecontent@{}%
7879   \numgdef\itemcount@{0}%
7880   \def\do##1{%
7881     \ifnumequal{\itemcount@}{0}%
7882       {%
7883         \appto\sidenotecontent@{##1}}% Not print not separator before
the 1st note
7884       {\appto\sidenotecontent@{\@sidenotesep ##1}}%
7885     }%
7886     \numgdef\itemcount@{\itemcount@+1}%
7887   }%
7888   \dolistloop{\l@dcsonotetext}%
7889   \ifnumgreater{\itemcount@}{1}{\led@err@ManySidenotes}{}%
7890 %

```

And we do the same for left and right notes (not movable).

```

7891 \gdef\@templ@d{%
7892 \gdef\@templ@n{\l@dcsonotetext\l@dcsonotetext@l\l@dcsonotetext@r}%
7893 \ifx\@templ@d\@templ@n \else%
7894   \if@twocolumn%
7895     \if@firstcolumn%
7896       \setl@dlp@rbox{##1}{\sidenotecontent@}%
7897     \else%
7898       \setl@drp@rbox{\sidenotecontent@}%
7899     \fi%
7900   \else%

```

```

7901 \l@dttempcntb=\sidenote@margin%
7902 \ifnum\l@dttempcntb>\@ne%
7903 \advance\l@dttempcntb by\page@num%
7904 \fi%
7905 \ifodd\l@dttempcntb%
7906 \setl@drp@rbox{\sidenotecontent@}%
7907 \gdef\sidenotecontent@{}%
7908 \numgdef{\itemcount@}{0}%
7909 \dolistloop{\l@dcstotetext@l}%
7910 \ifnumgreater{\itemcount@}{1}{\led@err@ManyLeftnotes}{}%
7911 \setl@dlp@rbox{\sidenotecontent@}%
7912 \else%
7913 \setl@dlp@rbox{\sidenotecontent@}%
7914 \gdef\sidenotecontent@{}%
7915 \numgdef{\itemcount@}{0}%
7916 \dolistloop{\l@dcstotetext@r}%
7917 \ifnumgreater{\itemcount@}{1}{\led@err@ManyRightnotes}{}%
7918 \setl@drp@rbox{\sidenotecontent@}%
7919 \fi%
7920 \fi%
7921 \fi%
7922 \advance\@edindex@fornote@m@ne%
7923 }
7924 %

```

XXVII Minipages and such

We can put footnotes into minipages. The preparatory code has been set up earlier, all that remains is to ensure that it is available inside a minipage box. This requires some alteration to the kernel code, specifically the `\@iiminipage` and `\endminipage` macros. We will arrange this so that additional series can be easily added.

`\l@dfeetbeginmini` These will be the hooks in `\@iiminipage` and `\endminipage`.
`\l@dfeetendmini` They can be extended to handle other things if necessary.

```

7925 \ifnoledgroup@\else%
7926 \newcommand*\l@dfeetbeginmini{\@ledgrouptrue\l@dedbeginmini\l@dfambeginmini}
7927 \newcommand*\l@dfeetendmini}{%
7928 \IfStrEq{critical-familiar}{\@mpfnpos}%
7929 {\l@dedendmini\l@dfamendmini}%
7930 {%
7931 \IfStrEq{familiar-critical}{\@mpfnpos}%
7932 {\l@dfamendmini\l@dedendmini}%
7933 {\do@feet@custom@order{mp@}{\@mpfnpos}}}%
7934 }%
7935 }%
7936 %

```

`\l@dedbeginmini` These handle the initiation and closure of critical footnotes in a minipage environment.

`\l@dedendmini`

`\mp@append@Xnotes`

```

7937 \newcommand*{\l@dedbeginmini}{%
7938   \unless\ifnocritical@%
7939   \def\do##1{%
7940     \csletcs{v##1footnote}{mpv##1footnote}%
7941   }%
7942   \dolistloop{\@series}%
7943   \fi%
7944 }
7945 \newcommand*{\l@dedendmini}{%
7946   \unless\ifnocritical@%
7947   \ifl@dpairing%
7948     \ifledRcol%
7949       \flush@notesR%
7950     \else%
7951       \flush@notes%
7952     \fi%
7953   \fi
7954   \def\do##1{%
7955     \mp@append@Xnotes{##1}%
7956   }%
7957   \dolistloop{\@series}%
7958   \fi%
7959 }%
7960 \newcommand{\mp@append@Xnotes}[1]{%
7961   \ifvoid\csuse{mp#1footins}\else%
7962     \ifl@dpairing%
7963       \ifparledgroup%
7964         \ifledRcol%
7965           \dingdef{\parledgroup@beforenotesR}{\parledgroup@beforenotesR+\skip
7966             \@nameuse{mp#1footins}}%
7967         \else%
7968           \dingdef{\parledgroup@beforenotesL}{\parledgroup@beforenotesL+
7969             \skip\@nameuse{mp#1footins}}%
7970         \fi%
7971       \fi%
7972     \ifcsstring{series@display#1}{paragraph}{%
7973       \setbox\@nameuse{mp#1footins}=\vbox{%
7974         \csuse{Xnotefontsize@#1}%
7975         \ifcsdef{Xhsize}\csuse{series@display#1}@#1{%
7976           \hsize\csuse{Xhsize}\csuse{series@display#1}@#1}%
7977         }{}%
7978         \noindent\csuse{Xtxtbeforenotes@#1}%
7979         \unvbox\@nameuse{mp#1footins}%
7980         \@parboxrestore%
7981       }%
7982     \csuse{mp#1footgroup}{#1}%

```



```

7983 \fi%
7984 }%
7985 %

```

`\l@dfambeginmini` These handle the initiation and closure of familiar footnotes in a minipage environment.

```

\l@dfamendmini
\mp@append@notesX
7986 \newcommand*{\l@dfambeginmini}{%
7987 \unless\ifnofamiliar{%
7988 \def\do##1{\csletcs{vfootnote##1}{mpvfootnote##1}}%
7989 \dolistloop{\@series}%
7990 \fi%
7991 }%
7992
7993 \newcommand*{\l@dfamendmini}{%
7994 \unless\ifnofamiliar{%
7995 \def\do##1{%
7996 \mp@append@notesX{##1}%
7997 }%
7998 \dolistloop{\@series}%
7999 \fi%
8000 }%
8001 \newcommand{\mp@append@notesX}[1]{%
8002 \ifvoid\csuse{mpfootins#1}\else%
8003 \ifcsstring{series@displayX#1}{paragraph}{%
8004 \setbox\@nameuse{mpfootins#1}=\vbox{%
8005 \csuse{notefontsizeX@#1}%
8006 \ifcsdef{hsize}\csuse{series@display#1}X@#1}{%
8007 \hsize\csuse{hsize}\csuse{series@display#1}X@#1}%
8008 }{}%
8009 \noindent\csuse{txtbeforenotesX@#1}%
8010 \unvbox\@nameuse{mpfootins#1}%
8011 \@parboxrestore%
8012 }%
8013 }%
8014 \csuse{mpfootgroup#1}{#1}%
8015 \fi%
8016 }%
8017 %

```

`\@iiiminipage` This is our extended form of the kernel `\@iiiminipage` defined in `ltboxes.dtx`.

```

8018 \patchcmd%
8019 {\@iiiminipage}%
8020 {\let\@footnotetext\@mpfootnotetext}%
8021 {\let\@footnotetext\@mpfootnotetext\l@dfambeginmini}%
8022 {}%
8023 {\led@error@fail@patch@%iiiminipage}%
8024 %

```

`\endminipage` This is our extended form of the kernel `\endminipage` defined in `ltboxes.dtx`.

```

8025 \patchcmd%
8026   {\endminipage}%
8027   {\footnoterule}%
8028   {\footnoterule\l@advance@parledgroup@beforenormalnotes}%
8029   {}%
8030   {\led@error@fail@patch@endminipage}
8031
8032 \patchcmd%
8033   {\endminipage}%
8034   {\@minipagefalse}%
8035   {\l@dfeetendmini\@minipagefalse}%
8036   {}%
8037   {\led@error@fail@patch@endminipage}
8038
8039 %

```

`\l@dunboxmpfoot` `\@ldunboxmpfoot` insert normal footnotes for ledgroup.
`\l@advance@parledgroup@beforenormalnotes`

```

8040 \newcommand*{\l@dunboxmpfoot}{%
8041   \vskip\skip\@mpfootins
8042   \normalcolor
8043   \footnoterule
8044   \l@advance@parledgroup@beforenormalnotes
8045   \unvbox\@mpfootins%
8046 }
8047 %

```

When using parallel ledgroup, we need to store the vertical space added before footnote, in order to compensate them between left and right pages.

```

8048 \newcommand{\l@advance@parledgroup@beforenormalnotes}{%
8049   \ifparledgroup
8050     \ifl@dpairing
8051       \ifledRcol
8052         \dimgdef{\parledgroup@beforenotesR}{\parledgroup@beforenotesR\
skip\@mpfootins}
8053       \else
8054         \dimgdef{\parledgroup@beforenotesL}{\parledgroup@beforenotesL\
skip\@mpfootins}
8055       \fi
8056     \fi
8057   \fi
8058 }
8059 %

```

`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.

```

8060

```

```

8061 \newenvironment{ledgroup}{%
8062   \resetprevpage@num%
8063   \def\@mpfn{mpfootnote}\def\thempfn{\thempfootnote}\c@mpfootnote\z@%
8064   \let\@footnotetext\@mpfootnotetext
8065   \l@dfetbeginmini%
8066 }{%
8067   \par
8068   \unskip
8069   \ifvoid\@mpfootins\else
8070     \l@dunboxmpfoot
8071   \fi
8072   \l@dfetendmini%
8073   \@ledgroupfalse%
8074 }
8075
8076
8077 %

```

`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.

```

8078 \newenvironment{ledgroupsize}[2][1]{%
8079 %

```

Set the various text measures.

```

8080 \hspace #2\relax
8081 %

```

Initialize fills for centering.

```

8082 \let\ledllfill\hfil
8083 \let\ledrlfill\hfil
8084 \def\@tempa{#1}\def\@tempb{1}%
8085 %

```

Left adjusted numbered lines

```

8086 \ifx\@tempa\@tempb
8087 \let\ledllfill\relax
8088 \else
8089 \def\@tempb{r}%
8090 \ifx\@tempa\@tempb
8091 %

```

Right adjusted numbered lines

```

8092 \let\ledrlfill\relax
8093 \fi
8094 \fi
8095 %

```

Set up the footnoting.

```

8096 \def\@mpfn{mpfootnote}\def\thempfn{\thempfootnote}\c@mpfootnote\z@
8097 \let\@footnotetext\@mpfootnotetext
8098 \l@dfetbeginmini%
8099 }f%
8100 \par
8101 \unskip
8102 \ifvoid\@mpfootins\else
8103   \l@dunboxmpfoot
8104 \fi
8105 \l@dfetendmini%
8106 }
8107
8108 %

```

Close the \ifnoledgroup@else.

```

8109 \fi%
8110 %

```

`\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`.

```

8111 \newif\ifledgroupnotesL@
8112 \newif\ifledgroupnotesR@
8113 %

```

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`.

```

8114 \AtBeginDocument{%
8115   \unless\ifl@imakeidx%
8116     \@ifpackageloaded{imakeidx}{\led@error@PackageAfterEledmac{imakeidx}}{}
8117   %
8118   \fi%
8119   \unless\ifl@indextools%
8120     \@ifpackageloaded{indextools}{\led@error@PackageAfterEledmac{indextools}}{}
8121   %
8122   \fi%
8123   \unless\ifl@footmisc%
8124     \@ifpackageloaded{footmisc}{\led@error@PackageAfterEledmac{footmisc}}{}
8125   %
8126   \fi%
8127 }
8128 %

```

XXVIII.2 Auxiliary macros for `\edindex`

`\pagelinesep` In order to get a correct line number we have to use the label/ref mechanism. These
`\edindexlab` macros are for that.

```
\c@labidx
8126 \newcommand{\pagelinesep}{-}
8127 \newcommand{\edindexlab}{$&}
8128 \newcounter{labidx}
8129 \setcounter{labidx}{0}
8130
8131 %
```

`\doedindexlabel` This macro sets an `\edlabel`.

```
8132 \newcommand{\doedindexlabel}{%
8133   \stepcounter{labidx}%
8134   \edlabel{\edindexlab\thelabidx}%
8135 }
8136
8137 %
```

`\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`.

```
8138 \newcounter{pageline}%
8139 \renewcommand{\thepageline}{%
8140   \thepage%
8141   \pagelinesep%
8142   \xlineref{\edindexlab\thelabidx}%
8143 }
8144 %
```

`\thestartpageline` These macros make up the page/line start/end number when the `\edindex` command
`\theendpageline` is called in critical notes.

```
8145 \newcommand{\thestartpageline}{%
8146   \l@dparsedstartpage%
8147   \pagelinesep%
8148   \l@dparsedstartline%
8149 }
8150 \newcommand{\theendpageline}{%
8151   \l@dparsedendpage%
8152   \pagelinesep%
8153   \l@dparsedendline%
8154 }
8155 %
```

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.

```
8156 \newcount\@edindex@fornote@
8157 %
```

`\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 |.

```
8158 \newcommand{\prepare@edindex@fornote}[1]{%
8159   \l@dp@rsefootspec#1|}%
8160   \advance\@edindex@fornote@\@ne%
8161 }
8162 %
```

`\get@edindex@ledinnote@command` The `\get@edindex@ledinnote@command` macro defines a `\@ledinnote@command` command which is added as an attribute (text inserted after |) of the next index entry. Consequently, we write the definition of the location reference attribute in the .xdy file.

```
8163 \newcommand{\get@edindex@ledinnote@command}{%
8164   \ifxindy%
8165     \gdef\@ledinnote@command{%
8166       ledinnote\thelabidx%
8167     }%
8168   \ifxindyhyperref%
8169     \immediate\write\eledmac@xindy@out{%
8170       (define-attributes ("ledinnote\thelabidx"))^^J
8171       \space\space(markup-locref^^J
8172       \eledmacmarkuplocrefdepth^^J
8173       :open "\string\ledinnote[\edindexlab\thelabidx]{\@index@command
8174       }\@currentseries}{\@currentnotetype}"^^J
8175       :close "}"^^J
8176       :attr "ledinnote\thelabidx"^^J
8177     )
8178   \else%
8179     \immediate\write\eledmac@xindy@out{%
8180       (define-attributes ("ledinnote\thelabidx"))^^J
8181       \space\space(markup-locref^^J
8182       \eledmacmarkuplocrefdepth^^J
8183       :open "\string\ledinnote{\@index@command}{\@currentseries}{\
8184       @currentnotetype}"^^J
8185       :close "}"^^J
8186       :attr "ledinnote\thelabidx"^^J
8187     }
```

```

8186         )
8187     }%
8188     \fi%
8189 %

```

If we do not use xindy option, \@ledinnote@command will produce something like ledinnote{formattingcommand}.

```

8190     \else%
8191         \gdef\@ledinnote@command{%
8192             ledinnote[\edindexlab\thelabidx]{\@index@command}{\@currentseries}{\@currentnotetype}%
8193         }%
8194     \fi%
8195 }
8196 %

```

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 |.

```

8197 \def\get@index@command#1|#2+{%
8198     \gdef\@index@txt{#1}%
8199     \gdef\@index@command{#2}%
8200     \xdef\@index@parenthesis{}%
8201     \IfBeginWith{\@index@command}{(}{%
8202         \StrGobbleLeft{\@index@command}{1}{\@index@command@}%
8203         \global\let\@index@command\@index@command@%
8204         \xdef\@index@parenthesis{(%}%
8205     }{%
8206         \IfBeginWith{\@index@command}{)}{%
8207             \StrGobbleLeft{\@index@command}{1}{\@index@command@}%
8208             \global\let\@index@command\@index@command@%
8209             \xdef\@index@parenthesis{)%}%
8210         }{%
8211         }%
8212     }%

```

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.

```

8213 \newcommandx{\ledinnote}[5][1,usedefault]{%
8214   \IfStrEq{#4}{critical}{%
8215     \def\tmp{Xinnotemark}%
8216   }{%
8217     \def\tmp{innotemarkX}%
8218   }%
8219   \ifboolexpr{%
8220     test{\ifdefequal{\iftrue}{\ifHy@hyperindex}}%
8221     or%
8222     bool {xindyhyperref@}%
8223   }%
8224   {%
8225     \csuse{#2}{\hyperlink{#1}{\csuse{\tmp @#3}{#5}}}%
8226   }%
8227   {%
8228     \csuse{#2}{\csuse{\tmp @#3}{#5}}%
8229   }%
8230 }%
8231 \newcommand{\ledinnotemark}[1]{#1\emph{n}}%
8232 %

```

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
8233 \newcommandx{\@wredindex}[2][1=\expandonce\jobname,usedefault]{%#1 = the
index name, #2 = the text
8234   \begingroup%
8235   \let\emph\@firstofone%
8236   \let\textbf\@firstofone%
8237   \let\textit\@firstofone%
8238   \let\textmd\@firstofone%
8239   \let\textnormal\@firstofone%
8240   \let\textrm\@firstofone%
8241   \let\textsc\@firstofone%
8242   \let\textsf\@firstofone%
8243   \let\textsl\@firstofone%
8244   \let\texttt\@firstofone%

```



```

8245 \let\textup\@firstofone%
8246 \xdef\@tmp{#2}%To be used in IfSubStr instead of #2 directly. Avoid
some expansion bugs (for example with \edindex{textsc{something}})
8247 \endgroup%
8248 \ifl@imakeidx%
8249 \ifnum\@edindex@fornote@>\z@%
8250 \IfSubStr[1]{\@tmp}{|}{\get@index@command#2}{\get@index@command#2|+}
%
8251 \get@edindex@ledinnote@command%
8252 \expandafter\imki@wrindexentry{#1}{\@index@txt|(\@ledinnote@command
}{\thestartpageline}%
8253 \expandafter\imki@wrindexentry{#1}{\@index@txt|)\@ledinnote@command
}{\theendpageline}%
8254 \else%
8255 \get@edindex@hyperref{#2}%
8256 \imki@wrindexentry{#1}{\@index@txt\@edindex@hyperref}{\thepageline}%
8257 \fi%
8258 \else%
8259 \ifnum\@edindex@fornote@>\z@%
8260 \IfSubStr[1]{\@tmp}{|}{\get@index@command#2}{\get@index@command#2|+}
%
8261 \get@edindex@ledinnote@command%
8262 \expandafter\protected@write\@indexfile{}%
8263 {\string\indexentry{\@index@txt|(\@ledinnote@command}{\thestartpageline}
}%
8264 }%
8265 \expandafter\protected@write\@indexfile{}%
8266 {\string\indexentry{\@index@txt|)\@ledinnote@command}{\theendpageline}
}%
8267 }%
8268 \else%
8269 \protected@write\@indexfile{}%
8270 {\string\indexentry{#2}{\thepageline}
}%
8271 }%
8272 \fi%
8273 \fi%
8274 \endgroup
8275 \@esphack%
8276 }
8277 %

```

Need to add the definition of `\edindex` to `\makeindex`, and initialise `\edindex` to do nothing.

```

8278 \pretocmd{\makeindex}{%
8279 \def\edindex{%
8280 \ifbool{bool{numbering} or bool{numberingR} or bool{
1@dprintingpages} or bool{1@dprintingcolumns}}{%
8281 \@bsphack%
8282 \doedindexlabel%
8283 \begingroup%
8284 \@sanitize%
8285 \@wredindex%

```

```

8286 }%
8287 {%
8288   \led@warn@edinde@outsidenumbering%
8289   \index%
8290 }%
8291 }%
8292 }%
8293 {}%
8294 {\led@error@fail@patch@makeindex}%
8295 \newcommand{\edindex}[1]{\@bsphack\@esphack}
8296 \newcommand{\dummy@edindex}[2][1=\expandonce\jobname,usedefault]{}%
8297 \newcommand{\disable@edindex}{\let\old@edindex\edindex\let\edindex\
  dummy@edindex}%
8298 \newcommand{\restore@edindex}{\let\edindex\old@edindex}%
8299 %

```

XXVIII.7 Hyperlink

\hyperlinkformat `\hyperlinkformat` command is to be used to have both a internal hyperlink and a format, when indexing.

```

8300 \newcommand{\hyperlinkformat}[3]{%
8301   \ifstrempy{#1}%
8302   {\hyperlink{#2}{#3}}%
8303   {\csuse{#1}{\hyperlink{#2}{#3}}}%
8304 }%
8305 %

```

\hyperlinkR `\hyperlinkR` command is to be used to create a internal hyperlink and `\ledRflag`, when indexing.

```

8306 \newcommand{\hyperlinkR}[2]{%
8307   \hyperlink{#1}{#2\@Rlineflag}%
8308 }%
8309 %
8310 %

```

\hyperlinkformatR `\hyperlinkformatR` command is to be used to create a internal hyperlink, a format and a `\@Rlineflag`, when indexing.

```

8311 \newcommand{\hyperlinkformatR}[3]{%
8312   \hyperlinkformat{#1}{#2}{#3\@Rlineflag}%
8313 }%
8314 %
8315 %

```

\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`).

```

8316 \newcommand{\get@edindex@hyperref}[1]{%
8317 %

```

We have to disable temporary spaces to work through a xstring bug (or feature?)

```

8318 \edef\temp@{%
8319 \catcode`\ =9 %space need for catcode
8320 \detokenize{#1}%For active character in unicode
8321 \catcode`\ =10 % space need for catcode
8322 }%
8323 %

```

Now, we define \@edindex@hyperref if the hyperindex of hyperref is enabled.

```

8324 \ifdefequal{\iftrue}{\ifHy@hyperindex}{%
8325 \IfSubStr{\temp@}{|}%
8326 {\get@index@command#1+%
8327 \ifledRcol%
8328 \gdef\@edindex@hyperref{|\@index@parenthesis %space kept
8329 hyperlinkformatR{\@index@command}%
8330 {\edindexlab\thelabidx}}%
8331 \else%
8332 \gdef\@edindex@hyperref{|\@index@parenthesis %space kept
8333 hyperlinkformat{\@index@command}%
8334 {\edindexlab\thelabidx}}%
8335 \fi%
8336 }%
8337 {\get@index@command#1|+%
8338 \ifledRcol%
8339 \gdef\@edindex@hyperref{hyperlinkR{\edindexlab\thelabidx}}%
8340 \else%
8341 \gdef\@edindex@hyperref{hyperlink{\edindexlab\thelabidx}}%
8342 \fi%
8343 }%
8344 }%
8345 %

```

```

8346 % If we use both xindy and hyperref, first get the \protect\cs{
index@command} command.
8347 % Then define \protect\cs{@edindex@hyperref} in the form \verb+eledmacXXX+
8348 % \begin{macrocode}
8349 {\ifxindyhyperref%
8350 \IfSubStr{\temp@}{|}%
8351 {\get@index@command#1+%
8352 {\get@index@command#1|+%
8353 \gdef\@edindex@hyperref{eledmac\thelabidx}%
8354 %

```

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.

```

8355 \IfStrEq{\@index@parenthesis}{(}%

```

```

8356      {%
8357      \csxdef{xindyparenthesis@\@index@txt}{\thelabidx}%
8358      \gdef\@edindex@hyperref{|\eledmac\thelabidx}%
8359      }%
8360      {}%
8361  %

```

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`.

```

8362      \IfStrEq{\@index@parenthesis{}}{%
8363      {%
8364      \xdef\@edindex@hyperref{|\eledmac\csuse{xindyparenthesis@\@index@txt}}%
8365      \global\csundef{xindyparenthesis@\@index@txt}%
8366      }%
8367  %

```

Write in the `.xdy` file the attributes of the location.

```

8368      {%
8369      \immediate\write\eledmac@xindy@out{%
8370      (define-attributes ("eledmac\thelabidx"))^^J
8371      \space\space(markup-locref^^J
8372      \eledmacmarkuplocdepth^^J
8373      :open "\string\hyperlink%
8374      \ifledRcol R\fi%
8375      {\edindexlab\thelabidx}%
8376      {\ifdefempty{\@index@command}%
8377      {}%
8378      {\@backslashchar\@index@command}%
8379      {"^^J
8380      :close "}"^^J
8381      :attr "eledmac\thelabidx"^^J
8382      )
8383      }%
8384      }%
8385  %

```

And now, in any other case.

```

8386      \else%
8387      \gdef\@index@txt{#1}%
8388      \gdef\@edindex@hyperref{}%
8389      \fi%
8390      }%
8391  }
8392  %

```

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.

```

8393 \newcommand{\led@set@index@fornote}[1]{%
8394   \ifbool{indtl@innote}%
8395     {\let\index\nindex}%
8396     {}%
8397   \ifbool{indtl@notenumber}%
8398     {%
8399     \renewcommand{\index}[2][\indtl@jobname]{%
8400       \orig@@index[##1]{%
8401         ##2|innotenumber{\this@footnoteX@reading}%
8402       }%
8403     }%
8404     }%
8405     {}%
8406   }%
8407   %

```

`\led@reinit@index@fornote` The `\led@reinit@index@fornote` just reset the default value of `\index`.

```

8408 \newcommand{\led@reinit@index@fornote}{%
8409   \ifbool{indtl@innote}%
8410     {\let\index\orig@@index}%
8411     {}%
8412   \ifbool{indtl@notenumber}%
8413     {\let\index\orig@@index}%
8414     {}%
8415   }%
8416   %

```

XXIX Glossaries

Here, we define the `\gls`-like commands prefixed by `ed`, only if the package *glossaries* is loaded.

```

8417 \AtBeginDocument{%
8418   \ifpackageloaded{glossaries}{%
8419     %

```

First those which arguments are `[<options>]{<label>}[<insert>]`.

```

8420   \gdef\edglsomo{}%
8421   \listgadd{\edglsomo}{gls}%
8422   \listgadd{\edglsomo}{Gls}%
8423   \listgadd{\edglsomo}{GLS}%

```

```

8424 \listgadd{\edglsomo}{glspl}%
8425 \listgadd{\edglsomo}{Glspl}%
8426 \listgadd{\edglsomo}{GLSpl}%
8427 \listgadd{\edglsomo}{glstext}%
8428 \listgadd{\edglsomo}{Glstext}%
8429 \listgadd{\edglsomo}{GLStext}%
8430 \listgadd{\edglsomo}{Glsfirst}%
8431 \listgadd{\edglsomo}{GLSfirst}%
8432 \listgadd{\edglsomo}{glsplural}%
8433 \listgadd{\edglsomo}{Glsplural}%
8434 \listgadd{\edglsomo}{GLSplural}%
8435 \listgadd{\edglsomo}{glsfirstplural}%
8436 \listgadd{\edglsomo}{Glsfirstplural}%
8437 \listgadd{\edglsomo}{GLSfirstplural}%
8438 \listgadd{\edglsomo}{glsname}%
8439 \listgadd{\edglsomo}{Glsname}%
8440 \listgadd{\edglsomo}{GLSname}%
8441 \listgadd{\edglsomo}{glssymbol}%
8442 \listgadd{\edglsomo}{Glsymbol}%
8443 \listgadd{\edglsomo}{GLSsymbol}%
8444 \listgadd{\edglsomo}{glsdesc}%
8445 \listgadd{\edglsomo}{Glsdesc}%
8446 \listgadd{\edglsomo}{GLSdesc}%
8447 \listgadd{\edglsomo}{glsuseri}%
8448 \listgadd{\edglsomo}{Glsuseri}%
8449 \listgadd{\edglsomo}{GLSuseri}%
8450 \listgadd{\edglsomo}{glsuserii}%
8451 \listgadd{\edglsomo}{Glsuserii}%
8452 \listgadd{\edglsomo}{GLSuserii}%
8453 \listgadd{\edglsomo}{glsuseriii}%
8454 \listgadd{\edglsomo}{Glsuseriii}%
8455 \listgadd{\edglsomo}{GLSuseriii}%
8456 \listgadd{\edglsomo}{glsuseriv}%
8457 \listgadd{\edglsomo}{Glsuseriv}%
8458 \listgadd{\edglsomo}{GLSuseriv}%
8459 \listgadd{\edglsomo}{glsuserv}%
8460 \listgadd{\edglsomo}{Glsuserv}%
8461 \listgadd{\edglsomo}{GLSuserv}%
8462 \listgadd{\edglsomo}{glsuservi}%
8463 \listgadd{\edglsomo}{Glsuservi}%
8464 \listgadd{\edglsomo}{GLSuservi}%
8465 \renewcommand{\do}[1]{%
8466 \expandafter\DeclareRobustCommandx\csname ed#1\endcsname[3][1,3,
usedefault]{%
8467 \doedindexlabel%
8468 \ifdef{\hypertarget}{%
8469 \Hy@raisedlink@left{\hypertarget{pageline.\thepageline}{}}%
8470 }{}%
8471 \csname#1\endcsname[counter=pageline,##1]{##2}[##3]%
8472 }%

```

```

8473 \expandafter\WithSuffix\expandafter\DeclareRobustCommand\csname ed
#1\endcsname*[3][1,3,usedefault]{%
8474 \doedindexlabel%
8475 \ifdef{\hypertarget}{%
8476 \Hy@raisedlink@left{\hypertarget{pageline.\thepageline}}}%
8477 }{}%
8478 \csname#1\endcsname*[counter=pageline,##1]{##2}{##3}%
8479 }%
8480 }%
8481 \dolistloop{\edglsomo}%
8482 %

```

Then those which arguments are [*options*]{*label*}{*link text*}.

```

8483 \gdef\edglsomm{%
8484 \listgadd\edglsomm{glsdisp}%
8485 \listgadd\edglsomm{glslink}%
8486 \renewcommand{\do}[1]{%
8487 \expandafter\DeclareRobustCommand\csname ed#1\endcsname[3][1,
usedefault]{%
8488 \doedindexlabel%
8489 \ifdef{\hypertarget}{%
8490 \Hy@raisedlink@left{\hypertarget{pageline.\thepageline}}}%
8491 }{}%
8492 \csname#1\endcsname[counter=pageline,##1]{##2}{##3}%
8493 }%
8494 \expandafter\WithSuffix\expandafter\DeclareRobustCommand\csname ed
#1\endcsname*[3][1,usedefault]{%
8495 \doedindexlabel%
8496 \ifdef{\hypertarget}{%
8497 \Hy@raisedlink@left{\hypertarget{pageline.\thepageline}}}%
8498 }{}%
8499 \csname#1\endcsname*[counter=pageline,##1]{##2}{##3}%
8500 }%
8501 }%
8502 \dolistloop{\edglsomm}%
8503 %

```

Then those which arguments are [*options*]{*label*}.

```

8504 \gdef\edglsom{%
8505 \listgadd\edglsom{glsadd}%
8506 \renewcommand{\do}[1]{%
8507 \expandafter\DeclareRobustCommand\csname ed#1\endcsname[2][1,
usedefault]{%
8508 \doedindexlabel%
8509 \ifdef{\hypertarget}{%
8510 \Hy@raisedlink@left{\hypertarget{pageline.\thepageline}}}%
8511 }{}%
8512 \csname#1\endcsname[counter=pageline,##1]{##2}%
8513 }%

```

```

8514 \expandafter\WithSuffix\expandafter\DeclareRobustCommand\csname ed
#1\endcsname*[2][1,usedefault]{%
8515 \doedindexlabel%
8516 \ifdef{\hypertarget}{%
8517 \Hy@raisedlink@left{\hypertarget{pageline.\thepageline}{}}%
8518 }{}%
8519 \csname#1\endcsname*[counter=pageline,##1]{##2}%
8520 }%
8521 }%
8522 \dolistloop{\edglsom}%
8523 %

```

`\l@noexpands@edgl` is a macro to avoid expanding glossaries macros in `\edtext` first argument. It is called by `\no@expands`. Its definition is made dynamically.

```

\l@noexpands@edgl
8524 \def\l@noexpands@edgl{%
8525 \renewcommand\do[1]{%
8526 \xappto\l@noexpands@edgl{%
8527 \noexpand\RenewDocumentCommand\csname ed#1\endcsname{omos}{}%
8528 }%
8529 }%
8530 \dolistloop{\edglsomo}%
8531 \renewcommand\do[1]{%
8532 \xappto\l@noexpands@edgl{%
8533 \noexpand\RenewDocumentCommand\csname ed#1\endcsname{omms}{}%
8534 }%
8535 }%
8536 \dolistloop{\edglsomm}%
8537 \renewcommand\do[1]{%
8538 \xappto\l@noexpands@edgl{%
8539 \noexpand\RenewDocumentCommand\csname ed#1\endcsname{oms}{}%
8540 }%
8541 }%
8542 \dolistloop{\edglsom}%
8543 }{\let\l@noexpands@edgl\relax}%
8544 }%
8545 %

```

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.

```
8546 \def\@hangingsymbol{}
8547 \newcommand*\sethangingsymbol[1]{%
8548   \gdef\@hangingsymbol{#1}%
8549 }%
8550 \newif\ifinstanza
8551 %
```

`\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.

```
8552 \newif\ifinserthangingsymbol
8553 \newcommand*\inserthangingsymbol{%
8554   \ifinserthangingsymbol%
8555     \ifinstanza%
8556       \@hangingsymbol%
8557     \fi%
8558   \fi%
8559 }
8560 %
```

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.

```
8561 \newcommand*\ampersand{\char`\&}
8562
8563 %
```

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.

```
8564 \chardef\body=\catcode`\@
8565 \catcode`\@=11
8566 \chardef\next=\catcode`\&
8567 \catcode`\&=\active
8568
8569 %
```

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.

```
8570 \newcount\stanza@count
8571 \newlength{\stanzaindentbase}
8572 \setlength{\stanzaindentbase}{20pt}
8573
8574 %
```

`\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.

```
8575 \def\strip@szacnt#1,#2|{\def\@tempb{#1}\def\@tempa{#2|}}
8576 \newcommand*\setstanzavalues[2]{\def\@tempa{#2,|}%
8577   \stanza@count\z@
8578   \def\next{\expandafter\strip@szacnt\@tempa
8579     \ifx\@tempb\empty\let\next\relax\else
8580       \expandafter\mathchardef\csname #1@\number\stanza@count
8581       \@endcsname\@tempb\relax
8582       \advance\stanza@count\@ne\fi\next}%
8583   \next}
8584
8585 %
```

`\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).

```
8586 \newcommand*\setstanzaindents[1]{\setstanzavalues{sza}{#1}}
8587 \newcommand*\setstanzapenalties[1]{\setstanzavalues{szp}{#1}}
8588 %
8589 %
```

`\managestanza@modulo` Since the `stanzaindentrepetition` counter can be used when the indentation is repeated every n verses.

The `startstanzaindentrepetition` 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 `stanzaindentrepetition` counter, the command restarts it.

```

8590 \newcounter{stanzaindentrepetition}
8591 \newcounter{startstanzaindentrepetition}%
8592 \setcounter{startstanzaindentrepetition}{1}%
8593 \newcount\stanza@modulo
8594
8595 \newcommand*{\managestanza@modulo}[0]{%
8596   \advance\stanza@modulo\@ne%
8597   \ifnum\stanza@modulo>\value{stanzaindentrepetition}%
8598     \stanza@modulo\@ne%
8599   \fi%
8600 }
8601 %

```

\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.

```

8602 \newcommand{\stanzaindent}[1]{%
8603   \hspace{\dimexpr#1\stanzaindentbase-\parindent\relax}%
8604   \ignorespaces%
8605 }%
8606 \WithSuffix\newcommand\stanzaindent*[1]{%
8607   \stanzaindent{#1}%
8608   \global\advance\stanza@modulo-\@ne%
8609   \ifnum\stanza@modulo=0%
8610     \global\stanza@modulo=\value{stanzaindentrepetition}%
8611   \fi%
8612   \ignorespaces%
8613 }%
8614 %

```

XXX.5 Numbering stanza

Here, macro for numbering stanza. First, the stanza counter.

```

\thestanza 8615 \newcounter{stanza}
8616 \renewcommand{\thestanza}{%
8617   \textbf{\arabic{stanza}}%
8618 }
8619 %

```

\ifnumberstanza Then, macro to activate automatically numbering of stanza.

```

8620 \newif\ifnumberstanza%
8621 %

```

\@insertstanzanumber Now, macro called at the first line of of verse of a stanza.

```

8622 \newcommand{\@insertstanzanumber}[0]{%
8623   \ifnumberstanza%
8624     \ifl@dpairing%
8625       \ifledRcol%
8626         \stanzanumwrapper{\thestanzaR}%
8627       \else%
8628         \stanzanumwrapper{\thestanzaL}%
8629       \fi%
8630     \else%
8631       \stanzanumwrapper{\thestanza}%
8632     \fi%
8633     \setline{1}%
8634   \fi%
8635 }%
8636 %

```

\@advancestanzanumber Also a command to advance the counter of stanza.

```

8637 \newcommand{\@advancestanzanumber}[0]{%
8638   \ifnumberstanza%
8639     \ifl@dpairing%
8640       \ifledRcol%
8641         \addtocounter{stanzaR}{1}%
8642       \else%
8643         \addtocounter{stanzaL}{1}%
8644       \fi%
8645     \else%
8646       \addtocounter{stanza}{1}%
8647     \fi%
8648   \fi%
8649 }%
8650 %

```

\stanzanumwrapper And finally, the wrapper for stanza number

```

8651 \newcommand{\stanzanumwrapper}[1]{%
8652   \flagstanza{#1}%
8653 }%
8654 %

```

XXX.6 Stanza number in note

Here, the command called when printing stanza number in notes.

```

8655 \newcommand{\printstanza}[0]{%
8656   \ifboolexpr{bool{l@dpairing} or bool{l@dprintingpages} or bool{
8657     l@dprintingcolumns}}{%
8658     \ifledRcol%
8659       \thestanzaR%

```

```

8659     \else%
8660         \thestanzaL%
8661     \fi%
8662 }{%
8663     \thestanza%
8664 }%
8665 }
8666 %

```

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`

```

8667 \newcommandx{\stanza@line}[2][1,2,usedefault]{%
8668     \stanza@indent%
8669     \pstart[#1][#2]\stanza@hang\ignorespaces%
8670 }%
8671 \newcommand{\stanza@indent}{%
8672     \ifboolexpr{%
8673         test {\ifnumequal{\value{stanzaindentrepetition}}{0}}%
8674         or%
8675         test {\ifnumless{\number\stanza@count}{\value{
8676             startstanzaindentrepetition}}}%
8677     }{%
8678         \ifcsdef{sza@number\stanza@count @}{%
8679             {%
8680                 \parindent=\csname sza@number\stanza@count @\endcsname\
8681                 stanzaindentbase%
8682             }%
8683         }%
8684         \ifcsdef{sza@number\stanza@modulo @}{%
8685             \parindent=\csname sza@%
8686             \the\numexpr%
8687             \stanza@modulo%
8688             - 1%
8689             + \value{startstanzaindentrepetition}%
8690             \relax%
8691             @\endcsname\stanzaindentbase%
8692             \managestanza@modulo%
8693         }%

```

```

8694     {%
8695     \led@err@StanzaIndentNotDefined%
8696     }%
8697 }%
8698 }%
8699 \xdef\stanza@hang{\noexpand\leavevmode\noexpand\startlock
8700     \hangindent\expandafter
8701     \noexpand\csname sza@0@\endcsname\stanzaindentbase
8702     \hangafter\@ne}
8703 \def\sza@penalty{\count@\csname szp@\number\stanza@count @\endcsname
8704     \ifnum\count@>@M\advance\count@-\@M\penalty-\else
8705     \penalty\fi\count@}
8706 %

```

\@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 \&.

```

8707 \xdef\@startstanza[#1][#2]{%
8708     \noexpand\instanzatrue\expandafter
8709     \begingroup%
8710     \catcode`\noexpand\&\active%
8711     \unexpanded{\newdimen\parindent@beforestanza%
8712     \parindent@beforestanza=\parindent}%Keep in memory the standard \
8713     parindent
8714     \global\stanza@count\@ne\stanza@modulo\@ne
8715     \noexpand\ifnum\expandafter\noexpand\csname sza@0@\endcsname=z@%
8716     \def\noexpand\stanza@hang{\unexpanded{\leavevmode\null}}%
8717     \let\noexpand\endlock\relax%
8718     \noexpand\else%
8719     \interlinepenalty%
8720     \@M\rightskip\z@ plus 1fil\relax%
8721     \noexpand\fi%
8722     \noexpand\ifnum\expandafter\noexpand\csname szp@0@\endcsname=z@%
8723     \let\noexpand\sza@penalty%
8724     \relax\noexpand\fi%
8725     \def\noexpand&{%
8726         \noexpand\newverse[] []}%
8727     \def\noexpand\&{\noexpand\@stopstanza}%
8728     \noexpand\@advancestanzanumber%
8729     \noexpand\stanza@line[#1][#2]%
8730     \noexpand\@insertstanzanumber%
8731     \let\par\relax\ignorespaces%No paragraph in verses
8732 }

```

```

8732 \newcommandx{\stanza}[2][1,2,usedefault]{%
8733   \ifboolexpr{%
8734     not test{\ifdefvoid{\at@every@stanza}}%
8735     and test{\ifstrempy{#1}}%
8736     and test{\ifstrempy{#2}}%
8737     {\@startstanza[][\at@every@stanza]\at@start@every@stanza}%
8738     {\@startstanza[#1][#2]\at@start@every@stanza}%
8739   }%
8740 }%
8741
8742 \newcommandx{\@stopstanza}[2][1,2,usedefault]{%
8743   \unskip%
8744   \endlock%
8745   \ifboolexpr{%
8746     not test{\ifdefvoid{\at@every@stop@stanza}}%
8747     and test{\ifstrempy{#1}}%
8748     and test{\ifstrempy{#2}}%
8749     {\before@every@stop@stanza\pend[][\at@every@stop@stanza]}%
8750     {\before@every@stop@stanza\pend[#1][#2]}%
8751   }%
8752   \instanzafalse%
8753   \global\undef\parindent@beforestanza%
8754 }
8755
8756 \newcommand{\AtEveryStopStanza}[1]{%
8757   \ifstrempy{#1}%
8758     {\gdef\at@every@stop@stanza{}}%
8759     {\gdef\at@every@stop@stanza{\noindent#1}}%
8760 }%
8761 \WithSuffix\newcommand\AtEveryStopStanza*[1]{%
8762   \ifstrempy{#1}%
8763     {\gdef\at@every@stop@stanza{}}%
8764     {\gdef\at@every@stop@stanza{#1}}%
8765 }%
8766 \def\at@every@stop@stanza{}%
8767
8768 \newcommand{\AtEveryStanza}[1]{%
8769   \ifstrempy{#1}%
8770     {\gdef\at@every@stanza{}}%
8771     {\gdef\at@every@stanza{\noindent#1}}%
8772 }%
8773 \WithSuffix\newcommand\AtEveryStanza*[1]{%
8774   \ifstrempy{#1}%
8775     {\gdef\at@every@stanza{}}%
8776     {\gdef\at@every@stanza{#1}}%
8777 }%
8778
8779
8780
8781 \newcommand{\AtStartEveryStanza}[1]{%

```

```

8782 \ifstrempy{#1}%
8783   {\gdef\at@start@every@stanza{}}}%
8784   {\gdef\at@start@every@stanza{#1}}}%
8785 }%
8786 \def\at@start@every@stanza{}%
8787
8788 \newcommand{\BeforeEveryStopStanza}[1]{%
8789   \ifstrempy{#1}%
8790     {\gdef\before@every@stop@stanza{}}}%
8791     {\gdef\before@every@stop@stanza{#1}}}%
8792 }%
8793 \def\before@every@stop@stanza{}%
8794
8795 \newcommand*{\newverse}[4][1,2,3,4,usedefault]{%
8796   \unskip%
8797   \endlock\pend[#1][#3]\sza@penalty\global%
8798   \advance\stanza@count\@ne\stanza@line[#2][#4]%
8799   }
8800
8801 %

```

\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`.

```

8802 \newcommand*{\flagstanza}[2][\stanzaindentbase]{%
8803   \hskip -#1\llap{#2}\hskip #1\ignorespaces}
8804
8805 %

```

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.

```

8806 \catcode`\&=\next
8807 \catcode`\@=\body
8808 \setstanzavalues{szp}{0}
8809
8810 %

```

XXX.9 Dealing with antilabe

\firsthalf@antilabe@wd If the `antilabe` option is enabled, after having vboxed the content of `\pstart...pend`
\get@firsthalf@antilabe@wd structure, we get the width of the last line of the `\pend`.
\antilabe


```

8811 \ifantilabe@%
8812 \newdimen\firsthalf@antilabe@wd%
8813 \newcommand\get@firsthalf@antilabe@wd{%
8814 \ifboolexpr{bool {istanza} or bool {inastanzaL} or bool {inastanzaR}}{
%
8815 \vbadness=10000%
8816 \splittopskip=\z@%
8817 \ifl@dpairing%
8818 \ifledRcol%
8819 \setbox0=\copy\csname l@dRcolrawbox\the\l@dnumstartsR\endcsname%
8820 \else%
8821 \setbox0=\copy\csname l@dLcolrawbox\the\l@dnumstartsL\endcsname%
8822 \fi%
8823 \else%
8824 \setbox0=\copy\raw@text%
8825 \fi%
8826 \dimen0=\dimexpr\ht0+\dp0-\baselineskip\relax%
8827 \ifdim\dimen0>0pt%
8828 \setbox1=\vsplit0 to \dimen0%
8829 \fi%
8830 \setbox0=\vbox{\unvbox0\global\setbox1=\lastbox}%
8831 \setbox1=\hbox{\noindent\unhbox1}%
8832 \global\firsthalf@antilabe@wd=\wd1%
8833 }{}%
8834 }%
8835 \def\beforeantilabe{}%
8836 \def\afterantilabe{}%
8837 \newcommand{\antilabe}[1]{%
8838 \beforeantilabe\hbox to\firsthalf@antilabe@wd{#1\hfill}\afterantilabe%
8839 }%
8840 \else%
8841 \let\get@firsthalf@antilabe@wd\relax%
8842 \fi%
8843 %

```

XXXI Apparatus of Manuscripts

XXXI.1 User level macro

\msdata The user level `\msdata` command only writes the manuscripts data in a numbered auxiliary file. There are two associated etoolbox counters.

\msdata@c

```

8844 \def\msdata@c{}%
8845 \def\msdata@cR{}%
8846 \newcommand{\msdata}[1]{%
8847 \leavevmode%
8848 \unless\ifstopmsdata@inserted@%
8849 \stopmsdata%
8850 \led@warning@msdatawithoutstop%

```

```

8851 \fi%
8852 \global\stopmsdata@inserted@false%
8853 \unless\ifledRcol%
8854   \numgdef{\msdata@c}{\msdata@c+1}%
8855   \ifdef{\hypertarget}{%
8856     \edlabel{\msdata@c:start:msdata}%
8857   }{}%
8858   \protected@write\linenum@out{}{%
8859     \string\@msd{#1}%
8860   }%
8861 \else%
8862   \numgdef{\msdata@cR}{\msdata@cR+1}%
8863   \ifdef{\hypertarget}{%
8864     \edlabel{\msdata@cR:start:msdata}%
8865   }{}%
8866   \protected@write\linenum@outR{}{%
8867     \string\@msd{#1}%
8868   }%
8869 \fi%
8870 }%
8871 %

```

\stopmsdata The user level `\stopmsdata` command only writes information about the end of manuscripts data in numbered auxiliary file.

```

8872 \newcommand{\stopmsdata}[0]{%
8873   \leavevmode%
8874   \unless\ifledRcol%
8875     \protected@write\linenum@out{}{%
8876       \string\@stopmsd%
8877     }%
8878     \ifdef{\hypertarget}{%
8879       \edlabel{\msdata@c:end:msdata}%
8880     }{}%
8881   \else%
8882     \protected@write\linenum@outR{}{%
8883       \string\@stopmsd%
8884     }%
8885     \ifdef{\hypertarget}{%
8886       \edlabel{\msdata@cR:end:msdata}%
8887     }{}%
8888   \fi%
8889   \global\stopmsdata@inserted@true%
8890 }%
8891 %

```

\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`

```

8892 \newif\ifstopmsdata@inserted@%
8893 %

```

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?

```

8894 \newcommand{\setmsdataseries}[1]{%
8895   \gdef\@msdata@series{#1}%
8896 }%
8897 \def\@msdata@series{A}%
8898 %

```

\setmsdataposition The label for the manuscripts data.

```

8899 \def\ms@data@position{msdata-regular}%
8900 \newcommand{\setmsdataposition}[1]{%
8901   \gdef\ms@data@position{#1}%
8902 }%
8903 %

```

\setmsdatalabel The label for the manuscripts data.

```

8904 \def\ms@data@label{Ms.}%
8905 \newcommand{\setmsdatalabel}[1]{%
8906   \gdef\ms@data@label{#1}%
8907 }%
8908 %

```

XXXI.3 Counters and lists

\@msd@c \@msd@c is a counter incremented at each \@msd read in auxiliary file.

```

8909 \numdef{\@msd@c}{0}
8910 \numdef{\@msd@cR}{0}
8911 %

```

\add@msd@ \add@msd@ is a counter incremented at each \add@msddata, that is at each time we prepare the insertion of manuscripts data footnote.

```

8912 \numdef{\add@msd@c}{0}%
8913 \numdef{\add@msd@cR}{0}%
8914 %

```

\@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.

```

8915 \def\@msdata@list{%
8916 %

```

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.

```

8917 \newcommand{\@msd}[1]{%
8918   \unless\ifledRcol%
8919     \global\newdef{\@msd@c}{\@msd@c+\@one}%
8920     \csgdef{\@msdata@\@msd@c @data}{#1}%
8921     \csxdef{\@msdata@\@msd@c @linenumber}{\the\line@num}%
8922     \csxdef{\@msdata@\@msd@c @sublinenumber}{\the\subline@num}%
8923     \csxdef{\@msdata@\@msd@c @abslinenumber}{\the\absline@num}%
8924     \xright@appenditem{\the\absline@num}\to\actionlines@list%
8925     \xright@appenditem{-1010}\to\actions@list%
8926   \else%
8927     \global\newdef{\@msd@cR}{\@msd@cR+\@one}%
8928     \csgdef{\@msdata@\@msd@cR @dataR}{#1}%
8929     \csxdef{\@msdata@\@msd@cR @linenumberR}{\the\line@numR}%
8930     \csxdef{\@msdata@\@msd@cR @sublinenumberR}{\the\subline@numR}%
8931     \csxdef{\@msdata@\@msd@cR @abslinenumberR}{\the\absline@numR}%
8932     \xright@appenditem{\the\absline@numR}\to\actionlines@listR%
8933     \xright@appenditem{-1010}\to\actions@listR%
8934   \fi%
8935 }%
8936 %

```

\@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.

```

8937 \newcommand{\@stopmsd}[0]{%
8938   \unless\ifledRcol%
8939     \ifcsundef{\@msdata@\@msd@c @stoplinenumber}{%
8940       \csxdef{\@msdata@\@msd@c @stopabslinenumber}{\the\absline@num}%
8941       \csxdef{\@msdata@\@msd@c @stoplinenumber}{\the\line@num}%
8942       \csxdef{\@msdata@\@msd@c @stopsublinenumber}{\the\subline@num}%
8943     }%
8944   \else%
8945     \ifcsundef{\@msdata@\@msd@cR @stoplinenumberR}{%
8946       \csxdef{\@msdata@\@msd@cR @stopabslinenumberR}{\the\absline@numR}%
8947       \csxdef{\@msdata@\@msd@cR @stoplinenumberR}{\the\line@numR}%
8948       \csxdef{\@msdata@\@msd@cR @stopsublinenumberR}{\the\subline@numR}%
8949     }%
8950   }%
8951 \fi%

```

```
8952 }%
8953 %
```

XXXI.5 Action macro

\add@msdata \add@msdata is executed on each line when action code 1010 is seen. It will not insert immediately the manuscript data footnote, as action code are executed before the line be typeset, and, consequently, could be on the previous page. So it just stores the manuscript data footnote to \@msdata@list.

```
8954 \newcommand{\add@msdata}{%
8955   \bgroup%
8956   \normalfont%
8957   \unless\ifledRcol{%
8958     \numgdef{\add@msd@c}{\add@msd@c+\@ne}%
8959     \ifcsdef{@msdata@\add@msd@c @data}{%
8960       \letcs{@data}{@msdata@\add@msd@c @data}%
8961       \edef\l@d@nums{%
8962         000|}% Start page = we don't print it
8963         \csuse{@msdata@\add@msd@c @linenumber}|}% Start line number
8964         \csuse{@msdata@\add@msd@c @sublinenumber}|}% Start subline number
8965         000|}% End page number, we don't print it
8966         \ifnumless{\csuse{@msdata@\add@msd@c @stopabslinenumber}}{\csuse{
@lastabsline@forpage@\the\page@num}}}%
8967         {\csuse{@msdata@\add@msd@c @stoplinenumber}}}%End line number if
in the same page
8968         {\csuse{@lastline@forpage@\the\page@num}}}%Otherwise, last
number of the page
8969         |%
8970         \ifnumless{\csuse{@msdata@\add@msd@c @stopabslinenumber}}{\csuse{
@lastabsline@forpage@\the\page@num}}}%
8971         {\csuse{@msdata@\add@msd@c @stopsublinenumber}}}%End subline
number if in the same page
8972         {\csuse{@lastsubline@forpage@\the\page@num}}}%Otherwise, last
subnumber of the page
8973         |%
8974         \edfont@info%Font
8975       }%
8976       \@msd@options@fullpagefalse%
8977       \if@firstlineofpage%Try if the data are for the full page. If yes
, will add options to the list.
8978       \unless\if@msdata@insertedfrompreviouspage%
8979       \ifnumless{\csuse{@lastabsline@forpage@\the\page@num}}{\csuse
{@msdata@\add@msd@c @stopabslinenumber}+\@ne}%
8980       {%
8981       \numdef{@tmp}{\add@msd@c+\@ne}%
8982       \ifcsdef{@msdata@\@tmp @abslinenumber}%
8983       {\ifnumequal{\csuse{@msdata@\@tmp @abslinenumber}}{\csuse{
@lastabsline@forpage@\the\page@num}}}%
```

```

8984         {}%
8985         {\@msd@options@fullpagetrue}%
8986     }%
8987     {\@msd@options@fullpagetrue}%
8988 }%
8989 {}%
8990 \fi%
8991 \fi%
8992 \listxadd{\@msdata@list}{%
8993     \@msd@options@iffullpage%
8994     \ifluatex%
8995         \csxdef{footnote@luatextextdir}{\the\textdir}%
8996         \csxdef{footnote@luatexpardir}{\the\pardir}%
8997     \fi%
8998     \csdef{@this@crossref@start}{\add@msd@c:start:msdata}%
8999     \csdef{@this@crossref@end}{\add@msd@c:end:msdata}%
9000     \unexpanded{%
9001         \def\annot@start{}%
9002         \def\annot@end{}%
9003     }%
9004     \noexpand\csuse{v\@msdata@series footnote}{\@msdata@series}{\{
expandonce\l@d@nums}{\ms@data@label}{\expandonce\data}}%
9005     \reset@msd@options@iffullpage%
9006 }%
9007 }%
9008 {}%
9009 \else%
9010     \numgdef{\add@msd@cR}{\add@msd@cR+\@one}%
9011     \ifcsdef{@msdata@\add@msd@cR @dataR}{%
9012         \letcs{\@data}{\@msdata@\add@msd@cR @dataR}%
9013         \edef\l@d@nums{%
9014             000| % Start page = we don't print it
9015             \csuse{@msdata@\add@msd@cR @linenumberR}| % Start line number
9016             \csuse{@msdata@\add@msd@cR @sublinenumberR}| % Start subline
number, for now, not used
9017             000| % End page number, we don't print it
9018             \ifnumless{\csuse{@msdata@\add@msd@cR @stopabslinenumberR}}{\
csuse{@lastline@forpageR@the\page@numR}}{%
9019                 {\csuse{@msdata@\add@msd@cR @stoplinenumberR}}}%End line number
if in the same page
9020                 {\csuse{@lastline@forpageR@the\page@numR}}}%Otherwise, last
number of the page
9021             |%
9022             \ifnumless{\csuse{@msdata@\add@msd@cR @stopabslinenumberR}}{\
csuse{@lastline@forpageR@the\page@numR}}{%
9023                 {\csuse{@msdata@\add@msd@cR @stopsublinenumberR}}}%End subline
number if in the same page
9024                 {\csuse{@lastsubline@forpageR@the\page@numR}}}%Otherwise, last
number of the page
9025             000| % End sub line number, for now, not used

```

```

9026     \edfont@info%Font
9027   }%
9028   \@msd@options@fullpagefalse%
9029   \if@firstlineofpageR%
9030     \unless\if@msdata@insertedfrompreviouspage%
9031       \ifnumless{\csuse{@lastabsline@forpageR@the\page@numR}}{\
csuse{@msdata@add@msd@c @stopabslinenumberR}+\@ne}%
9032       {%
9033         \numdef{\@tmp}{\add@msd@cR+\@ne}%
9034         \ifcsdef{@msdata@\@tmp @abslinenumberR}%
9035           {\ifnumequal{\csuse{@msdata@\@tmp @abslinenumberR}}{\csuse{
@lastabsline@forpageR@the\page@numR}}}%
9036           {}%
9037           {\@msd@options@fullpagetrue}%
9038         }%
9039         {\@msd@options@fullpagetrue}%
9040       }%
9041     {}%
9042   \fi%
9043 \fi%
9044 \listxadd{\@msdata@list}{%
9045   \@msd@options@iffullpage%
9046   \ifluatex%
9047     \csxdef{footnote@luatextextdir}{\the\textdir}%
9048     \csxdef{footnote@luatexpardir}{\the\pardir}%
9049   \fi%
9050   \csdef{@this@crossref@start}{\add@msd@cR:start:msdata}%
9051   \csdef{@this@crossref@end}{\add@msd@cR:end:msdata}%
9052   \unexpanded{%
9053     \def\annot@start{}%
9054     \def\annot@end{}%
9055   }%
9056   \noexpand\csuse{v\@msdata@series footnote}{\@msdata@series}{\
expandonce\l@d@nums}{\ms@data@label}{\expandonce\data}}%
9057   \reset@msd@options@iffullpage%
9058 }%
9059 }%
9060 {}%
9061 \fi%
9062 \egroup%
9063 }%
9064 %

```

[\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.

```

9065 \newif\if@msdata@insertedfrompreviouspage%
9066 %

```

`\add@msdata@firstlineofpage` `\add@msdata@firstlineofpage` is called at the first line of every page. It inserts manuscript data which start on one of the previous pages and continue on this page.

```

9067 \newcommand{\add@msdata@firstlineofpage}{%
9068   \bgroup%
9069   \normalfont%
9070   \unless\ifledRcol{%
9071     \ifcsdef{@msdata@\add@msd@c @data}{%
9072       \ifnumless{\the\absline@num-\@ne}{\csuse{@msdata@\add@msd@c
@stopabslinenumber}}}%
9073       {%
9074         \global\@msdata@insertedfrompreviouspagetrue%
9075         \letcs{@data}{@msdata@\add@msd@c @data}%
9076         \edef\l@d@nums{%
9077           000|% Start page = we don't print it
9078           \numexpr\the\line@num+\@ne\relax|% Start line number = first line
of the page. As \add@msdata@firstlineofpage is called before line number
has been incremented, we increment it for printing
9079           000|% Start subline number, for now, not used
9080           000|% End page number, we don't print it
9081           \ifnumless{\csuse{@msdata@\add@msd@c @stopabslinenumber}}{\csuse{
@lastabsline@forpage@\the\page@num}}}%
9082           {\csuse{@msdata@\add@msd@c @stoplinenumber}}}%End line number if
in the same page
9083           {\csuse{@lastline@forpage@\the\page@num}}}%Otherwise, last
number of the page
9084           |%
9085           000|% End sub line number, for now, not used
9086           \edfont@info%Font
9087           }%
9088           \@msd@options@fullpagefalse%
9089           \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
9090           {%
9091             \numdef{\@tmp}{\add@msd@c+\@ne}%
9092             \ifcsdef{@msdata@\@tmp @abslinenumber}%
9093             {\ifnumequal{\csuse{@msdata@\@tmp @abslinenumber}}{\csuse{
@lastabsline@forpage@\the\page@num}}}%
9094             }%
9095             {\@msd@options@fullpagetrue}%
9096             }%
9097             {\@msd@options@fullpagetrue}%
9098             }%
9099             }%
9100           \listxadd{\@msdata@list}{%
9101             \@msd@options@iffullpage%
9102             \ifluatex%
9103               \csxdef{footnote@luatextextdir}{\the\textdir}%
9104               \csxdef{footnote@luatexpardir}{\the\pardir}%

```



```

9105 \fi%
9106 \csdef{@this@crossref@start}{\add@msd@c:start:msdata}%
9107 \csdef{@this@crossref@end}{\add@msd@c:end:msdata}%
9108 \unexpanded{%
9109 \def\annot@start{}%
9110 \def\annot@end{}%
9111 }%
9112 \noexpand\csuse{v\@msdata@series footnote}{\@msdata@series}{\@
expandonce\l@d@nums}{\ms@data@label}{\expandonce\data}}%
9113 \reset@msd@options@iffullpage%
9114 }%
9115 }%
9116 {\global\@msdata@insertedfrompreviouspagefalse}%
9117 }{}%
9118 \else%
9119 \ifcsdef{@msdata@\add@msd@cR @dataR}{%
9120 \ifnumless{\the\absline@numR-\@ne}{\csuse{@msdata@\add@msd@cR
@stopabslinenumberR}}%
9121 {%
9122 \global\@msdata@insertedfrompreviouspagetrue%
9123 \letcs{\@data}{@msdata@\add@msd@cR @dataR}%
9124 \edef\l@d@nums{%
9125 000}% Start page = we don't print it
9126 \numexpr\the\line@numR+\@ne\relax}% Start line number = first
line of the page. As \add@msdata@firstlineofpage is called before line
number has been incremented, we increment it for printing
9127 000}% Start subline number, for now, not used
9128 000}% End page number, we don't print it
9129 \ifnumless{\csuse{@msdata@\add@msd@cR @stopabslinenumberR}}{\@
csuse{@lastline@forpageR@\the\page@numR}}%
9130 {\csuse{@msdata@\add@msd@cR @stoplinenumberR}}%End line number
if in the same page
9131 {\csuse{@lastline@forpageR@\the\page@numR}}%Otherwise, last
number of the page
9132 |%
9133 000}% End sub line number, for now, not used
9134 \edfont@info%Font
9135 }%
9136 \@msd@options@fullpagefalse%
9137 \ifnumless{\csuse{@lastabsline@forpageR@\the\page@numR}}{\@csuse{
@msdata@\add@msd@cR @stopabslinenumberR}+\@ne}%
9138 {%
9139 \numdef{\@tmp}{\add@msd@cR+\@ne}%
9140 \ifcsdef{@msdata@\@tmp @abslinenumberR}%
9141 {\ifnumequal{\csuse{@msdata@\@tmp @abslinenumberR}}{\@csuse{
@lastabsline@forpageR@\the\page@numR}}%
9142 {}%
9143 {\@msd@options@fullpagetrue}%
9144 }%
9145 {\@msd@options@fullpagetrue}%

```

```

9146 }%
9147 {}%
9148 \listxadd{\@msdata@list}{%
9149 \@msd@options@iffullpage%
9150 \ifluatex%
9151 \csxdef{footnote@luatextextdir}{\the\textdir}%
9152 \csxdef{footnote@luatexpardir}{\the\pardir}%
9153 \fi%
9154 \csdef{@this@crossref@start}{\add@msd@cR:start:msdata}%
9155 \csdef{@this@crossref@end}{\add@msd@cR:end:msdata}%
9156 \unexpanded{%
9157 \def\annot@start{%
9158 \def\annot@end{%
9159 }%
9160 \noexpand\csuse{v\@msdata@series footnote}{\@msdata@series}{\{
expandonce\l@d@nums}{\ms@data@label}{\expandonce\@data}}%
9161 \reset@msd@options@iffullpage%
9162 }%
9163 }%
9164 {\global\@msdata@insertedfrompreviouspagefalse}%
9165 }{}%
9166 \fi%
9167 \egroup%
9168 }%
9169 %

```

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 70 \newcommand{\insert@msdata}{%
9171 \def\do##1{##1}%
9172 \dolistloop{\@msdata@list}%
9173 \global\let\@msdata@list\relax%
9174 }%
9175 %

```

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.

```

9176 \newif\if@msd@options@fullpage%
9177 \newcommand{\@msd@options@iffullpage}[0]{%
9178 \if@msd@options@fullpage%
9179 \noexpand\toggletrue{nonum@}%

```

```

9180 \ifdefvoid{\ms@data@label}%
9181 {\noexpand\toggletrue{nosep@}}%
9182 {}%
9183 \fi%
9184 }%
9185 \newcommand{\reset@msd@options@iffullpage}[0]{%
9186 \noexpand\togglefalse{nonum@}%
9187 \noexpand\togglefalse{nosep@}%
9188 }%
9189 %

```

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.

```

9190 \newtoks\@emptytoks
9191
9192 %

```

The rest is from `amsmath`.

`\l@denvbody` A token register to contain the body.

```

9193 \newtoks\l@denvbody
9194
9195 %

```

`\addtol@denvbody` `\addtol@denvbody{arg}` adds `arg` to the token register `\l@denvbody`.

```

9196 \newcommand{\addtol@denvbody}[1]{%
9197 \global\l@denvbody\expandafter{\the\l@denvbody#1}}
9198 %

```

`\if@edtabular` A boolean set to true inside a `reledmac`’ tabular environment.

```

9199 \newif\if@edtabular%
9200 %

```

`\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`.

```

9201 \newcommand{\l@dcollect@body}[1]{%
9202   \ifvmode\else%
9203   \\\%Ensure that we do not have spurious indentation at the very first
line, caused by some text before.
9204   \fi%
9205   \l@denbody{\expandafter#1\expandafter{\the\l@denbody}}%
9206   \edef\processl@denbody{\the\l@denbody\noexpand\end{\@currenvir}}%
9207   \l@denbody\@emptytoks \def\l@dbegin@stack{b}%
9208   \begingroup
9209   \edtabulartrue%
9210   \expandafter\let\csname\@currenvir\endcsname\l@dcollect@@body
9211   \edef\processl@denbody{\expandafter\noexpand\csname\@currenvir\
endcsname}%
9212   \processl@denbody%
9213   }%
9214
9215 %

```

`\l@dpush@begins` When adding a piece of the current environment's contents to `\l@denbody`, we scan it to check for additional `\begin` tokens, and add a "b" to the stack for any that we find.

```

9216 \def\l@dpush@begins#1\begin#2{%
9217   \ifx\end#2\else b\expandafter\l@dpush@begins\fi}
9218
9219 %

```

`\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.

```

9220 \def\l@dcollect@@body#1\end#2{%
9221   \edef\l@dbegin@stack{\l@dpush@begins#1\begin\end
\expandafter\@gobble\l@dbegin@stack}%
9222   \ifx\@empty\l@dbegin@stack
9223     \endgroup
9224     \@checkend{#2}%
9225     \addtol@denbody{#1}%
9226   \else
9227     \addtol@denbody{#1\end{#2}}%
9228   \fi
9229   \processl@denbody % A little tricky! Note the grouping
9230

```

```

9231 }
9232
9233 %

```

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.

```

9234 \newcommand*{\l@dtabnoexpands}{%
9235   \let\rtab=0%
9236   \let\ctab=0%
9237   \let\ltab=0%
9238   \let\rtabtext=0%
9239   \let\ltabtext=0%
9240   \let\ctabtext=0%
9241   \let\edbeforetab=0%
9242   \let\edaftertab=0%
9243   \let\edatleft=0%
9244   \let\edatright=0%
9245   \let\edvertline=0%
9246   \let\edvertdots=0%
9247   \let\edrowfill=0%
9248 }
9249
9250 %

```

`\disable@familiarnotes` Macros to disable and restore familiar notes, to prevent them from printing multiple times in `edtabularx` and `edarrayx` environments.

`\restore@familiarnotes`

```

9251 \newcommand{\disable@familiarnotes}{%
9252   \unless\ifnofamiliar%
9253     \def\do##1{%
9254       \csletcs{footnote@##1}{footnote##1}%
9255       \expandafter\renewcommand \csname footnote##1\endcsname[1]{%
9256         \protected@csxdef{@thefnmark##1}{\csuse{thefootnote##1}}%
9257         \csuse{@footnotemark##1}%
9258       }%
9259     }%
9260     \dolistloop{\@series}%
9261     \fi%
9262   }%
9263 \newcommand{\restore@familiarnotes}{%
9264   \unless\ifnofamiliar%
9265     \def\do##1{%

```

```

9266         \csletcs{footnote##1}{footnote@@##1}%
9267     }%
9268     \dolistloop{\@series}%
9269 \fi%
9270 }%
9271
9272 %

```

`\disable@sidenotes` The same for sidenotes.

```

\restore@sidenotes
9273 \newcommand{\disable@sidenotes}{%
9274     \let\@@ledrightnote\ledrightnote%
9275     \let\@@ledleftnote\ledleftnote%
9276     \let\@@ledsidenote\ledsidenote%
9277     \let\ledrightnote@gobble%
9278     \let\ledleftnote@gobble%
9279     \let\ledsidenote@gobble%
9280 }%
9281 \newcommand{\restore@sidenotes}{%
9282     \let\ledrightnote\@@ledrightnote%
9283     \let\ledleftnote\@@ledleftnote%
9284     \let\ledsidenote\@@ledsidenote%
9285 }%
9286 %

```

`\disable@endnotes` The same for endnotes.

```

\restore@endnotes
9287 \newcommand{\disable@endnotes}{%
9288     \unless\ifnoend@%
9289         \def\do##1{%
9290             \csletcs{##1@endnote}{##1endnote}%
9291             \expandafter\renewcommand \csname ##1endnote\endcsname[1]{}%
9292         }%
9293         \dolistloop{\@series}%
9294     \fi%
9295 }%
9296 \newcommand{\restore@endnotes}{%
9297     \unless\ifnofamiliar@%
9298         \def\do##1{%
9299             \csletcs{##1endnote}{##1@endnote}%
9300         }%
9301         \dolistloop{\@series}%
9302     \fi%
9303 }%
9304 %

```

`\disable@notes` Disable/restore side, familiar and end notes.

```

\restore@notes
9305 \newcommand{\disable@notes}{%
9306     \disable@sidenotes%

```



```

9307 \disable@familiarnotes%
9308 \disable@endnotes%
9309 }%
9310 \newcommand{\restore@notes}{%
9311 \restore@sidenotes%
9312 \restore@familiarnotes%
9313 \restore@endnotes%
9314 }%
9315 %

```

\EDTEXT We need to be able to modify the `\edtext` macros and also restore their original definitions.

\xedtext

```

9316 \let\EDTEXT=\edtext
9317 \newcommand{\xedtext}[2]{\EDTEXT{#1}{#2}}
9318 %

```

\EDLABEL We need to be able to modify and restore the `\edlabel` macro.

\xedlabel

```

9319 \let\EDLABEL=\edlabel
9320 \newcommand*{\xedlabel}[1]{\EDLABEL{#1}}
9321 %

```

\xedindex Macros supporting modification and restoration of `\edindex`.

\nulledindex

```

9322 \AtBeginDocument{\let\xedindex\edindex}%
9323 \newcommand{\nulledindex}[2][\jobname]{\@bsphack\@esphack}
9324
9325 %

```

\@line@num Macro supporting restoration of `\linenum`.

```

9326 \let\@line@num=\linenum
9327 %

```

\l@dgobbleoptarg `\l@dgobbleoptarg[⟨arg⟩]{⟨arg⟩}` replaces these two arguments (first is optional) by `\relax`.

```

9328 \newcommand*{\l@dgobbleoptarg}[2][\relax]%
9329
9330 %

```

\l@secondmandarg `\l@secondoptarg[⟨arg⟩]{⟨arg⟩}` gobble the first (optional) argument, and expand to the second (mandatory) argument.

```

9331 \NewExpandableDocumentCommand{\l@secondmandarg}{om}{#2}%
9332 %

```

```
\Relax9333 \let\Relax=\relax
```

```
\NEXT9334 \let\NEXT=\next
```

```
9335
```

```
9336 %
```

\l@dmodforedtext Modify and restore various macros for when \edtext is used.

\l@drestoreforedtext

```
9337 \newcommand{\l@dmodforedtext}{%
```

```
9338 \let\edtext=\relax
```

```
9339 \def\do##1{\global\csletcs{##1footnote}{\l@gobbleoptarg}}%
```

```
9340 \dolistloop{\@series}%
```

```
9341 \let\edindex\nulledindex
```

```
9342 \let\linenum\@gobble}
```

```
9343 \newcommand{\l@drestoreforedtext}{%
```

```
9344 \def\do##1{\global\csletcs{##1footnote}{##1@footnote}}
```

```
9345 \dolistloop{\@series}%
```

```
9346 \let\edindex\xedindex}
```

```
9347 %
```

\l@dnullfills Nullify and restore some column fillers, etc.

\l@drestorefills

```
9348 \newcommand{\l@dnullfills}{%
```

```
9349 \def\edlabel##1{}
```

```
9350 \def\edrowfill##1##2##3{}
```

```
9351 }
```

```
9352 \newcommand{\l@drestorefills}{%
```

```
9353 \def\edrowfill##1##2##3{\@EDROWFILL@{##1}{##2}{##3}}%
```

```
9354 }
```

```
9355
```

```
9356 %
```

\letsforverteilen Gathers some lets and other code that is common to the *verteilen* macros.

```
9357 \newcommand{\letsforverteilen}{%
```

```
9358 \let\edtext=\xedtext
```

```
9359 \let\edindex=\xedindex
```

```
9360 \def\do##1{\global\csletcs{##1footnote}{##1@footnote}}
```

```
9361 \dolistloop{\@series}%
```

```
9362 \let\linenum\@line@num
```

```
9363 \hilfe\skip=\l@dcwidth%
```

```
9364 \advance\hilfe\skip by -\wd\hilfebox
```

```
9365 \def\edlabel##1{\xedlabel{##1}}
```

```
9366
```

```
9367 %
```

\disablel@dtabfeet Declarations for using or using \edtext inside tabulars. The default at this point is for

\enablel@dtabfeet \edtext.

```
9368 \newcommand\disablel@dtabfeet{\l@dmodforedtext}%
```

```
9369 \newcommand\enablel@dtabfeet{\l@drestoreforedtext}%
```

```
9370 %
```

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.

```

9371 \newcount\l@dampcount
9372 \l@dampcount=1\relax
9373 \newcount\l@dcolcount
9374 \l@dcolcount=0\relax
9375
9376 %

```

`\hilfsbox` Some (temporary) helper items.

```

\hilfsskip
\Hilfsbox
\hilfscount
9377 \newbox\hilfsbox
9378 \newskip\hilfsskip
9379 \newbox\Hilfsbox
9380 \newcount\hilfscount
9381
9382 %

```

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).

```

9383 \newdimen\dcoli
9384 \newdimen\dcolii
9385 \newdimen\dcoliii
9386 \newdimen\dcoliv
9387 \newdimen\dcolv
9388 \newdimen\dcolvi
9389 \newdimen\dcolvii
9390 \newdimen\dcolviii
9391 \newdimen\dcolix
9392 \newdimen\dcolx
9393 \newdimen\dcolxi
9394 \newdimen\dcolxii
9395 \newdimen\dcolxiii
9396 \newdimen\dcolxiv
9397 \newdimen\dcolxv
9398 \newdimen\dcolxvi
9399 \newdimen\dcolxvii
9400 \newdimen\dcolxviii
9401 \newdimen\dcolxix
9402 \newdimen\dcolxx
9403 \newdimen\dcolxxi
9404 \newdimen\dcolxxii
9405 \newdimen\dcolxxiii
9406 \newdimen\dcolxxiv
9407 \newdimen\dcolxxv
9408 \newdimen\dcolxxvi
9409 \newdimen\dcolxxvii

```

```

9410 \newdimen\dcollxxviii
9411 \newdimen\dcollxxix
9412 \newdimen\dcollxxx
9413 \newdimen\dcollerr    % added for error handling
9414
9415 %

```

\l@dcollwidth This is a cunning way of storing the columnwidths indexed by the column number \l@dcollcount, like an array. (was \Dimenzuordnung)

```

9416 \newcommand{\l@dcollwidth}{\ifcase \the\l@dcollcount \dcoli %???
9417 \or \dcoli \or \dcolii \or \dcoliii
9418 \or \dcoliv \or \dcolv \or \dcolvi
9419 \or \dcolvii \or \dcolviii \or \dcolix \or \dcolx
9420 \or \dcolxi \or \dcolxii \or \dcolxiii
9421 \or \dcolxiv \or \dcolxv \or \dcolxvi
9422 \or \dcolxvii \or \dcolxviii \or \dcolxix \or \dcolxx
9423 \or \dcolxxi \or \dcolxxii \or \dcolxxiii
9424 \or \dcolxxiv \or \dcolxxv \or \dcolxxvi
9425 \or \dcolxxvii \or \dcolxxviii \or \dcolxxix \or \dcolxxx
9426 \else \dcollerr \fi}
9427
9428 %

```

\step1@dcollcount This increments the column counter, and issues an error message if it is too large.

```

9429 \newcommand*\step1@dcollcount{\advance\l@dcollcount\@ne
9430 \ifnum\l@dcollcount>30\relax
9431 \led@err@TooManyColumns
9432 \fi}
9433
9434 %

```

\l@dssetmaxcollwidth Sets the column width to the maximum value seen so far.

```

9435 \newcommand{\l@dssetmaxcollwidth}{%
9436 \ifdim\l@dcollwidth < \wd\hilfsbox
9437 \l@dcollwidth = \wd\hilfsbox
9438 \else \relax \fi}
9439
9440 %

```

\measuremcell Measure (recursively) the width required for a math cell.

```

9441 \def\measuremcell #1&{%
9442 \ifx #1\ \ifnum\l@dcollcount=0\let\NEXT\relax%
9443 \else\l@dscheckcols%
9444 \l@dcollcount=0%
9445 \let\NEXT\measuremcell%
9446 \fi%

```

```

9447 \else\setbox\hilfsbox=\hbox{$\displaystyle{#1}$}%
9448 \step1@dcolcount%
9449 \l@setmaxcolwidth%
9450 \let\NEXT\measuremcell%
9451 \fi\NEXT}
9452
9453 %

```

\measuretcell Measure (recursively) the width required for a text cell.

```

9454 \def\measuretcell #1{%
9455 \ifx #1\ \ifnum\l@dcolcount=0\let\NEXT\relax%
9456 \else\l@dcheckcols%
9457 \l@dcolcount=0%
9458 \let\NEXT\measuretcell%
9459 \fi%
9460 \else\setbox\hilfsbox=\hbox{#1}%
9461 \step1@dcolcount%
9462 \l@setmaxcolwidth%
9463 \let\NEXT\measuretcell%
9464 \fi\NEXT}
9465
9466 %

```

\measuremrow Measure (recursively) the width required for a math row.

```

9467 \def\measuremrow #1{%
9468 \ifx #1&\let\NEXT\relax%
9469 \else\measuremcell #1&\&\&%
9470 \let\NEXT\measuremrow%
9471 \fi\NEXT}
9472 %

```

\measuretrow Measure (recursively) the width required for a text row.

```

9473 \def\measuretrow #1{%
9474 \ifx #1&\let\NEXT\relax%
9475 \else\measuretcell #1&\&\&%
9476 \let\NEXT\measuretrow%
9477 \fi\NEXT}
9478
9479 %

```

\edtabcolsep The length \edtabcolsep controls the distance between columns.

```

9480 \newskip\edtabcolsep
9481 \global\edtabcolsep=10pt
9482
9483 %

```

```
\variab84 \newcommand{\variab}{\relax}
```

```
9485
```

```
9486 %
```

\l@dccheckcols Check that the number of columns is consistent.

```
9487 \newcommand*{\l@dccheckcols}{%
9488   \ifnum\l@dcolcount=1\relax
9489   \else
9490     \ifnum\l@dampcount=1\relax
9491     \else
9492       \ifnum\l@dcolcount=\l@dampcount\relax
9493       \else
9494         \l@dcerr@UnequalColumns
9495       \fi
9496     \fi
9497     \l@dampcount=\l@dcolcount
9498   \fi}
9499
9500 %
```

\edfilldimen A length.

```
9501 \newdimen\edfilldimen
9502 \edfilldimen=0pt
9503
9504 %
```

\c@addcolcount A counter to hold the number of a column. We use a roman number so that we can grab the column dimension from `\dcol`. We do not use the `\roman` \TeX command, because some packages, like `babel` can override it in some specific cases (Greek, for example).

\theadcolcount

```
9505 \newcounter{addcolcount}
9506 \renewcommand{\theadcolcount}{\romannumeral \c@addcolcount}
9507 %
```

XXXII.2.3 Tabular typesetting

\setmcellright Typeset (recursively) cells of display math right justified.

```
9508 \def\setmcellright #1&{\def\edlabel##1}%
9509   \let\edindex\nulledindex
9510   \ifx #1\\ \ifnum\l@dcolcount=0%\removelastskip
9511     \let\Next\relax%
9512   \else\l@dcolcount=0%
9513     \let\Next=\setmcellright%
9514   \fi%
9515 \else%
9516   \disablel@dtabfeet%
```

```

9517         \stepl@dc colcount%
9518         \disable@notes%
9519         \setbox\hilfsbox=\hbox{$\displaystyle{#1}$}%
9520         \restore@notes%
9521         \letsforverteilen%
9522         \hskip\hilfsskip$\displaystyle{#1}$%
9523         \hskip\edtabcolsep%
9524         \let\Next=\setmcellright%
9525     \fi\Next}
9526
9527 %

```

\settcclright Typeset (recursively) cells of text right justified.

```

9528 \def\settcclright #1&{\def\edlabel##1{}%
9529     \let\edindex\nulledindex
9530     \ifx #1\ \ifnum\l@dc colcount=0%\removelastskip
9531         \let\Next\relax%
9532     \else\l@dc colcount=0%
9533         \let\Next=\settcclright%
9534     \fi%
9535 \else%
9536     \disablel@dtabfeet%
9537     \stepl@dc colcount%
9538     \disable@notes%
9539     \setbox\hilfsbox=\hbox{#1}%
9540     \restore@notes%
9541     \letsforverteilen%
9542     \hskip\hilfsskip#1%
9543     \hskip\edtabcolsep%
9544     \let\Next=\settcclright%
9545 \fi\Next}
9546 %

```

\setmcellleft Typeset (recursively) cells of display math left justified.

```

9547 \def\setmcellleft #1&{\def\edlabel##1{}%
9548     \let\edindex\nulledindex
9549     \ifx #1\ \ifnum\l@dc colcount=0 \let\Next\relax%
9550     \else\l@dc colcount=0%
9551         \let\Next=\setmcellleft%
9552     \fi%
9553 \else \disablel@dtabfeet%
9554     \stepl@dc colcount%
9555     \disable@notes%
9556     \setbox\hilfsbox=\hbox{$\displaystyle{#1}$}%
9557     \restore@notes%
9558     \letsforverteilen%
9559     $\displaystyle{#1}$\hskip\hilfsskip\hskip\edtabcolsep%
9560     \let\Next=\setmcellleft%

```

```

9561 \fi\Next}
9562
9563 %

```

\settcclleft Typeset (recursively) cells of text left justified.

```

9564 \def\settcclleft #1&{\def\edlabel##1{}}%
9565 \let\edindex\nulledindex
9566 \ifx #1\\ \ifnum\l@dc@lcount=0 \let\Next\relax%
9567 \else\l@dc@lcount=0%
9568 \let\Next=\settcclleft%
9569 \fi%
9570 \else \disablel@dtabfeet%
9571 \stepl@dc@lcount%
9572 \disable@notes%
9573 \setbox\hilfsbox=\hbox{#1}%
9574 \restore@notes%
9575 \letsforverteilen%
9576 #1\hskip\hilfsskip\hskip\edtabcolsep%
9577 \let\Next=\settcclleft%
9578 \fi\Next}
9579 %

```

\setmcellcenter Typeset (recursively) cells of display math centered.

```

9580 \def\setmcellcenter #1&{\def\edlabel##1{}}%
9581 \let\edindex\nulledindex
9582 \ifx #1\\ \ifnum\l@dc@lcount=0\let\Next\relax%
9583 \else\l@dc@lcount=0%
9584 \let\Next=\setmcellcenter%
9585 \fi%
9586 \else \disablel@dtabfeet%
9587 \stepl@dc@lcount%
9588 \disable@notes%
9589 \setbox\hilfsbox=\hbox{$\displaystyle{#1}$}%
9590 \restore@notes%
9591 \letsforverteilen%
9592 \hskip 0.5\hilfsskip$\displaystyle{#1}$\hskip0.5\hilfsskip%
9593 \hskip\edtabcolsep%
9594 \let\Next=\setmcellcenter%
9595 \fi\Next}
9596
9597 %

```

\settcclcenter Typeset (recursively) cells of text centered.

```

9598 \def\settcclcenter #1&{\def\edlabel##1{}}%
9599 \let\edindex\nulledindex
9600 \ifx #1\\ \ifnum\l@dc@lcount=0 \let\Next\relax%
9601 \else\l@dc@lcount=0%

```



```

9602         \let\Next=\settcellcenter%
9603         \fi%
9604     \else \disablel@dtabfeet%
9605         \stepl@dcolcount%
9606         \disable@notes%
9607         \setbox\hilfsbox=\hbox{#1}%
9608         \restore@notes%
9609         \letsforverteilen%
9610         \hskip 0.5\hilfsskip #1\hskip 0.5\hilfsskip%
9611         \hskip\edtabcolsep%
9612         \let\Next=\settcellcenter%
9613     \fi\Next}
9614
9615 %

```

\NEXT₁₆ \let\NEXT=\relax

```

9617
9618 %

```

\setmrowright Typeset (recursively) rows of right justified math.

```

9619 \def\setmrowright #1\{%
9620     \ifx #1& \let\NEXT\relax
9621     \else \centerline{\setmcellright #1&\\&\\&}
9622         \let\NEXT=\setmrowright
9623     \fi\NEXT}
9624 %

```

\settroright Typeset (recursively) rows of right justified text.

```

9625 \def\settroright #1\{%
9626     \ifx #1& \let\NEXT\relax
9627     \else \centerline{\settcellright #1&\\&\\&}
9628         \let\NEXT=\settroright
9629     \fi\NEXT}
9630
9631 %

```

\setmrowleft Typeset (recursively) rows of left justified math.

```

9632 \def\setmrowleft #1\{%
9633     \ifx #1&\let\NEXT\relax
9634     \else \centerline{\setmcellleft #1&\\&\\&}
9635         \let\NEXT=\setmrowleft
9636     \fi\NEXT}
9637 %

```

\settroright Typeset (recursively) rows of left justified text.

```

9638 \def\settrorleft #1\{\%
9639   \ifx #1& \let\NEXT\relax
9640   \else \centerline{\settcclleft #1&\&\&\&}
9641   \let\NEXT=\settrorleft
9642   \fi\NEXT}
9643
9644 %

```

\setmrowcenter Typeset (recursively) rows of centered math.

```

9645 \def\setmrowcenter #1\{\%
9646   \ifx #1& \let\NEXT\relax%
9647   \else \centerline{\setmcellcenter #1&\&\&\&}
9648   \let\NEXT=\setmrowcenter
9649   \fi\NEXT}
9650 %

```

\settrorcenter Typeset (recursively) rows of centered text.

```

9651 \def\settrorcenter #1\{\%
9652   \ifx #1& \let\NEXT\relax
9653   \else \centerline{\settrcellcenter #1&\&\&\&}
9654   \let\NEXT=\settrorcenter
9655   \fi\NEXT}
9656
9657 %

```

\nullsetzen 58 \newcommand{\nullsetzen}{\%

```

9659   \stepl@dc@colcount%
9660   \l@dc@colwidth=0pt%
9661   \ifnum\l@dc@colcount=30\let\NEXT\relax%
9662   \l@dc@colcount=0\relax
9663   \else\let\NEXT\nullsetzen%
9664   \fi\NEXT}
9665
9666 %

```

\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.

```

9667 \newcommand{\edatleft}[3][@empty]{%
9668   \ifx#1@empty
9669     \vbox to 10pt{\vss\hbox{$\left#2\vrule width0pt height #3
9670       depth 0pt \right. $\hss}\vfil}
9671   \else
9672     \vbox to 4pt{\vss\hbox{$#1\left#2\vrule width0pt height #3
9673       depth 0pt \right. $\}\vfil}
9674   \fi}
9675 %

```

`\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.

```
%676 \newcommand{\edatright}[3][\@empty]{%
%677   \ifx#1\@empty
%678     \vbox to 10pt{\vss\hbox{\$ \left.\vrule width0pt height #3
%679       depth 0pt \right#2 \$\hss}\vfil}
%680   \else
%681     \vbox to 4pt{\vss\hbox{\$ \left.\vrule width0pt height #3
%682       depth 0pt \right#2 #1 \$}\vfil}
%683   \fi}
%684
%685 %
```

`\edvertline` `\edvertline{ $\langle len \rangle$ }` vertical line $\langle len \rangle$ high.

```
%686 \newcommand{\edvertline}[1]{\vbox to 8pt{\vss\hbox{\vrule height #1}\vfil}}
%687
%688 %
```

`\edvertdots` `\edvertdots{ $\langle len \rangle$ }` vertical dotted line $\langle len \rangle$ high.

```
%689 \newcommand{\edvertdots}[1]{\vbox to 1pt{\vss\vbox to #1%
%690   {\cleaders\hbox{\$ \m@th\hbox{.}\vbox to 0.5em{ }$}\vfil}}}
%691
%692 %
```

`\l@dtabaddcols` `\l@dtabaddcols{ $\langle startcol \rangle$ }{ $\langle endcol \rangle$ }` adds the widths of the columns $\langle startcol \rangle$ through $\langle endcol \rangle$ to `\edfilldimen`. It is a \LaTeX style reimplementaion of the original `\@add@`.

```
%693 \newcommand{\l@dtabaddcols}[2]{%
%694   \l@dccheckstartend{#1}{#2}%
%695   \ifl@dstartendok
%696     \setcounter{addcolcount}{#1}%
%697     \@whilenum \value{addcolcount}<#2\relax \do
%698       {\advance\edfilldimen by \the \csname dcol\theadcolcount\endcsname
%699         \advance\edfilldimen by \edtabcolsep
%700         \stepcounter{addcolcount}}%
%701     \advance\edfilldimen by \the \csname dcol\theadcolcount\endcsname
%702   \fi
%703 }
%704
%705 %
```

`\ifl@dstartendok` `\l@dccheckstartend{ $\langle startcol \rangle$ }{ $\langle endcol \rangle$ }` checks that the values of $\langle startcol \rangle$ and $\langle endcol \rangle$ are sensible. If they are then `\ifl@dstartendok` is set TRUE, otherwise it is set FALSE.

```
%706 \newif\ifl@dstartendok
%707 \newcommand{\l@dccheckstartend}[2]{%
```

```

9708 \l@dstartendoktrue
9709 \ifnum #1<\@ne
9710   \l@dstartendokfalse
9711   \led@err@LowStartColumn
9712 \fi
9713 \ifnum #2>30\relax
9714   \l@dstartendokfalse
9715   \led@err@HighEndColumn
9716 \fi
9717 \ifnum #1>#2\relax
9718   \l@dstartendokfalse
9719   \led@err@ReverseColumns
9720 \fi
9721 }
9722
9723 %

```

`\edrowfill` `\edrowfill{<startcol>}{<endcol>}` fill fills columns `<startcol>` to `<endcol>` inclusive with `<fill>` (e.g. `\hrulefill`, `\upbracefill`). This is a \TeX style reimplementation and generalization of the original `\waklam`, `\Waklam`, `\waklamec`, `\wastricht` and `\wapunktel` macros.

```

9724 \newcommand*{\edrowfill}[3]{%
9725   \l@dtabaddcols{#1}{#2}%
9726   \hb@xt@ \the\l@dcolwidth{\hb@xt@ \the\edfilldimen{#3}\hss}}
9727 \let\@edrowfill=\edrowfill
9728 \def\@EDROWFILL@#1#2#3{\@edrowfill@{#1}{#2}{#3}}
9729
9730 %

```

`\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`.

```

9731 \newcommand{\leftltab}[1]{%
9732   \hb@xt@\z@{\vbox{\edtabindent%
9733     \moveleft\Hilfsskip\hbox{\ #1}}\hss}}
9734
9735 %

```

`\leftrtab` `\leftrtab{<text>}{<math>}` for `\edbeforetab` in `\rtab`.

```

9736 \newcommand{\leftrtab}[2]{%
9737   #2\hb@xt@\z@{\vbox{\edtabindent%
9738     \advance\Hilfsskip by\dcoli%
9739     \moveleft\Hilfsskip\hbox{\ #1}}\hss}}
9740
9741 %

```

`\leftctab` `\leftctab{<text>}{<math>}` for `\edbeforetab` in `\ctab`.

```

9742 \newcommand{\leftctab}[2]{%
9743     \hb@xt@\z@{\vbox{\edtabindent\l@dcolcount=\l@dampcount%
9744     \advance\Hilfsskip by 0.5\dcoli%
9745     \setbox\hilfsbox=\hbox{\def\edlabel##1{}%
9746     \disablel@dtabfeet$\displaystyle{#2}$}%
9747     \advance\Hilfsskip by -0.5\wd\hilfsbox%
9748     \moveleft\Hilfsskip\hbox{\ #1}}\hss}%
9749     #2}
9750
9751 %

```

`\rightctab` `\rightctab{<math>}{<text>}` for `\edaftertab` in `\ctab`.

```

9752 \newcommand{\rightctab}[2]{%
9753     \setbox\hilfsbox=\hbox{\def\edlabel##1{}%
9754     \disablel@dtabfeet#2}\l@dampcount=\l@dcolcount%
9755     #1\hb@xt@\z@{\vbox{\edtabindent\l@dcolcount=\l@dampcount%
9756     \advance\Hilfsskip by 0.5\l@dcolwidth%
9757     \advance\Hilfsskip by -\wd\hilfsbox%
9758     \setbox\hilfsbox=\hbox{\def\edlabel##1{}%
9759     \disablel@dtabfeet$\displaystyle{#1}$}%
9760     \advance\Hilfsskip by -0.5\wd\hilfsbox%
9761     \advance\Hilfsskip by \edtabcolsep%
9762     \moveright\Hilfsskip\hbox{ #2}}\hss}%
9763     }
9764
9765 %

```

`\rightltab` `\rightltab{<math>}{<text>}` for `\edaftertab` in `\ltab`.

```

9766 \newcommand{\rightltab}[2]{%
9767     \setbox\hilfsbox=\hbox{\def\edlabel##1{}%
9768     \disablel@dtabfeet#2}\l@dampcount=\l@dcolcount%
9769     #1\hb@xt@\z@{\vbox{\edtabindent\l@dcolcount=\l@dampcount%
9770     \advance\Hilfsskip by\l@dcolwidth%
9771     \advance\Hilfsskip by-\wd\hilfsbox%
9772     \setbox\hilfsbox=\hbox{\def\edlabel##1{}%
9773     \disablel@dtabfeet$\displaystyle{#1}$}%
9774     \advance\Hilfsskip by-\wd\hilfsbox%
9775     \advance\Hilfsskip by\edtabcolsep%
9776     \moveright\Hilfsskip\hbox{ #2}}\hss}%
9777     }
9778
9779 %

```

`\rightrtab` `\rightrtab{<math>}{<text>}` for `\edaftertab` in `\rtab`.

```

9780 \newcommand{\rightrtab}[2]{%

```

```

9781 \setbox\hilfsbox=\hbox{\def\edlabel##1}{%
9782 \disablel@dtabfeet#2}%
9783 #1\hb@xt@{z}{\vbox{\edtabindent%
9784 \advance\Hilfsskip by-\wd\hilfsbox%
9785 \advance\Hilfsskip by\edtabcolsep%
9786 \moveright\Hilfsskip\hbox{ #2}}\hss}%
9787 }
9788
9789 %

```

\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.

```

9790 \newcommand{\rtab}[1]{%
9791 \l@dnnullfills
9792 \def\edbeforetab##1##2{\lefttrtab{##1}{##2}}%
9793 \def\edaftertab##1##2{\righttrtab{##1}{##2}}%
9794 \measurembody{#1}%
9795 \l@drestorefills
9796 \variab
9797 \setmrowright #1\&\%
9798 \enablel@dtabfeet}
9799
9800 %

```

\measurembody `\measurembody{<body>}` measures the array `<body>`.

```

9801 \newcommand{\measurembody}[1]{%
9802 \disablel@dtabfeet%
9803 \l@dcolcount=0%
9804 \nullsetzen%
9805 \l@dcolcount=0
9806 \measuremrow #1\&\%
9807 \global\l@dampcount=1}
9808
9809 %

```

\rtabtext `\rtabtext{<body>}` typesets `<body>` as a tabular with the entries right justified.

```

9810 \newcommand{\rtabtext}[1]{%
9811 \l@dnnullfills
9812 \measuretbody{#1}%
9813 \l@drestorefills
9814 \variab
9815 \settrrowright #1\&\%
9816 \enablel@dtabfeet}
9817
9818 %

```

`\measuretbody` `\measuretbody{<body>}` measures the tabular `<body>`.

```

9819 \newcommand{\measuretbody}[1]{%
9820   \disable@notes%
9821   \disablel@dtabfeet%
9822   \l@dc@colcount=0%
9823   \nullsetzen%
9824   \l@dc@colcount=0
9825   \measuretrrow #1\\&\\%
9826   \restore@notes%
9827   \global\l@dampcount=1}
9828
9829 %

```

`\ltab` Array with entries left justified.

```

\edbeforetab \newcommand{\ltab}[1]{%
\edaftertab  \l@dnullfills
9830
9831   \def\edbeforetab##1##2{\leftltab{##1}{##2}}%
9832   \def\edaftertab##1##2{\rightltab{##1}{##2}}%
9833   \measuretbody{#1}%
9834   \l@drestorefills
9835   \variab
9836   \setmrowleft #1\\&\\%
9837   \enablel@dtabfeet}
9838
9839 %
9840 %

```

`\ltabtext` Tabular with entries left justified.

```

9841 \newcommand{\ltabtext}[1]{%
9842   \l@dnullfills
9843   \measuretbody{#1}%
9844   \l@drestorefills
9845   \variab
9846   \settrrowleft #1\\&\\%
9847   \enablel@dtabfeet}
9848
9849 %

```

`\ctab` Array with centered entries.

```

\edbeforetab \newcommand{\ctab}[1]{%
\edaftertab  \l@dnullfills
9850
9851   \def\edbeforetab##1##2{\leftctab{##1}{##2}}%
9852   \def\edaftertab##1##2{\rightctab{##1}{##2}}%
9853   \measuretbody{#1}%
9854   \l@drestorefills
9855   \variab
9856   \setmrowcenter #1\\&\\%
9857

```

```

9858 \enablel@dtabfeet}
9859
9860 %

```

\ctabtext Tabular with entries centered.

```

9861 \newcommand{\ctabtext}[1]{%
9862   \l@dnnullfills
9863   \measuretbody{#1}%
9864   \l@drestorefills
9865   \variab
9866   \settrrowcenter #1\\&\\%
9867   \enablel@dtabfeet}
9868
9869 %

```

\spreadtext 70 \newcommand{\spreadtext}[1]{%\l@dcolcount=\l@dampcount%
 \hb@xt@ \the\l@dcolwidth{\hbox{#1}\hss}}
 %

\spreadmath 73 \newcommand{\spreadmath}[1]{%
 \hb@xt@ \the\l@dcolwidth{\hbox{\$\displaystyle{#1}\$}\hss}}
 %

\HILFSskip More helpers.

\Hilfsskip
 9877 \newskip\HILFSskip
 9878 \newskip\Hilfsskip
 9879
 9880 %

\EDTABINDENT 81 \newcommand{\EDTABINDENT}{%
 9882 \ifnum\l@dcolcount=30\let\NEXT\relax\l@dcolcount=0%
 9883 \else\step\l@dcolcount%
 9884 \advance\Hilfsskip by\l@dcolwidth%
 9885 \ifdim\l@dcolwidth=0pt\advance\hilfscount\@ne
 9886 \else\advance\Hilfsskip by \the\hilfscount\edtabcolsep%
 9887 \hilfscount=1\fi%
 9888 \let\NEXT=\EDTABINDENT%
 9889 \fi\NEXT}%
 9890 %

\edtabindent (was \tabindent)


```

9891 \newcommand{\edtabindent}{%
9892   \l@dcolcount=0\relax
9893   \Hilfsskip=0pt%
9894   \hilfscount=1\relax
9895   \EDTABINDENT%
9896   \hilfsskip=\hsize%
9897   \advance\hilfsskip -\Hilfsskip%
9898   \Hilfsskip=0.5\hilfsskip%
9899   }%
9900
9901 %

```

\EDTAB (was \TAB)

```

9902 \def\EDTAB #1|#2|{%
9903   \setbox\tabhilfbox=\hbox{$\displaystyle{#1}$}%
9904   \setbox\tabHilfbox=\hbox{$\displaystyle{#2}$}%
9905   \advance\tabelskip -\wd\tabhilfbox%
9906   \advance\tabelskip -\wd\tabHilfbox%
9907   \unhbox\tabhilfbox\hskip\tabelskip%
9908   \unhbox\tabHilfbox}%
9909
9910 %

```

\EDTABtext (was \TABtext)

```

9911 \def\EDTABtext #1|#2|{%
9912   \setbox\tabhilfbox=\hbox{#1}%
9913   \setbox\tabHilfbox=\hbox{#2}%
9914   \advance\tabelskip -\wd\tabhilfbox%
9915   \advance\tabelskip -\wd\tabHilfbox%
9916   \unhbox\tabhilfbox\hskip\tabelskip%
9917   \unhbox\tabHilfbox}%
9918 %

```

\tabhilfbox Further helpers.

\tabHilfbox

```

9919 \newbox\tabhilfbox
9920 \newbox\tabHilfbox
9921
9922 %

```

XXXII.2.4 Environments

edarrayl edarrayc edarrayr The environment forms for \ltab, \ctab and \rtab.

```

9923 \newenvironment{edarrayl}{\l@dcollect@body\ltab}{}
9924 \newenvironment{edarrayc}{\l@dcollect@body\ctab}{}
9925 \newenvironment{edarrayr}{\l@dcollect@body\rtab}{}
9926
9927 %

```

edtabularl edtabularc edtabularr The environment forms for \ltabtext, \ctabtext and \rtabtext.

```

9928 \newenvironment{edtabularl}{\l@ddcollect@body\ltabtext}{}
9929 \newenvironment{edtabularc}{\l@ddcollect@body\ctabtext}{}
9930 \newenvironment{edtabularr}{\l@ddcollect@body\rtabtext}{}
9931
9932 %

```

XXXIII Quotation's commands

`\initnumbering@quote` This macro, called at the beginning of any numbered section, locally redefines the quotation and quote environments, in order to allow their use inside of numbered sections.

```

\quotation \initnumbering@quote defines quotation environment.
\endquotation
\quote
\endquote
9933 \newcommand{\initnumbering@quote}{
9934   \ifnoquotation@else
9935     \renewcommand{\quotation}{\par\leavevmode%
9936                                   \parindent=1.5em%
9937                                   \skipnumbering%
9938                                   \ifautopar%
9939                                     \vskip-\parskip%
9940                                   \else%
9941                                     \vskip\topsep%
9942                                   \fi%
9943                                   \global\leftskip=\leftmargin%
9944                                   \global\rightskip=\leftmargin%
9945                                   }
9946     \renewcommand{\endquotation}{\par%
9947                                   \global\leftskip=0pt%
9948                                   \global\rightskip=0pt%
9949                                   \leavevmode%
9950                                   \skipnumbering%
9951                                   \ifautopar%
9952                                     \vskip-\parskip%
9953                                   \else%
9954                                     \vskip\topsep%
9955                                   \fi%
9956                                   }
9957     \renewcommand{\quote}{\par\leavevmode%
9958                                   \parindent=0pt%
9959                                   \skipnumbering%
9960                                   \ifautopar%
9961                                     \vskip-\parskip%
9962                                   \else%
9963                                     \vskip\topsep%
9964                                   \fi%

```

```

9965             \global\leftskip=\leftmargin%
9966             \global\rightskip=\leftmargin%
9967     }
9968     \renewcommand{\endquote}{\par%
9969             \global\leftskip=0pt%
9970             \global\rightskip=0pt%
9971             \leavevmode%
9972             \skipnumbering%
9973             \ifautopar%
9974                 \vskip-\parskip%
9975             \else%
9976                 \vskip\topsep%
9977             \fi%
9978         }
9979     \fi
9980 }
9981 %

```

XXXIV Section's title commands

XXXIV.1 Commands to disable some feature

\ledsectnotoc The \ledsectnotoc only disables the \addcontentsline macro.

```

9982 \newcommand{\ledsectnotoc}{\let\addcontentsline\@gobblethree}
9983 %

```

\ledsectnomark The \ledsectnomark only disables the \chaptermark, \sectionmark and \subsectionmark macros.

```

9984 \newcommand{\ledsectnomark}{%
9985     \let\chaptermark\@gobble%
9986     \let\sectionmark\@gobble%
9987     \let\subsectionmark\@gobble%
9988 }
9989 %

```

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).

```
9990 \catcode`\#=12
9991 \notbool{@noeled@sec}{%
9992 %
```

`\beforeeledchapter` For technical reasons, not yet solved, page-breaking before chapters can't be made automatically by `eledmac`. Users have to use `\beforeeledchapter`.

```
9993 \ifl@dmemoir
9994   \newcommand\beforeeledchapter{%
9995     \clearforchapter%
9996   }
9997 \else
9998   \newcommand\beforeeledchapter{%
9999     \if@openright%
10000     \cleardoublepage%
10001   \else%
10002     \clearpage%
10003   \fi%
10004 }
10005 \fi
10006 %
```

XXXIV.4 Auxiliary commands

`\print@leftmargin@eledsection` `\print@leftmargin@eledsection` and `\print@rightmargin@eledsection` are added by `reledmac` inside the code of sectioning command, in order to affix lines numbers. They include tests for RTL languages.

```

10007 \def\print@rightmargin@eledsection{%
10008   \if@eled@sectioning%
10009     \begingroup%
10010     \if@RTL%
10011       \let\llap\rlap%
10012       \let\leftlinenum\rightlinenum%
10013       \let\leftlinenumR\rightlinenumR%
10014       \let\l@drd@ta\l@dld@ta%
10015       \let\l@drsn@te\l@dlsn@te%
10016     \fi%
10017     \hfill\l@drd@ta \csuse{LR}{\l@drsn@te}%
10018     \endgroup%
10019   \fi%
10020 }%
10021
10022 \def\print@leftmargin@eledsection{%
10023   \if@eled@sectioning%
10024     \leavevmode%
10025     \begingroup%
10026     \if@RTL%
10027       \let\rlap\llap%
10028       \let\rightlinenum\leftlinenum%
10029       \let\rightlinenumR\leftlinenumR%
10030       \let\l@dld@ta\l@drd@ta%
10031       \let\l@dlsn@te\l@drsn@te%
10032     \fi%
10033     \l@dld@ta\csuse{LR}{\l@dlsn@te}%
10034     \endgroup%
10035   \fi%
10036 }%
10037
10038 %

```

XXXIV.5 Patching standard commands

`\M@sect`
`\@mem@old@ssect`
`\@makechapterhead`
`\@makechapterhead`
`\@makeschapterhead`
`\@sect`
`\@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.

```

10039 \AtBeginDocument{%
10040
10041
10042 \pretocmd{\M@sect}

```

```

10043 {\let\old@edtext=\edtext%
10044 \let\edtext=\dummy@edtext@showlemma%
10045 }
10046 {}
10047 {}
10048
10049 \apptocmd{\M@sect}
10050 {\let\edtext=\old@edtext}
10051 {}
10052 {}
10053
10054 \patchcmd{\M@sect}
10055 { #9}
10056 { #9%
10057 \print@rightmargin@eledsection%
10058 }
10059 {}
10060 {}
10061
10062 \patchcmd{\M@sect}
10063 {\hskip #3\relax}
10064 {\hskip #3\relax%
10065 \print@leftmargin@eledsection%
10066 }
10067 {}
10068 {}
10069
10070 \patchcmd{\@mem@old@ssect}
10071 {#5}
10072 {#5%
10073 \print@leftmargin@eledsection%
10074 }
10075 {}
10076 {}
10077
10078 \patchcmd{\@mem@old@ssect}
10079 {\hskip #1}
10080 {\hskip #1%
10081 \print@rightmargin@eledsection%
10082 }
10083 {}
10084 {}
10085
10086
10087
10088 \patchcmd{\scr@startchapter}{\if@openright\cleardoublepage\else\clearpage\fi}{%
10089 \if@eled@sectioning\else%
10090 \ifl@dprintingpages\else%
10091 \if@openright\cleardoublepage\else\clearpage\fi}%No clearpage inside a

```

```

\Pages: will keep critical notes from printing on the title page. Here for
scrbook.
    \fi%
10092 \fi%
10093 \fi%
10094 }
10095 {}
10096 {}
10097
10098 \patchcmd{\@makechapterhead}
10099 {#1}
10100 {\print@leftmargin@eledsection%
10101 #1%
10102 \print@rightmargin@eledsection%
10103 }
10104 {}
10105 {}
10106
10107 \patchcmd{\@makechapterhead}% For BIDI
10108 {\if@RTL\raggedleft\else\raggedright\fi}%
10109 {\if@eled@sectioning\else%
10110 \if@RTL\raggedleft\else\raggedright\fi%
10111 \fi%
10112 }%
10113 {}%
10114 {}%
10115
10116 \patchcmd{\@makeschapterhead}
10117 {#1}
10118 {\print@leftmargin@eledsection%
10119 #1%
10120 \print@rightmargin@eledsection%
10121 }
10122 {}
10123 {}
10124
10125 \pretocmd{\@sect}
10126 {\let\old@edtext=\edtext
10127 \let\edtext=\dummy@edtext@showlemma%
10128 }
10129 {}
10130 {}
10131
10132 \apptocmd{\@sect}
10133 {\let\edtext=\old@edtext}
10134 {}
10135 {}
10136
10137 \pretocmd{\@ssect}
10138 {\let\old@edtext=\edtext%
10139 \let\edtext=\dummy@edtext@showlemma%

```

```

10140 }
10141 {}
10142 {}
10143
10144 \apptocmd{\@ssect}
10145 {\let\edtext=\old@edtext}
10146 {}
10147 {}
10148
10149 %

```

hyperref also redefines \@sect. That is why, when manipulating arguments, we patch \@sect and the same only if hyperref is not used. If it is, we patch the \NR commands.

```

10150 \@ifpackageloaded{nameref}{
10151
10152   \patchcmd{\NR@sect}
10153     {#8}
10154     {#8%
10155       \print@rightmargin@eledsection%
10156     }
10157     {}
10158     {}
10159
10160   \patchcmd{\NR@sect}
10161     {\hskip #3\relax}
10162     {\hskip #3\relax%
10163       \print@leftmargin@eledsection%
10164     }
10165     {}
10166     {}
10167
10168   \patchcmd{\NR@ssect}
10169     {#5}
10170     {#5%
10171       \print@rightmargin@eledsection%
10172     }
10173     {}
10174     {}
10175
10176   \patchcmd{\NR@ssect}
10177     {\hskip #1}
10178     {\hskip #1%
10179       \print@leftmargin@eledsection%
10180     }
10181     {}
10182     {}
10183   }%
10184   {
10185     \patchcmd{\@sect}
10186     {#8}

```



```

10187     {#8%
10188     \print@rightmargin@eledsection%
10189     }
10190     {}
10191     {}
10192
10193     \patchcmd{\@sect}
10194     {\hskip #3\relax}
10195     {\hskip #3\relax%
10196     \print@leftmargin@eledsection%
10197     }
10198     {}
10199     {}
10200
10201     \patchcmd{\@ssect}
10202     {#5}
10203     {#5%
10204     \print@rightmargin@eledsection%
10205     }
10206     {}
10207     {}
10208
10209     \patchcmd{\@ssect}
10210     {\hskip #1}
10211     {\hskip #1%
10212     \print@leftmargin@eledsection%
10213     }
10214     {}
10215     {}
10216     }%
10217 }%
10218 %

```

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 `#`.

```

10219 {}}%
10220 \protect\catcode`\#=6 %Space NEEDS by \catcode
10221 %

```

\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.

```

10222 \AtBeginDocument{%
10223 \patchcmd{\chapter}{\clearforchapter}{%
10224 \if@eled@sectioning\else%
10225 \ifl@dprintingpages\else%
10226 \clearforchapter%
10227 \fi%
10228 \fi%

```

```

10229 }%
10230 {}%
10231 {}%
10232
10233 \patchcmd{\chapter}{\if@openright\cleardoublepage\else\clearpage\fi}{%
10234 \if@eled@sectioning\else%
10235 \ifl@dprintingpages%
10236 \endgraf%
10237 \else%
10238 \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
10239 \fi%
10240 \fi%
10241 }%
10242 {}%
10243 {}%
10244 }%
10245 %

```

\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.

```

10246 \newif\if@eled@sectioning%
10247 %

```

We reopen a new `\notbool{@noeled@sec}` statement, as we will define the `\elesection` commands.

```

10248 \notbool{@noeled@sec}{%
10249 %

```

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.

```

10250 \newwrite\eled@sectioning@out
10251 %

```

\eledchapter **\eledsection** And now, the user sectioning commands, which write to the file, and also add content as a “normal” line.

```

10252 \eledsubsection \newcommand{\eledchapter}[2] [] {%
10253 \eledsubsubsection \disable@familiarnotes%
10254 \eledchapter* #2%
10255 \eledsection* \restore@familiarnotes%
10256 \eledsubsection* \ifledRcol%
\eledsubsubsection*

```

```

10257 \immediate\write\eled@sectioningR@out{%
10258 \string\eled@chapter{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR}{R}
10259 }%
10260 \else%
10261 \immediate\write\eled@sectioning@out{%
10262 \string\eled@chapter{#1}{\unexpanded{#2}}{\the\pstarts@read@L}{R}
10263 }%
10264 \fi%
10265 }
10266
10267 \newcommand{\eledsection}[2][{}]{%
10268 \disable@familiarnotes%
10269 #2%
10270 \restore@familiarnotes%
10271 \ifledRcol%
10272 \immediate\write\eled@sectioningR@out{%
10273 \string\eled@section{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR}{R}
10274 }%
10275 \else%
10276 \immediate\write\eled@sectioning@out{%
10277 \string\eled@section{#1}{\unexpanded{#2}}{\the\pstarts@read@L}{R}
10278 }%
10279 \fi%
10280 }
10281
10282 \newcommand{\eledsubsection}[2][{}]{%
10283 \disable@familiarnotes%
10284 #2%
10285 \restore@familiarnotes%
10286 \ifledRcol%
10287 \immediate\write\eled@sectioningR@out{%
10288 \string\eled@subsection{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR}{R}
10289 }%
10290 \else%
10291 \immediate\write\eled@sectioning@out{%
10292 \string\eled@subsection{#1}{\unexpanded{#2}}{\the\pstarts@read@L}{R}
10293 }%
10294 \fi%
10295 }
10296 \newcommand{\eledsubsubsection}[2][{}]{%
10297 \disable@familiarnotes%
10298 #2%
10299 \restore@familiarnotes%
10300 \ifledRcol%
10301 \immediate\write\eled@sectioningR@out{%
10302 \string\eled@subsubsection{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR}
10303 }{R}
10304 }%

```

```

10304 \else%
10305 \immediate\write\eled@sectioning@out{%
10306 \string\eled@subsubsection{#1}{\unexpanded{#2}}{\the\pstarts@read@L
10307 }{}{}}%
10308 \fi%
10309 }
10310
10311
10312 \WithSuffix\newcommand\eledchapter*[2][]{%
10313 \disable@familiarnotes%
10314 #2%
10315 \restore@familiarnotes%
10316 \ifledRcol%
10317 \immediate\write\eled@sectioningR@out{%
10318 \string\eled@chapter{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR}{*}{R}
10319 }%
10320 \else%
10321 \immediate\write\eled@sectioning@out{%
10322 \string\eled@chapter{#1}{\unexpanded{#2}}{\the\pstarts@read@L}{*}{}}%
10323 }%
10324 \fi%
10325 }
10326
10327 \WithSuffix\newcommand\eledsection*[2][]{%
10328 \disable@familiarnotes%
10329 #2%
10330 \restore@familiarnotes%
10331 \ifledRcol%
10332 \immediate\write\eled@sectioningR@out{%
10333 \string\eled@section{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR}{*}{R}
10334 }%
10335 \else%
10336 \immediate\write\eled@sectioning@out{%
10337 \string\eled@section{#1}{\unexpanded{#2}}{\the\pstarts@read@L}{*}{}}%
10338 }%
10339 \fi%
10340 }
10341
10342 \WithSuffix\newcommand\eledsubsection*[2][]{%
10343 \disable@familiarnotes%
10344 #2%
10345 \restore@familiarnotes%
10346 \ifledRcol%
10347 \immediate\write\eled@sectioningR@out{%
10348 \string\eled@subsection{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR}{*}{R}
10349 }%
10350 \else%
10351 \immediate\write\eled@sectioning@out{%

```

```

10352 \string\eled@subsection{#1}{\unexpanded{#2}}{\the\pstarts@read@L
10353 }{*}{}%
10354 }%
10355 \fi%
10356 }
10357 \WithSuffix\newcommand\eledsubsubsection*[2] [] {%
10358 \disable@familiarnotes%
10359 #2%
10360 \restore@familiarnotes%
10361 \ifledRcol%
10362 \immediate\write\eled@sectioningR@out{%
10363 \string\eled@subsubsection{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR
10364 }{*}{R}}
10365 }%
10366 \else%
10367 \immediate\write\eled@sectioning@out{%
10368 \string\eled@subsubsection{#1}{\unexpanded{#2}}{\the\pstarts@read@L
10369 }{*}{}%
10370 }%
10371 \fi%
10372 }
10373 %

```

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.

```

10372 \def\eled@chapter#1#2#3#4#5{%
10373 \ifstrempy{#4}%
10374 {%
10375 \ifstrempy{#1}%
10376 {%
10377 \csgdef{eled@sectioning@#3#5}{\let\edtext=\dummy@edtext@showlemma\
chapter{#2}}}%
10378 \csgdef{eled@sectmark@#3#5}{\let\edtext=\dummy@edtext{}\chaptermark
{#2}}}%
10379 }%Need for \pairs, because of using parbox.
10380 {%

```

```

10381 \csgdef{eled@sectioning@#3#5}{\let\edtext=\dummy@edtext@showlemma\
chapter[#1]{#2}}%
10382 \csgdef{eled@sectmark@#3#5}{\let\edtext=\dummy@edtext{}\chaptermark
{#2}}%Need for \pairs, because of using parbox.
10383 }%
10384 }%
10385 {%
10386 \ifstrempy{#1}%
10387 {\csgdef{eled@sectioning@#3#5}{\let\edtext=\dummy@edtext@showlemma\
chapter*{#2}}}%
10388 {\csgdef{eled@sectioning@#3#5}{\let\edtext=\dummy@edtext@showlemma\
chapter*{#1}{#2}}}%Bug in LaTeX!
10389 }%
10390 \listcsgadd{eled@sections#5@@}{#3}%
10391 }
10392 \def\eled@section#1#2#3#4#5{%
10393 \ifstrempy{#4}%
10394 {\ifstrempy{#1}%
10395 {%
10396 \csgdef{eled@sectioning@#3#5}{\section{#2}}%
10397 \csgdef{eled@sectmark@#3#5}{\let\edtext=\dummy@edtext{}\sectionmark
{#2}}%Need for \pairs, because of using parbox.
10398 }%
10399 {%
10400 \csgdef{eled@sectioning@#3#5}{\section[#1]{#2}}%
10401 \csgdef{eled@sectmark@#3#5}{\let\edtext=\dummy@edtext{}\sectionmark
{#1}}%Need for \pairs, because of using parbox.
10402 }%
10403 }%
10404 {\ifstrempy{#1}%
10405 {\csgdef{eled@sectioning@#3#5}{\section*{#2}}}%
10406 {\csgdef{eled@sectioning@#3#5}{\section*{#1}{#2}}}%Bug in LaTeX!
10407 }
10408 \listcsgadd{eled@sections#5@@}{#3}%
10409 }
10410 \def\eled@subsection#1#2#3#4#5{%
10411 \ifstrempy{#4}%
10412 {\ifstrempy{#1}%
10413 {%
10414 \csgdef{eled@sectioning@#3#5}{\subsection{#2}}%
10415 \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)
10416 }%
10417 {%
10418 \csgdef{eled@sectioning@#3#5}{\subsection[#1]{#2}}%
10419 \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)
10420 }%

```

```

10421 }%
10422 {\ifstrempy{#1}%
10423   {\csgdef{eled@sectioning@#3#5}{\subsection*{#2}}}%
10424   {\csgdef{eled@sectioning@#3#5}{\subsection*{#1}{#2}}}%Bug in LaTeX!
10425 }
10426 \listcsgadd{eled@sections#5@@}{#3}%
10427 }
10428 \def\eled@subsubsection#1#2#3#4#5{%
10429   \ifstrempy{#4}%
10430     {\ifstrempy{#1}%
10431       {\csgdef{eled@sectioning@#3#5}{\subsubsection{#2}}}%
10432       {\csgdef{eled@sectioning@#3#5}{\subsubsection{#1}{#2}}}%
10433     }%
10434     {\ifstrempy{#1}%
10435       {\csgdef{eled@sectioning@#3#5}{\subsubsection*{#2}}}%
10436       {\csgdef{eled@sectioning@#3#5}{\subsubsection*{#1}{#2}}}%Bug in
LaTeX!
10437     }
10438     \listcsgadd{eled@sections#5@@}{#3}%
10439   }
10440
10441 %

```

End of the conditional test about noeledsec option.

```

10442 }{}
10443 %

```

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.

```

10444 \def\normal@page@break{}
10445 %

```

\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.

```

10446 \def\l@prev@pb{}
10447 \def\l@prev@nopb{}
10448 %

```

\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.

```

10449 \newcommand{\ledpb}{\write\linenum@out{\string\led@pb}}
10450 \newcommand{\ledpbnum}[1]{\write\linenum@out{\string\led@pbnum{#1}}}
10451 \newcommand{\lednopb}{\write\linenum@out{\string\led@nopb}}
10452 \newcommand{\lednopbnum}[1]{\write\linenum@out{\string\led@nopbnum{#1}}}
10453 %

```

\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.

```

10454 \newcommand{\led@pb}{\listxadd{\l@prev@pb}{\the\absline@num}}
10455 \newcommand{\led@pbnum}[1]{\listxadd{\l@prev@pb}{#1}}
10456 \newcommand{\led@nopb}{\listxadd{\l@prev@nopb}{\the\absline@num}}
10457 \newcommand{\led@nopbnum}[1]{\listxadd{\l@prev@nopb}{#1}}
10458 %

```

\ledpbsetting The **\ledpbsetting** macro only changes the value of **\led@pb@macro**, for which the default value is before.

\led@pb@setting

```

10459 \def\led@pb@setting{before}
10460 \newcommand{\ledpbsetting}[1]{\gdef\led@pb@setting{#1}}
10461 %

```

\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**.

\led@check@nopb

```

10462 \newcommand{\led@check@pb}{\xifinlist{\the\absline@num}{\l@prev@pb}{\pagebreak[4]}}
10463 \newcommand{\led@check@nopb}{%
10464   \IfStrEq{\led@pb@setting}{before}{%
10465     \xifinlist{\the\absline@num}{\l@prev@nopb}%
10466     {\numdef{\abs@prevline}{\the\absline@num-1}%
10467     \xifinlist{\abs@prevline}{\normal@page@break}%
10468     {\nopagebreak[4]\enlargethispage{\baselineskip}}}%
10469     {}}%
10470   }%
10471 }%
10472 }%
10473 \IfStrEq{\led@pb@setting}{after}{%

```



```

10474 \xifinlist{\the\absline@num}{\l@prev@nopb}{%
10475 \xifinlist{\the\absline@num}{\normal@page@break}%
10476 {\nopagebreak[4]\enlargethispage{\baselineskip}}%
10477 {}%
10478 }%
10479 {}%
10480 {}%
10481 {}%
10482 }
10483 %

```

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.

```

10484 \newcommand{\check@pb@in@verse}{%
10485 \ifinstanza\iflednopbinverse\ifinserthangingsymbol% Using stanzas and
enabling page breaks in verse control, while on a hanging verse.
10486 \ifnum\page@num=\last@page@num\else%If we have change page
10487 \IfStrEq{\led@pb@setting}{before}{%
10488 \numgdef{\abs@line@verse}{\the\absline@num-1}%
10489 \ledpbnum{\abs@line@verse}%
10490 }{}%
10491 \IfStrEq{\led@pb@setting}{after}{%
10492 \numgdef{\abs@line@verse}{\the\absline@num-1}%
10493 \lednopbnum{\abs@line@verse}%
10494 }{}%
10495 \fi%
10496 \fi\fi\fi%
10497 }
10498 %

```

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

However, this command disrupts the line breaking mechanism when it is called before any word. This is why eledmac defines `\Hy@raisedlink@left` that is called to the left of words, at the beginning of `\edtext` or inside the `\edlabel` commands.³⁹

```

10499 \def\Hy@raisedlink@left#1{%
10500     \ifvmode
10501         #1%
10502     \else
10503         \Hy@SaveSpaceFactor
10504         \llap{\smash{%
10505             \begingroup
10506                 \let\HyperRaiseLinkLength\@tempdima
10507                 \setlength\HyperRaiseLinkLength\HyperRaiseLinkDefault
10508                 \HyperRaiseLinkHook
10509             \expandafter\endgroup
10510             \expandafter\raise\the\HyperRaiseLinkLength\hbox{%
10511                 \Hy@RestoreSpaceFactor
10512                 #1%
10513                 \Hy@SaveSpaceFactor
10514             }%
10515         }}%
10516         \Hy@RestoreSpaceFactor
10517         \penalty\@M\hskip\z@\relax
10518     \fi
10519 }
10520 %

```

XXXVIII Compatibility with eledmac

Here, we define some commands for the eledmac-compat option.

```

10521 \ifeledmaccompat%
10522
10523     \newcommand{\footnormalX}[1]{\arrangementX[#1]{normal}}%
10524     \newcommand{\footparagraphX}[1]{\arrangementX[#1]{paragraph}}%
10525     \newcommand{\foottwocolX}[1]{\arrangementX[#1]{twocol}}%
10526     \newcommand{\footthreecolX}[1]{\XarrangementX[#1]{threecol}}%
10527
10528     \unless\ifnocritical@
10529         \newcommand{\footnormal}[1]{\Xarrangement[#1]{normal}}%
10530         \newcommand{\footparagraph}[1]{\Xarrangement[#1]{paragraph}}%

```

³⁸<http://tex.stackexchange.com/a/17138/7712>.

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-coupe-de-ligne/801>.

```

10531 \newcommand{\foottwocol}[1]{\Xarrangement[#1]{twocol}}%
10532 \newcommand{\footthreecol}[1]{\Xarrangement[#1]{threecol}}%
10533 \let\hsizetwocol\Xhsizetwocol
10534 \let\hsizethreecol\Xhsizethreecol
10535 \let\bhookXnote\Xbhooknote
10536 \let\boxsymlinenum\Xboxsymlinenum
10537 \let\symlinenum\Xsymlinenum
10538 \let\beforenumberinfootnote\Xbeforenumber
10539 \let\afternumberinfootnote\Xafternumber
10540 \let\beforeXsymlinenum\XbeforeXsymlinenum
10541 \let\afterXsymlinenum\XafterXsymlinenum
10542 \let\inplaceofnumber\Xinplaceofnumber
10543 \let\Xlemmaseparator\lemmaseparator
10544 \let\afterlemmaseparator\Xafterlemmaseparator
10545 \let\beforelemmaseparator\Xbeforelemmaseparator
10546 \let\inplaceoflemmaseparator\Xinplaceoflemmaseparator
10547 \let\txbeforeXnotes\Xtxbeforenotes
10548 \let\afterXrule\Xafterrule
10549 \let\numberonlyfirstinline\Xnumberonlyfirstinline
10550 \let\numberonlyfirstintwoline\Xnumberonlyfirstintwoline
10551 \let\nonumberinfootnote\Xnonumberinfootnote
10552 \let\pstartinfootnote\Xpstart
10553 \let\pstartinfootnoteeverytime\Xpstarteverytime
10554 \let\onlyXpstart\Xonlypstart
10555 \let\Xnonumberinfootnote\Xnonumber
10556 \let\nonbreakableafternumber\Xnonbreakableafternumber
10557 \let\maxhXnotes\Xmaxhnotes
10558 \let\beforeXnotes\Xbeforenotes
10559 \let\boxlinenum\Xboxlinenum
10560 \let\boxlinenumalign\Xboxlinenumalign
10561 \let\boxstartlinenum\Xboxstartlinenum
10562 \let\boxendlinenum\Xboxendlinenum
10563 \let\twoline\Xtwoline
10564 \let\morethantwoline\Xmorethantwoline
10565 \let\twolinebutnotmore\Xtwolinebutnotmore
10566 \let\twolineonlyinsamepage\Xtwolineonlyinsamepage
10567 \fi
10568
10569 \unless\ifnofamiliar@
10570 \let\notesXwidthliketwocolumns\noteswidthliketwocolumnsX
10571 \fi
10572 \newcommandx{\parafootsep}[2][1,usedefault]{%
10573 \Xparafootsep[#1]{#2}%
10574 \parafootsepX[#1]{#2}
10575 }%
10576
10577 \newcommandx{\afternote}[2][1,usedefault]{%
10578 \Xafternote[#1]{#2}%
10579 \afternoteX[#1]{#2}%
10580 }%

```

```

10581 \unless\ifnoend@
10582   \let\XendXtwolines\Xendtwolines
10583   \let\XendXmorethantwolines\Xendmorethantwolines
10584   \let\hookXendnote\Xendhooknote
10585   \let\boxXendlinenum\Xendboxlinenum%
10586   \let\boxXendlinenumalign\Xendboxlinenumalign%
10587   \let\boxXendstartlinenum\Xendboxstartlinenum%
10588   \let\boxXendendlinenum\Xendboxendlinenum%
10589   \let\XendXlemmaseparator\Xendlemmaseparator
10590   \let\XendXbeforelemmaseparator\Xendbeforelemmaseparator
10591   \let\XendXafterlemmaseparator\Xendafterlemmaseparator
10592   \let\XendXinplaceoflemmaseparator\Xendinplaceoflemmaseparator
10593 \fi
10594
10595 \AtBeginDocument{%
10596   \ifdef\lineref{}\let\lineref\edlineref}%
10597 }%
10598
10599
10600 \fi%
10601 %
10602 %

```

</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
```

⁴⁰A name like `\text` is likely to be defined by other \TeX packages (it certainly is by the AMS packages) and it seems sensible to try and avoid clashes with other definitions.

```
\let<=\critext
```

Then you might say `<{Smith}\variant{Jones}/`. This of course destroys the ability to use `<` in any new macro definitions, so long as it remains in effect; hence it should be used with care.

Changing the character at the end of the command requires more work:

```
\catcode'\<=\active
\def\xtext#1#2>{\critext{#1}{#2}/}
\let<=\xtext
```

This allows you to say `<{Smith}\Afootnote{Jones}>`.

Aliases for `\critext` of the first kind shown here also can't be nested—that is, you can't use the alias in the text that forms the first argument to `\critext`. (See VI p. 140 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. 42), and remove any `\Xfootnote` command.

If you have customized the `\XXXXXfmt` command, please check whether you can achieve the same by the commands documented for display options (7 p. 43) or `\Xfootnote` options (6.2.2 p. 30). 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 `\XXXXXfmt` command, but you must replace:

```
\renewcommand*{XXXXfmt}[3]
```

with

```
\renewcommandx*{XXXXfmt}[4][4=Z]
```

⁴¹<https://github.com/maieul/ledmac/issues>

If you do not make that, you will get a spurious [X], where X is series letter.

If you used a `\protect` command inside a `\footnote` command inside a numbered section, you must change the `\protect` to `\noexpand`. Otherwise the command after the `\protect` will be discarded.

A.3 Migration to *eledmac* 1.5.1

The version 1.5.1 corrects a bug in `stanzaindentsrepetition` (cf. 9.3 p. 59). This bug had two consequences:

1. `stanzaindentsrepetition` did not work when its value was greater than 2.
2. `stanzaindentsrepetition` worked wrong when its value was equal to 2.

So, if you used `stanzaindentsrepetition` with a value equal to 2, you had to change your `\setstanzaindents`. Explanation:

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

This code, in versions prior to 1.5.1, made the first line have an indentation of 0, the second line of 1, the third verse of 0, the fourth verse of 1 and so forth.

But this code should have instead achieved quite the contrary: the first line would have an indentation of 1, the second line of 0, the third line of 1, the fourth line of 0 and so forth.

So version 1.5.1 corrected this bug. If you want to keep the former presentation, you must change:

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

to:

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

A.4 Migration to *eledmac* 1.12.0

The migration to *eledmac* 1.12.0 is easy:

- You must first delete all the auxiliary files, then compile your document three times as usual.
- If you have modified `\l@reg`, which is not advisable, you must rename it to `\@nl@reg`.

There is an additional problem. If you have put text into brackets just after `\pstart` or `\pend`, this text will be considered to be an optional argument of `\pstart` or `\pend` (see 5.2.3 p. 19). If so, add a `\relax` between `\pstart`/`\pend` and the first bracket.

The version 1.12.0 also introduce a better way to handle sectional divisions inside numbered text. Please read 16.2 p. 77.

A.5 Migration to eledmac 17.1

This version changes the default setting of `\Xpstart`. Henceforth, `pstart` numbers will be printed in footnotes within the section of text where you have called `\numberpstarttrue`.

We do not see any reason to print them in the other sections. However, if you want to print the `\pstart` numbers in every footnote, whatever the section, without having to use `\numberpstarttrue`, you can use `\Xpstarteverytime`.

A.6 Migration to eledmac 1.21.0

A.6.1 `\Xledsetnormalparstuff` and `\ledsetnormalparstuffX`

The `\ledsetnormalparstuff` has been split into two different commands:

- `\Xledsetnormalparstuff` for critical notes;
- `\ledsetnormalparstuffX` for familiar notes.

Both commands can take an optional argument which is the series letter. If you have redefined `\ledsetnormalparstuff` or any of the commands which call them, you must change them accordingly.

A.6.2 Endnotes

In any case, delete the `.end` file before the next run.

The previous version of Eledmac had a bug: there were two spaces between the starting page number and the starting line number, but only one space between the ending page number and the ending line number.

As a matter of fact, a spurious space was added after the first `\printnnum`. This spurious space has been deleted. However, if you want to keep the previous spurious space, you may load the package with the `oldprintnnumspace` option.

If you have redefined `\endprint`, you must:

- Contact us and ask for the feature that required your hack, in order to avoid such a hack in the future.
- Use the new fifth argument.
- Add `\xdef\@currentseries{#4}` at the beginning of your own command.

A.7 Migration to eledmac 1.22.0

The `\ledinnote` command now takes a first optional argument, which is the label for the hyperreference. If you have redefined it, change your redefinition, and check whether you can avoid this redefinition by only redefining `\ledinnotemark`.

A.8 Migration to eledmac 1.23.0

You must delete the numbered auxiliary files before compiling with the new version of eledmac.

A.9 Migration from eledmac to reledmac

There are many changes in reledmac which require the user to make modifications.

A.9.1 Risk of ‘no room for a new’

The risk to obtain a ‘no room for a new something’ error is greater in reledmac than it is in eledmac. See 19.1.3 p. 80 in order to know how to limit it.

A.9.2 Multiple indices with memoir

Eledmac and ledmac used the specific indexing tools of the memoir class designed to produce multiple indices. However, eledmac could also use imakeidx or indextools tools independently of the memoir class. This system forced to maintain redundant code. Since reledmac, we use only the imakeidx or indextools tools.

Consequently: Users of memoir are invited to use indextool or imakeidx to produce multiple indices.

A.9.3 Deprecated commands and options

The table of deprecated commands and their alternatives follows. Note that the way some commands must be used may have changed. Please read the handbook.

<i>Deprecated command</i>	<i>Replaced with</i>
<code>\addfootins</code>	<code>\newseries</code>
<code>\addfootinsX</code>	<code>\newseries</code>
<code>\critext</code>	<code>\edtext</code>
<code>\falseverse</code>	<code>\newverse</code>
<code>\interparanoteglue</code>	<code>\Xafternote</code> and <code>\afternoteX</code>
<code>\ledchapter</code>	<code>\eledchapter</code>
<code>\ledsection</code>	<code>\eledsection</code>
<code>\ledsetnormalparstuff</code>	<code>\Xledsetnormalparstuff</code> and <code>\ledsetnormalparstuffX</code>
<code>\ledsubsection</code>	<code>\eledsubsection</code>
<code>\ledsubsubsection</code>	<code>\eledsubsubsection</code>
<code>\noeledsec</code>	Package option <code>noeledsec</code>
<code>\noendnotes</code>	Package option <code>noendnotes</code>
<code>\pageparbreak</code>	<code>\ledpb</code>

The `ledsecnolinenumber` option has been removed, because it was related to deprecated commands.

The `oldprintnpnumspace` option has been removed too, because it was related to a historical bug. The `\usingedtext` and `\usingcritext` commands are also deprecated.

A.9.4 `\renewcommand` replaced by command

Many uses of `\renewcommand` have been replaced with uses of specific commands. Please read handbook about specific commands.

<i>Deprecated <code>\renewcommand</code></i>	<i>Replaced with</i>
<code>\@led@extranofeet</code>	<code>\newseries</code>
<code>\apprefprefixmore</code>	<code>\setapprefprefixmore</code>
<code>\apprefprefixsingle</code>	<code>\setapprefprefixsingle</code>
<code>\endstanzaextra</code>	Optional argument of <code>\&</code>
<code>\hangingsymbol</code>	<code>\sethangingsymbol</code>
<code>\ledfootinsdim</code>	<code>\Xmaxhnotes</code> and <code>\maxhnotesX</code>
<code>\parafootftmsep</code>	<code>\Xparafootsep</code> and <code>\parafootsepX</code>
<code>\notenumfont</code>	<code>\Xnotenumfont</code> , <code>\Xendnotenumfont</code> and <code>\notenumfontX</code>
<code>\notefontsetup</code>	<code>\Xnotefontsize</code> , <code>\Xendnotefontsize</code> and <code>\notefontsizeX</code>
<code>\sidenotesep</code>	<code>\setsidenotsep</code>
<code>\startstanzahook</code>	Optional argument of <code>\stanza</code>
<code>\symplinenum</code>	<code>\Xsymplinenum</code>

A.9.5 Commands the names of which have been changed

In order to help the migration from `eledmac` to `reledmac`, you may load `reledmac` with `eledmac-compat` option. However, it is advised not to, and to change the command names themselves instead. In many cases, you use only a few of them, except the `\footparagraph` command.

<i>Old command</i>	<i>New command</i>
<code>\footparagraph</code>	<code>\Xarrangement</code>
<code>\footnormal</code>	<code>\Xarrangement</code>
<code>\foottwocol</code>	<code>\Xarrangement</code>
<code>\footthreecol</code>	<code>\Xarrangement</code>
<code>\footparagraphX</code>	<code>\arrangementX</code>
<code>\footnormalX</code>	<code>\arrangementX</code>
<code>\foottwocolX</code>	<code>\arrangementX</code>
<code>\footthreecolX</code>	<code>\arrangementX</code>
<code>\afterlemmaseparator</code>	<code>\Xafterlemmaseparator</code>
<code>\afternote</code>	<code>\Xafternote</code> and <code>\afternoteX</code>
<code>\afternumberinfootnote</code>	<code>\Xafternumber</code>
<code>\afterXrule</code>	<code>\Xafterrule</code>
<code>\afterXsymplinenum</code>	<code>\Xaftersymplinenum</code>
<code>\beforelemmaseparator</code>	<code>\Xbeforelemmaseparator</code>
<code>\beforenumberinfootnote</code>	<code>\Xbeforenumber</code>
<code>\beforeXnotes</code>	<code>\Xbeforenotes</code>
<code>\beforeXsymplinenum</code>	<code>\Xbeforesymplinenum</code>

<i>Old command</i>	<i>New command</i>
<code>\bhookXnote</code>	<code>\Xbhookendnote</code>
<code>\bhookXnote</code>	<code>\Xbhooknote</code>
<code>\boxendlinenum</code>	<code>\Xboxendlinenum</code>
<code>\boxlinenum</code>	<code>\Xboxlinenum</code>
<code>\boxlinenumalign</code>	<code>\Xboxlinenumalign</code>
<code>\boxstartlinenum</code>	<code>\Xboxstartlinenum</code>
<code>\boxsymlinenum</code>	<code>\Xboxsymlinenum</code>
<code>\boxXendlinenum</code>	<code>\Xendboxlinenum</code>
<code>\boxXendlinenumalign</code>	<code>\Xendboxlinenumalign</code>
<code>\boxXendstartlinenum</code>	<code>\boxXendstartlinenum</code>
<code>\letboxXendendlinenum</code>	<code>\Xendletboxendlinenum</code>
<code>\hsizetwocol</code>	<code>\Xhsizetwocol</code>
<code>\hsizethreecol</code>	<code>\Xhsizethreecol</code>
<code>\inplaceoflemmaseparator</code>	<code>\Xinplaceoflemmaseparator</code>
<code>\inplaceofnumber</code>	<code>\Xinplaceofnumber</code>
<code>\lemmaseparator</code>	<code>\Xlemmaseparator</code>
<code>\maxhXnotes</code>	<code>\Xmaxhnotes</code>
<code>\morethantwolines</code>	<code>\Xmorethantwolines</code>
<code>\nonumberinfootnote</code>	<code>\Xnonumber</code>
<code>\notesXwidthliketwocolumns</code>	<code>\noteswidthliketwocolumnsX</code>
<code>\noXlemmaseparator</code>	<code>\Xnolemmaseparator</code>
<code>\numberonlyfirstinline</code>	<code>\Xnumberonlyfirstinline</code>
<code>\numberonlyfirstintwolines</code>	<code>\Xnumberonlyfirstintwolines</code>
<code>\nonbreakableafternumber</code>	<code>\Xnonbreakableafternumber</code>
<code>\onlyXpstart</code>	<code>\Xonlypstart</code>
<code>\parafootsep</code>	<code>\Xparafootsep</code> and <code>\parafootsepX</code>
<code>\pstartinfootnote</code>	<code>\Xpstart</code>
<code>\pstartinfootnoteeverytime</code>	<code>\Xpstarteverytime</code>
<code>\symlinenum</code>	<code>\Xsymlinenum</code>
<code>\twolines</code>	<code>\Xtwolines</code>
<code>\twolinesbutnotmore</code>	<code>\Xtwolinesbutnotmore</code>
<code>\twolinesonlyinsamepage</code>	<code>\Xtwolinesonlyinsamepage</code>
<code>\txtbeforeXnotes</code>	<code>\Xtxtbeforenotes</code>
<code>\XendXafterlemmaseparator</code>	<code>\Xendafterlemmaseparator</code>
<code>\XendXbeforelemmaseparator</code>	<code>\Xendbeforelemmaseparator</code>
<code>\XendXinplaceoflemmaseparator</code>	<code>\Xendinplaceoflemmaseparator</code>
<code>\XendXlemmaseparator</code>	<code>\Xendlemmaseparator</code>
<code>\XendXmorethantwolines</code>	<code>\Xendmorethantwolines</code>
<code>\XendXtwolines</code>	<code>\Xendtwolines</code>
<code>\Xnonumberinfootnote</code>	<code>\Xnonumber</code>
<code>\lineref</code>	<code>\edlineref</code>

A.9.6 Endnotes

With `reledmac`, there is now one auxiliary file for each endnotes set (`.Aend`, `.Bend`, `.Cend` etc.). If you have overridden `\doendnotes` (which you should not have done) you must adapt your code.

A.9.7 Z Series

The ‘Z’ series of notes has been removed. Only five series are provided now by default: A, B, C, D, E.

A.9.8 Internal commands

Users who have overridden internal commands, which is wrong, must adapt according to the following. Or better, they should not override any of such commands and use `reledmac` options instead.

- If you have modified `\Xfootfmt`, note that the fourth argument is now mandatory.
- `\unvxh` has been replaced with `\Xunvxh` and `\unvxhX` with two mandatory arguments.

A.10 Migration to `reledmac 2.1.0`

`Reledmac 2.1.0` fix some bugs when using `\Xbhooknote` and `\bhooknoteX` not in order to execute code at the beginning of each notes, but to insert content of at the beginning of each notes.

People who use these commands to do it, which is not the original idea, must change the following:

1. Horizontal space is no longer automatically added after the content of the `\Xbhooknote/\bhooknoteX` argument. You must include it manually. So instead of `\Xbhooknote{content}`, use `\Xbhooknote{content }.`
2. Indent is no longer automatically added before the content of the `\Xbhooknote/\bhooknoteX` argument. If you want to keep it, add `\indent` to the argument of `\Xbhooknote/\bhooknoteX`.

A.11 Migration to `reledmac 2.1.3`

`Reledmac 2.1.3` fix an historical bug, (style in `ledmac 0.7!`) which doubled the space before the rules of paragraphed familiar footnotes. Consequently, if you use paragraphed familiar footnotes, you should maybe adapt it, playing with `\beforenotesX`.

A.12 Migration to `reledmac 2.3.0`

Before `reledmac 2.3.0`, for typesetting verse, any empty line was considered a paragraph inside verses. Counting empty lines this created breaking verse, hanging verses, and also added spurious vertical spaces. Version 2.3.0 disables paragraph in stanza. If you want vertical space, use the optional argument of `\stanza` or `\endverse`.

A.13 Migration to reledmac 2.4.0

It is not mandatory, but strongly recommended, to change any `\renewcommand{\endashchar}{\langle...\rangle}` to the use of `\Xlinangeseparator` or `/` and `\Xendlinangeseparator` (7.4.4 p. 46).

A.14 Migration to reledmac 2.5.0

It is strongly recommended to stop redefining `\printnpnum` and to use the hooks documented in 7.4.3 p. 46.

`\xlineref` does not print anymore the side flag (R for right side), because it is incompatible with numerical test. Use `\xflagref` to obtain it.

The `\printlines` and `\printendlines` commands take now an eighth argument, which is the side flag. It is strongly recommended to NEVER redefine these two commands and to use the setting commands instead (or to ask for new setting commands if the actual does not answer to your needs). However, if you have done it, just change your redefinition to have a new argument.

It is strongly recommended to stop redefining `\fullstop` and to use `\Xsublinesep` instead.

A.15 Migration to reledmac 2.7.0

`\SErefonlypage` (introduced in reledmac 2.5.0) added an parenthesis after the page number. This was just an error, linked to a bad imitation of `\SErefwithpage`. That has been deleted. And so, the `\XendafterpagenumberSErefonlypage` to set it was also deleted.

`\rigidbalance` is split to two new commands: `\Xrigidbalance` for critical footnotes and `\rigidbalanceX` for familiar footnotes. If you have redefined it — but why should you have ? —, you should split your single redefinition in two redefinitions.

A.16 Migration to reledmac 2.7.2

`\Xhsize` is already defined in the `floatrow` package. It becomes `\Xwidth`, and, consequently, `\hsizeX` becomes `\widthX`.

The ancient names are temporarily maintained as aliases.

A.17 Migration to reledmac 2.8.0

Reledmac 2.8.0 fix spurious indents for paragraphed critical and familiar footnotes in `ledgroup` and `minipage`. You can re-establish the indent with `\Xparindent` and `\parindentX`.

A.18 Migration to reledmac 2.13.1

Reledmac 2.5.0 added a bug, which makes the right flag to be printed on the right side of critical footnotes, even if not explicitly requested by using `\Xlineflag`.

Version 2.13.1 solves this issue. Please use `\Xlineflag` if you want to add the right flag.

A.19 Migration to reledmac 2.18.0

After updating reledmac, and before any new compilation, you need to clean your `.aux` files, if you use `\edlabel` or related.

A.20 Migration to reledmac 2.21.0

Previously, there was a bug, which meant that the description in the handbook was incorrect. If you wrote

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

“elephant” was indexed in the main text and in the critical footnotes. With the new version of reledmac, it is indexed only in main text. If you also want to index it in critical footnotes, do

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

A.21 Migration to reledmac 2.24.0

When using `\labelpstarttrue`, a spurious space^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. 34) 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>	58
<code>\@EDROWFILL@</code>	1
<code>\@adv</code>	1
<code>\@advancestanzaanumber</code>	1
<code>\@annot</code>	1
<code>\@annot@end@print</code>	1
<code>\@annot@start@print</code>	1
<code>\@beforeinsertofthisedtext</code>	1
<code>\@check@edtext@args</code>	1
<code>\@doclearpage</code>	1
<code>\@doreinfeetX</code>	1

<code>\@edindex@fornote@</code>	1
<code>\@edindex@hyperref</code>	1
<code>\@edrowfill@</code>	1
<code>\@edtext@level</code>	1
<code>\@emptytoks</code>	1
<code>\@fnpos</code>	1
<code>\@footnotemark</code>	1
<code>\@footnotetext</code>	1
<code>\@getfirstseries</code>	1
<code>\@gobblefour</code>	1
<code>\@gobbleseven</code>	1
<code>\@gobblethree</code>	1
<code>\@h</code>	1
<code>\@hangingsymbol</code>	1
<code>\@iiiminipage</code>	1
<code>\@insertstanzanumber</code>	1
<code>\@insidethisline</code>	1
<code>\@k</code>	1
<code>\@l@dttempcnta</code>	1
<code>\@l@dttempcntb</code>	1
<code>\@lab</code>	1
<code>\@led@testifnofoot</code>	1
<code>\@lemma</code>	1
<code>\@line@@num</code>	1
<code>\@linenumannotationsep</code>	1
<code>\@lock</code>	1
<code>\@lopL</code>	1
<code>\@lopR</code>	1
<code>\@makechapterhead</code>	1
<code>\@makeschapterhead</code>	1
<code>\@mem@extranofeet</code>	1
<code>\@mem@old@ssect</code>	1
<code>\@mpfnpos</code>	1
<code>\@msd</code>	1
<code>\@msd@c</code>	1
<code>\@msd@options@iffullpage</code>	1
<code>\@msdata@list</code>	1
<code>\@nl</code>	1
<code>\@nl@reg</code>	1
<code>\@opXfeet</code>	1
<code>\@outputpage</code>	1
<code>\@page</code>	1
<code>\@pend</code>	1
<code>\@pendR</code>	1
<code>\@ref</code>	1
<code>\@ref@reg</code>	1
<code>\@ref@reg@parse</code>	1
<code>\@resetannot</code>	1
<code>\@sect</code>	1
<code>\@series</code>	1

\@set	1
\@sidenotesepp	1
\@ssect	1
\@startstanza	1
\@stopmsd	1
\@stopstanza	1
\@sw	1
\@tag	1
\@wredindex	1
\@xloop	1
\@xympar	1
CLASSarticle	80
CLASSbook	80, 389
CLASSmemoir	229, 230, 297–300, 336, 389, 409, 477, 481
CLASSscrbook	481
COMMAND*footnote	81
COMMAND\...\@footnotemark...	233
COMMAND\...d@ta	169
COMMAND\<hook	
@<series	283
COMMAND\<hookname	
<pseudoserries	285, 286
COMMAND\<type	
footfmt	215
COMMAND\@@line	205
COMMAND\@MM	191, 478
COMMAND\@Rlineflag	338, 478
COMMAND\@SErefprefix	315
COMMAND\@SErefprefixmore	315
COMMAND\@add@	379
COMMAND\@adv	123
COMMAND\@annot	176
COMMAND\@annot@end@print	219
COMMAND\@annot@start@print	219, 224, 264
COMMAND\@apprefprefixmore	315
COMMAND\@apprefprefixsingle	315
COMMAND\@beforeinsertofthisedtext	147, 148
COMMAND\@bsphack	302
COMMAND\@doclearpage	298, 299, 471, 481
COMMAND\@doreinfeetX	481
COMMAND\@dprintingcolumns	478
COMMAND\@edindex@hyperref	338–340
COMMAND\@edtext@level	144
COMMAND\@esphack	302
COMMAND\@firstofone	38
COMMAND\@fnpos	251, 294
COMMAND\@footnotemark	230, 471, 481
COMMAND\@footnotetext	230, 231, 471
COMMAND\@gobble	28, 38, 142, 143, 176, 277
COMMAND\@gobblefive	480

COMMAND\@gobblefour	477
COMMAND\@gobbleseven	280
COMMAND\@gobblethree	470
COMMAND\@h	208
COMMAND\@hangingsymbol	344
COMMAND\@iiiminipage	327, 329, 470, 481
COMMAND\@iiiminpage	327
COMMAND\@l	475
COMMAND\@l@tempcnta	172, 177, 185
COMMAND\@l@tempcntb	177
COMMAND\@l@reg	475
COMMAND\@lab	119, 302, 305, 310, 470
COMMAND\@ldunboxmpfoot	330
COMMAND\@led@extranofeet	410
COMMAND\@ledinnote@command	334, 335
COMMAND\@lemma	147, 149
COMMAND\@lock	113, 345
COMMAND\@lopL	471
COMMAND\@lopR	471
COMMAND\@makecol	294, 297, 481
COMMAND\@mempnum	300
COMMAND\@mpfnpos	251
COMMAND\@msd	355, 356
COMMAND\@msd@c	355
COMMAND\@msd@options@iffullpage	362
COMMAND\@msdata@list	355, 357
COMMAND\@nl	119, 121, 122, 124, 134, 176, 305, 470, 471
COMMAND\@nl@reg	119, 407, 471, 475
COMMAND\@opXfeet	471
COMMAND\@opfeetX	481
COMMAND\@opxtrafeeti	481
COMMAND\@outputpage	300, 301
COMMAND\@page	119, 121, 305
COMMAND\@pend	471
COMMAND\@pendR	471
COMMAND\@ref	119, 129–131, 135, 136, 142
COMMAND\@ref@later	131, 137
COMMAND\@ref@reg	130, 176, 471
COMMAND\@ref@reg@parsearg	131
COMMAND\@reinserts	294–297, 481
COMMAND\@resetannot	176
COMMAND\@secondoftwo	82
COMMAND\@sect	392
COMMAND\@series	282
COMMAND\@set	124
COMMAND\@sidenotesep	326
COMMAND\@stopmsd	356
COMMAND\@sw	131, 151, 155, 156
COMMAND\@tag	143, 145, 149
COMMAND\@tempcnta	96

COMMAND\@tempcntb	96
COMMAND\@this@crossref@start	229
COMMAND\@toksa	104
COMMAND\@toksb	104
COMMAND\@xloop	186, 187
COMMAND\@xympar	319, 481
COMMAND\Aendnote	17, 31
COMMAND\Afootfmt	191
COMMAND\Afootgroup	191
COMMAND\Afootnote	9, 17, 29–31, 33, 145, 200, 229, 252, 272, 480
COMMAND\Afootstart	191
COMMAND\AtBeginDocument	297
COMMAND\AtEndEveryPend	20, 487
COMMAND\AtEveryPend	19, 20, 62, 162, 477, 479, 481, 487
COMMAND\AtEveryPend*	20
COMMAND\AtEveryPstart	19, 20, 62, 477, 479, 481, 484, 487
COMMAND\AtEveryPstart*	20
COMMAND\AtEveryStanza	61, 485, 487
COMMAND\AtEveryStopStanza	62, 485, 487
COMMAND\AtStartEveryPstart	20, 487
COMMAND\AtStartEveryStanza	61, 487
COMMAND\BeforeEveryStopStanza	62, 487
COMMAND\Bendnote	17, 29
COMMAND\Bfootnote	9, 17, 229, 252, 272
COMMAND\Centering	54
COMMAND\Cfootnote	229
COMMAND\Columns	97, 158, 196
COMMAND\Dfootnote	229
COMMAND\Efootnote	229
COMMAND\Gls	73
COMMAND\Hy@raisedlink	402
COMMAND\Hy@raisedlink@left	402
COMMAND\LTR	54
COMMAND\NR	392
COMMAND\Pages	97, 158, 295, 296, 299
COMMAND\ProcessOptionsX	86
COMMAND\RL	52
COMMAND\RaggedLeft	54
COMMAND\RaggedRight	54
COMMAND\RenewExpandableDocumentCommand	39, 153
COMMAND\SEonlypage	312, 483
COMMAND\SEref	66–69, 312, 315, 484, 486, 490
COMMAND\SErefonlypage	66–69, 413, 483
COMMAND\SErefwithpage	66, 68, 69, 312, 315, 413, 483, 485, 491
COMMAND\Stanza	476
COMMAND\Waklam	380
COMMAND\X@doreinfeet	296, 481
COMMAND\XR@prefix	319
COMMAND\XR@test	319
COMMAND\XR@test@mac	319

COMMAND\XR@test@mac@test	319
COMMAND\XXXXXXfmt	406
COMMAND\XXXXXXfmt	406
COMMAND\Xafterenumber	49
COMMAND\Xafterlemmaseparator	51, 410
COMMAND\Xafternote	55, 409, 410
COMMAND\Xafternumber	49, 410
COMMAND\Xafterrule	56, 253, 410, 476, 479
COMMAND\Xaftersymlinenum	49, 410
COMMAND\Xarrangement	44, 55, 81, 192, 193, 284, 410
COMMAND\Xarrangement@footparagraph	198
COMMAND\Xarrangement@normal	193
COMMAND\Xarrangement@paragraph	198
COMMAND\Xbeforeinserting	54
COMMAND\Xbeforelemmaseparator	51, 410
COMMAND\Xbeforenotes	56, 252, 410, 476, 479
COMMAND\Xbeforenumber	49, 410
COMMAND\Xbeforesymlinenum	49, 410
COMMAND\Xbhookendnote	411
COMMAND\Xbhookgroup	56, 483, 484
COMMAND\Xbhooknote	54, 411, 412, 481, 482
COMMAND\Xboxendlinenum	50, 411, 480
COMMAND\Xboxlinenum	50, 411
COMMAND\Xboxlinenumalign	50, 411, 480
COMMAND\Xboxstartlinenum	50, 411, 480
COMMAND\Xboxsymlinenum	50, 411
COMMAND\Xcolalign	54, 479
COMMAND\Xdo@feet	481, 486
COMMAND\Xemdnoidenticallinenumannotation	68
COMMAND\Xend	280
COMMAND\XendXafterlemmaseparator	411
COMMAND\XendXbeforelemmaseparator	411
COMMAND\XendXinplaceoflemmaseparator	411
COMMAND\XendXlemmaseparator	411
COMMAND\XendXmorethantwolines	411
COMMAND\XendXtwolines	411
COMMAND\Xendafterenumber	49, 482
COMMAND\Xendafterlemmaseparator	51, 411
COMMAND\Xendafternote	57, 484
COMMAND\Xendafternumber	53
COMMAND\Xendafterpagenumbe	485
COMMAND\Xendafterpagenumber	46, 68
COMMAND\XendafterpagenumberSErefonlypage	413
COMMAND\Xendaftersymlinenum	49, 53, 482
COMMAND\Xendahookinplaceofnumber	53, 482
COMMAND\Xendahooklinenum	53, 482
COMMAND\Xendbeforelemmaseparator	51, 411
COMMAND\Xendbeforelinenum	53
COMMAND\Xendbeforenumber	49, 482
COMMAND\Xendbeforepagenumber	46, 67, 68

COMMAND\XendbeforepagenumberSErefonlypage	67
COMMAND\Xendbeforesymlinenum	49, 53, 482
COMMAND\Xendbhookinplaceofnumber	53, 482
COMMAND\Xendbhooklinenum	53, 482
COMMAND\Xendbhooknote	54
COMMAND\Xendboxendlinum	50, 480
COMMAND\Xendboxlinenum	50, 411, 478
COMMAND\Xendboxlinenumalign	50, 411, 480
COMMAND\Xendboxstartlinenum	50, 480
COMMAND\Xendboxsymlinenum	50, 482
COMMAND\Xendhangindent	53, 482, 484
COMMAND\Xendinplaceoflemmaseparator	31, 51, 411
COMMAND\Xendinplaceofnumber	49, 481
COMMAND\Xendinplaceofpagenumber	46, 486
COMMAND\Xendinsertsep@	259
COMMAND\Xendlemmadisablefontselection	52
COMMAND\Xendlemmafont	52, 483
COMMAND\Xendlemmaseparator	32, 51, 411
COMMAND\Xendletboxendlinum	411
COMMAND\Xendlineflag	68
COMMAND\Xendlinenumannotationonlyfirst	48, 264, 490
COMMAND\Xendlinenumannotationonlyfirstintwo	48, 264, 490
COMMAND\Xendlinenumannotationposition	27, 68
COMMAND\Xendlineprefixmore	46, 68
COMMAND\Xendlineprefixsingle	46, 68
COMMAND\Xendlinerangeseparator	46, 68, 213, 413, 482
COMMAND\Xendmorethantwolines	31, 47, 68, 411, 479, 480
COMMAND\Xendnoidenticallinenumannotation	28, 264, 489
COMMAND\Xendnolinenum	48, 491
COMMAND\Xendnolinenumberifannotation	48, 491
COMMAND\Xendnonumber	47, 481
COMMAND\Xendnote	255, 278–280, 479
COMMAND\Xendnotefontsize	52, 410
COMMAND\Xendnotenumfont	51, 53, 410
COMMAND\Xendnotes	259
COMMAND\Xendnumberonlyfirstinline	45, 482
COMMAND\Xendnumberonlyfirstintwolines	45, 482
COMMAND\Xendpagenumberonlyfirst	45, 486
COMMAND\Xendpagenumberonlyfirstifsingle	45, 486
COMMAND\Xendpagenumberonlyfirstintwo	46, 486
COMMAND\Xendparagraph	57, 476
COMMAND\Xendsep	57, 490
COMMAND\Xendstorelineinfo	257
COMMAND\Xendsublinesep	49, 68, 213
COMMAND\Xendsymlinenum	45, 482, 490
COMMAND\Xendsymlinenumannotation	48, 490
COMMAND\Xendsympagenum	46, 486
COMMAND\Xendtwolines	31, 47, 68, 411, 479, 480
COMMAND\Xendtwolinesbutnotmore	47, 68, 479, 480
COMMAND\Xendtwolinesonlyinsamepage	47, 68, 479, 480

COMMAND\Xendtxtbeforenotes	56, 488
COMMAND\Xendwrapcontent	53, 485
COMMAND\Xendwraplemma	52, 485
COMMAND\Xendwraplinenumannotation	28, 68
COMMAND\Xfootfmt	412
COMMAND\Xfootgroup	197
COMMAND\Xfootins	196
COMMAND\Xfootnote	65, 71, 143, 406, 473, 477–479, 483, 485
COMMAND\Xfootstarts	197
COMMAND\Xgroupbyline	55, 184, 228, 292, 415, 488–490
COMMAND\Xgroupbylines	486
COMMAND\Xgroupbylineseparetwolines	55
COMMAND\Xhangindent	53, 482, 490
COMMAND\Xhsize	413, 483, 484
COMMAND\Xhsizethreecol	54, 57, 411
COMMAND\Xhsizetwocol	54, 57, 285, 411
COMMAND\Xinnotemark	71, 415, 491
COMMAND\Xinplaceoflemmaseparator	31, 51, 411
COMMAND\Xinplaceofnumber	31, 49, 411, 478, 480
COMMAND\Xinsertparafootsep	203, 204
COMMAND\Xledsetnormalparstuff	408, 409, 479
COMMAND\Xlemmadisablefontselection	52
COMMAND\Xlemmafnt	52, 483
COMMAND\Xlemmaseparator	51, 213, 287, 289, 293, 411
COMMAND\Xlineflag	67, 413, 414, 485
COMMAND\Xlinenumannotationonlyfirst	48, 224, 490
COMMAND\Xlinenumannotationonlyfirstintwo	48, 224, 490
COMMAND\Xlinenumannotationposition	27, 68, 175
COMMAND\Xlinenumannotationposition@side	175
COMMAND\Xlinenumannotationpositionside	27
COMMAND\Xlinangeseparator	46, 68, 213, 413, 482
COMMAND\Xmaxhnotes	56, 80, 81, 410, 411, 476, 478
COMMAND\Xmorethantwolines	31, 46, 47, 68, 411, 478
COMMAND\Xnoendnum	48
COMMAND\Xnoidentcallinenumannotation	28, 68, 224, 489
COMMAND\Xnoindent	482
COMMAND\Xnolemmaseparator	51, 293, 411
COMMAND\Xnolinenumber	48, 219, 491
COMMAND\Xnolinenumberifannotation	48, 219, 491
COMMAND\Xnonbreakableafternumber	49, 411, 474
COMMAND\Xnonote	45, 491
COMMAND\Xnonumber	47, 48, 411
COMMAND\Xnonumberinfootnote	411
COMMAND\Xnotefontsize	52, 410
COMMAND\Xnotefontsize@⟨s⟩	203, 207, 208
COMMAND\Xnotenumfont	51, 410
COMMAND\Xnoteswidthliketwocolumns	57, 477
COMMAND\Xnumberonlyfirstinline	45, 46, 55, 116, 215, 257, 286, 287, 289, 411, 473, 478, 487
COMMAND\Xnumberonlyfirstintwolines	45, 55, 411, 473, 487
COMMAND\Xonlypstart	47, 411, 473, 478

COMMAND\Xpagelinesep	49, 486
COMMAND\Xparafootsep	55, 116, 410, 411, 415, 485, 487, 488
COMMAND\Xparafootsep@series	203
COMMAND\Xparindent	53, 413, 415, 479, 482, 484, 490
COMMAND\Xprenotes	56, 253, 485
COMMAND\Xprenotes@	196, 253, 473
COMMAND\Xprintlines	115
COMMAND\Xpstart	47, 408, 411, 473, 478
COMMAND\Xpstarteverytime	47, 408, 411, 478
COMMAND\Xpstartonlyfirst	47, 491
COMMAND\Xpstartseparator	47, 491
COMMAND\Xragged	55
COMMAND\Xrigidbalance	205, 413, 483
COMMAND\Xsethangindent	212
COMMAND\Xstanza	48, 62
COMMAND\Xstanzaonlyfirst	48, 491
COMMAND\Xstanzaseparator	48
COMMAND\Xstorelineinfo	215
COMMAND\Xsublinesep	23, 49, 68, 213, 413
COMMAND\Xsublinesepside	23, 49
COMMAND\Xsymlinenum	45, 55, 410, 411, 480, 487
COMMAND\Xsymlinenumannotation	48, 490
COMMAND\Xtextbeforenotes	187
COMMAND\Xtoendnotes	32, 279
COMMAND\Xtwolines	31, 46, 47, 68, 223, 225, 285, 411, 478
COMMAND\Xtwolinesappref	285
COMMAND\Xtwolinesbutnotmore	47, 68, 411, 479
COMMAND\Xtwolinesbutnotmoreappref	286
COMMAND\Xtwolinesonlyinsamepage	47, 68, 411, 479
COMMAND\Xtxtbeforenotes	55, 56, 411, 485, 486, 488
COMMAND\Xtxtbeforenotesonlyonce	56, 488
COMMAND\Xtxtbeforenumber	46
COMMAND\Xunvxh	200, 412
COMMAND\Xwidth	57, 413, 484
COMMAND\Xwrapcontent	53, 485
COMMAND\Xwraplemma	52, 54, 485
COMMAND\Xwraplinenumannotation	27, 28, 68, 175
COMMAND\Xwraplinenumannotation@ref	175
COMMAND\Xwraplinenumannotation@side	175
COMMAND\Xwraplinenumannotationref	28
COMMAND\Xwraplinenumannotationside	28
COMMAND\&	410
COMMAND\absline@num	112, 171
COMMAND\accent	142
COMMAND\actionlines@list	113, 172
COMMAND\actions@list	113
COMMAND\add@Xgroupbyline	184
COMMAND\add@hooktoggle@specific@to@cs	292
COMMAND\add@inserts	113, 183, 184
COMMAND\add@inserts@next	183, 184

COMMAND\add@msd@	355
COMMAND\add@msdata	355, 357
COMMAND\add@msdata@firstlineofpage	360
COMMAND\add@msddata	355
COMMAND\add@penalties	171, 185
COMMAND\addcontentsline	387
COMMAND\addfootins	406, 409
COMMAND\addfootinsX	406, 409
COMMAND\addtoendnotes	279
COMMAND\advancelabel@refs	304
COMMAND\advanceline	24, 115, 123, 139, 481
COMMAND\affixline@num	177, 180, 182, 326, 471
COMMAND\affixpstart@num	182
COMMAND\afterXrule	410
COMMAND\afterXsymlinenum	410
COMMAND\afterantilabe	63
COMMAND\aftergroup	142, 146
COMMAND\afterlemmaseparator	410
COMMAND\afternote	410
COMMAND\afternoteX	55, 409, 410
COMMAND\afternumberinfootnote	410
COMMAND\afterruleX	56, 476, 479
COMMAND\annot@end@print	224, 264
COMMAND\annot@list	115, 177
COMMAND\annotationref	27, 28, 64
COMMAND\antilabe	16, 63, 491
COMMAND\applabel	67, 306, 307, 479, 485
COMMAND\appref	65, 67–69, 312, 315, 483, 484, 490
COMMAND\apprefprefixmore	67, 410
COMMAND\apprefprefixsingle	67, 410
COMMAND\apprefwithpage	67–69, 312, 315, 480, 483, 491
COMMAND\arrangementX	44, 81, 233, 284, 410
COMMAND\arrangementX@normal	239
COMMAND\article	16
COMMAND\at@every@pend	162
COMMAND\autopar	19, 159, 163, 164, 249, 472, 474, 475, 479, 489
COMMAND\ballast	81
COMMAND\ballast@count	171, 185
COMMAND\baselineskip	44, 199, 203
COMMAND\beforeXnotes	410
COMMAND\beforeXsymlinenum	410
COMMAND\beforeantilabe	63
COMMAND\beforeeledchapter	11, 78, 388
COMMAND\beforeinsertingX	54
COMMAND\beforelemmaseparator	410
COMMAND\beforenotesX	56, 412, 475, 476, 479
COMMAND\beforenumberinfootnote	410
COMMAND\begin	364
COMMAND\beginnumbering	18–21, 97, 98, 102, 103, 112, 116, 117, 133, 163, 255, 354, 473, 476, 480, 481, 486, 487

COMMAND\beginnumberingR	97
COMMAND\bf	473
COMMAND\bfseries	51, 52, 473
COMMAND\bhookXnote	411
COMMAND\bhookgroupX	56, 483
COMMAND\bhooknoteX	54, 412, 481, 482
COMMAND\body	345
COMMAND\bodyfootmarkA	41
COMMAND\book	16
COMMAND\boxXendlinenum	411
COMMAND\boxXendlinenumalign	411
COMMAND\boxXendstartlinenum	411
COMMAND\boxendlinenum	411
COMMAND\boxlinefootnote	217
COMMAND\boxlinenum	411
COMMAND\boxlinenumalign	411
COMMAND\boxstartlinenum	411
COMMAND\boxsymlinenum	411
COMMAND\break	44, 200
COMMAND\brokenpenalty	185
COMMAND\c@page	134
COMMAND\c@page@page	134
COMMAND\centering	54
COMMAND\ch@ck@l@ck	471
COMMAND\ch@cksub@l@ck	180, 471
COMMAND\chapter	77, 389, 393, 476, 479, 481, 485
COMMAND\chaptermark	387
COMMAND\check@pb@in@verse	401
COMMAND\colalignX	54, 479
COMMAND\collect@body	365
COMMAND\color	485
COMMAND\colorbox	82
COMMAND\columns	57
COMMAND\columnwidth	199, 477
COMMAND\command names	285, 286
COMMAND\continuousnumberingwithcolumns@sync@linenum@singletext	174
COMMAND\copyright	143
COMMAND\correct@Xfootins@box	478
COMMAND\correct@footinsX@box	478
COMMAND\count	206
COMMAND\critex	472
COMMAND\critext	149, 405, 406, 409
COMMAND\csize	86
COMMAND\csquotes	276
COMMAND\ctab	381, 385
COMMAND\ctabtext	386
COMMAND\current@annot	176
COMMAND\dccl	374
COMMAND\def	83, 100
COMMAND\dimen	206

COMMAND\dimexpr	57
COMMAND\discretionary	200
COMMAND\displaywidowpenalty	185
COMMAND\do@Xfeet	294, 471, 481, 486
COMMAND\do@actions	171, 173, 471
COMMAND\do@actions@fixedcode	471
COMMAND\do@actions@next	171, 172
COMMAND\do@ballast	171, 185
COMMAND\do@feet@custom@order	294
COMMAND\do@insidelinehook	474
COMMAND\do@line	113, 141, 161, 165, 168, 170, 183, 185, 345, 471, 472, 474, 476
COMMAND\do@linehook	471
COMMAND\do@lockoff	114
COMMAND\do@lockon	114
COMMAND\dodoreintrafeet	470
COMMAND\doendnotes	31, 259, 260, 412, 480, 487
COMMAND\doendnotesbysection	31, 260, 280, 480, 487
COMMAND\doennotes	487
COMMAND\doinsidelinehook	28, 476
COMMAND\doinsidethislinehook	29, 169, 489
COMMAND\dolinehook	28, 476
COMMAND\doreintrafeeti	481
COMMAND\doreintrafeetii	481
COMMAND\doxtrafeet	470
COMMAND\doxtrafeeti	481
COMMAND\doxtrafeetii	481
COMMAND\dummy@ref	142
COMMAND\edaftertab	76, 380, 381
COMMAND\edannotationref	309
COMMAND\edatleft	76, 378
COMMAND\edatright	76, 379
COMMAND\edbeforetab	76, 380, 381
COMMAND\edfilldimen	379
COMMAND\edfont@info	149
COMMAND\edgls	73, 333
COMMAND\edgls...	484
COMMAND\edglsadd	487
COMMAND\edindex	39, 71, 73, 333, 334, 337, 339, 340, 369, 474, 476, 478, 481, 482, 486–488, 490
COMMAND\edindexlab	73
COMMAND\edlabel	64– 67, 69, 143, 302, 304, 305, 308, 310, 318, 333, 369, 402, 414, 470, 473–475, 478, 483, 488
COMMAND\edlabelE	66, 307
COMMAND\edlabelS	66, 307
COMMAND\edlabelSE	66
COMMAND\edlineref	64, 302, 411, 478, 480, 483, 487
COMMAND\edmakelabel	66, 318
COMMAND\edpageref	64, 302, 308, 318
COMMAND\edrowfill	380
COMMAND\edsublineref	64
COMMAND\edtabcolsep	373

COMMAND\edtext	7, 25, 26, 30, 33, 35, 36, 38, 58, 64, 65, 67, 71, 74, 81, 101, 113, 115, 129–131, 135, 136, 140–149, 152, 153, 155–157, 174, 176, 177, 306, 307, 311, 344, 369, 370, 389, 402, 405, 406, 409, 470, 472, 474, 476–480, 486–490
COMMAND\edtext@level	480
COMMAND\edtextlater	132
COMMAND\edvertdots	77, 379
COMMAND\edvertline	76, 77, 379
COMMAND\elechapter	78
COMMAND\eled@sectioning@out	394
COMMAND\eledchapter	77, 409, 476, 481
COMMAND\eledchapter*	77
COMMAND\eledmac@error	470
COMMAND\eledsection	7, 17, 77, 142, 167, 388, 409, 478, 487, 489, 490
COMMAND\eledsection*	77
COMMAND\eledsubsection	77, 409
COMMAND\eledsubsection*	77
COMMAND\eledsubsubsection	77, 409
COMMAND\eledsubsubsection*	77
COMMAND\eledxxx	11, 78, 394, 476
COMMAND\eledxxxx	387
COMMAND\elesection	394
COMMAND\else	332, 388
COMMAND\emph	38
COMMAND\empty	96, 178, 302
COMMAND\end	364
COMMAND\end@lemmas	141, 142
COMMAND\endashchar	213
COMMAND\endgraf	161, 202, 249
COMMAND\endlock	24, 114, 139, 350
COMMAND\endminipage	327, 329, 470, 481
COMMAND\endmsdata	40
COMMAND\endnotes	479, 483
COMMAND\endnumbering	18, 19, 21, 98, 100, 101, 133, 471, 480, 486, 487
COMMAND\endnumberingR	97
COMMAND\endprint	255, 258, 280, 408
COMMAND\endstanzaextra	410
COMMAND\endsub	23, 114, 138
COMMAND\endverse	412
COMMAND\everypar	163
COMMAND\extensionchars	79, 98
COMMAND\externaldocument	69, 318
COMMAND\fx@x@l@cks	471
COMMAND\falseverse	409, 474, 476
COMMAND\fi	388
COMMAND\firstlinenum	22, 177, 472
COMMAND\firstsublinenum	22, 472
COMMAND\flag@end	135, 136, 148, 476
COMMAND\flag@end@RTL	137
COMMAND\flag@end@later	137
COMMAND\flag@start	135, 136, 148, 476, 477

COMMAND\flag@start@RTL	137
COMMAND\flag@start@later	137
COMMAND\flagstanza	63
COMMAND\floatingpenalty	191, 478
COMMAND\flush@notes	186
COMMAND\fnpos	43, 251, 475, 486
COMMAND\footfmt	191, 193
COMMAND\footfmt...	234
COMMAND\footfootmarkA	41
COMMAND\footfudgefactor	201
COMMAND\footfudgefiddle	81, 198, 199, 470
COMMAND\footgroup	191
COMMAND\footins	196
COMMAND\footnormal	284, 410, 470
COMMAND\footnormalX	410
COMMAND\footnote	41, 80, 229–231, 407, 471
COMMAND\footnote@lang	213
COMMAND\footnoteA	17, 41
COMMAND\footnoteB	17
COMMAND\footnoteC	29
COMMAND\footnoteE	41
COMMAND\footnoteX	9, 42, 276, 277, 486
COMMAND\footnoteX@reading	277
COMMAND\footnoteXmark	278, 487, 490
COMMAND\footnoteXmk	293
COMMAND\footnoteXtext	278, 487, 490
COMMAND\footnote⟨X⟩	143
COMMAND\footnote⟨X⟩mark	42
COMMAND\footnote⟨X⟩mk	42
COMMAND\footnote⟨X⟩nomk	42
COMMAND\footnote⟨X⟩text	42
COMMAND\footnotelang@lua	190
COMMAND\footnotelang@poly	190
COMMAND\footnotemark	42, 278
COMMAND\footnoteoption@	189, 482
COMMAND\footnoterule	206
COMMAND\footnotesize	52
COMMAND\footnotetext	42, 278
COMMAND\footparagraph	199, 284, 410, 476
COMMAND\footparagraphX	244, 410, 476
COMMAND\footsplitskips	471, 478
COMMAND\footstart	191, 196, 206
COMMAND\footstrut	202
COMMAND\footthreecol	410
COMMAND\footthreecolX	410, 479
COMMAND\foottwocol	410
COMMAND\foottwocolX	410, 479
COMMAND\foreignlanguage	53
COMMAND\fullstop	413
COMMAND\get@edindex@hyperref	338

COMMAND\get@edindex@ledinnote@command	334
COMMAND\get@fnmark	231
COMMAND\get@index@command	475
COMMAND\get@linelistfile	471
COMMAND\get@thisfootnote	237
COMMAND\getline@num	170, 172
COMMAND\gl@p	104
COMMAND\global	119
COMMAND\globaldefs	119
COMMAND\gls	73, 341
COMMAND\hangindentX	53, 479, 482, 490
COMMAND\hangingsymbol	410, 472
COMMAND\hbox	200
COMMAND\hfill	474
COMMAND\hidenumbering	25, 128, 414, 479, 489, 491
COMMAND\hidenumberingonleftpage	25, 128, 485
COMMAND\hidenumberingonrightpage	25, 128, 485
COMMAND\hline	74
COMMAND\hooktoggle@specific	292
COMMAND\hrulefill	380
COMMAND\hsize	44, 196, 199, 201, 207, 210, 250, 471, 477
COMMAND\hsizeX	413, 483, 484
COMMAND\hsizethreecol	411
COMMAND\hsizethreecolX	55, 57
COMMAND\hsizetwocol	411
COMMAND\hsizetwocolX	54, 57
COMMAND\hyperlinkR	338
COMMAND\hyperlinkformat	338
COMMAND\hyperlinkformatR	338
COMMAND\if@RTL	88, 146
COMMAND\if@edtext@	477, 480
COMMAND\if@eled@sectioning	394
COMMAND\if@firstlineofpage	88
COMMAND\if@firstlineofpageR	88
COMMAND\if@msd@options@fullpage	362
COMMAND\if@msdata@insertedfrompreviouspage	359
COMMAND\if@nobreak	162
COMMAND\if@noneed@Footnote	135
COMMAND\ifXnote@	97
COMMAND\ifafterendnumberingR	97
COMMAND\ifautopar	164
COMMAND\ifby@autopar	164
COMMAND\ifbypage@	104
COMMAND\ifbypage@R	105
COMMAND\ifbypstart@	104
COMMAND\ifbypstart@R	105
COMMAND\iffirst@linenum@out@	132, 133
COMMAND\ifindtl@innote	97
COMMAND\ifindtl@notenumber	97
COMMAND\ifinserthangingsymbol	345

COMMAND\ifinstanza	345
COMMAND\ifistwofollowinglines	225
COMMAND\ifl@d@Xlinenumber	219
COMMAND\ifl@d@Xmorethantwolines	219, 478
COMMAND\ifl@d@Xtwolines	219
COMMAND\ifl@d@dash	219
COMMAND\ifl@d@elin	219
COMMAND\ifl@d@esl	219
COMMAND\ifl@d@pnum	219
COMMAND\ifl@d@ssub	219
COMMAND\ifl@dend@X	278
COMMAND\ifl@dmemoir	470
COMMAND\ifl@dpaging	477
COMMAND\ifl@dpairing	96, 472
COMMAND\ifl@dprintingpages	478
COMMAND\ifl@dskipnumber	177
COMMAND\ifl@dstartendok	379
COMMAND\ifl@imakeidx	88
COMMAND\ifledRcol	97, 472
COMMAND\ifledRcol@	97, 476
COMMAND\iflemmacommand@	477
COMMAND\iflinenumannotationotherside	175
COMMAND\ifnoend@	260
COMMAND\ifnoledgroup@	332
COMMAND\ifnoteschanged@	115
COMMAND\ifnumberedpar@	158
COMMAND\ifnumbering	98, 101
COMMAND\ifnumberingR	97, 472
COMMAND\ifnumberline	148, 177
COMMAND\ifpst@rted	472
COMMAND\ifpst@rtedL	99
COMMAND\ifresumenumbering@start	102
COMMAND\ifseriesbefore	283
COMMAND\ifstopmsdata@inserted@	354
COMMAND\ifsublines@	112, 125
COMMAND\ifsublines@R	112
COMMAND\iftrue	480
COMMAND\ifvmode	304
COMMAND\ifxxx	388
COMMAND\ignorespaces	145
COMMAND\imki@wrindexentry	88
COMMAND\immediate	132, 133, 254
COMMAND\indent	19, 163, 412
COMMAND\index	341, 486
COMMAND\indtl@wrindexentry	88
COMMAND\initnumbering@quote	386, 481
COMMAND\initnumbering@reg	471
COMMAND\initnumbering@sectcmd	481
COMMAND\innotemarkX	71, 415, 491
COMMAND\inplaceofflemmaseparator	411

COMMAND\inplaceofnumber	411
COMMAND\insert	148, 183, 191, 193, 194, 208, 228, 234
COMMAND\insert@Xtxtbeforenotes	187, 208
COMMAND\insert@count	129, 135, 145
COMMAND\insert@countR	145
COMMAND\insert@msdata	355, 362
COMMAND\insert@txtbeforenotesX	187
COMMAND\inserthangingsymbol	474
COMMAND\insertlines@list	113, 129, 130
COMMAND\insertparafootsepX	248
COMMAND\inserts@list	141, 159, 183, 200
COMMAND\interAfootnotelinepenalty	472
COMMAND\interfootnotelinepenalty	472
COMMAND\interlinepenalty	191
COMMAND\interparanoteglue	409
COMMAND\justifying	54
COMMAND\keep@this@crossref@forinserting	229
COMMAND\l@advance@parledgroup@beforenormalnotes	481
COMMAND\l@d@@wrindexhyp	477
COMMAND\l@d@add	150
COMMAND\l@d@end	255, 278
COMMAND\l@d@nums	145, 148, 150, 219, 220
COMMAND\l@d@section	255
COMMAND\l@d@set	124, 139
COMMAND\l@dampcount	371
COMMAND\l@dbfnote	231, 471
COMMAND\l@dcheckstartend	379
COMMAND\l@dchset@num	124
COMMAND\l@dcolcount	371, 372
COMMAND\l@dcollect@@body	364
COMMAND\l@dcollect@body	364
COMMAND\l@dcsnote	476
COMMAND\l@dcsnotetext	169, 323
COMMAND\l@dcsnotetext@l	169, 323
COMMAND\l@dcsnotetext@r	169, 323
COMMAND\l@ddodoreinextrafeet	295, 470
COMMAND\l@ddoxtrafeet	294, 295, 470
COMMAND\l@demptyd@ta	471
COMMAND\l@dend@close	254
COMMAND\l@dend@open	254
COMMAND\l@dend@stuff	255
COMMAND\l@denvbody	364
COMMAND\l@dfeetbeginmini	470
COMMAND\l@dfeetendmini	470
COMMAND\l@dgetline@margin	472
COMMAND\l@dgetlock@disp	472
COMMAND\l@dgetref@num	310
COMMAND\l@dgetsidenote@margin	320, 472
COMMAND\l@dgobbeloptarg	477
COMMAND\l@dgobblearg	477

COMMAND\l@dobbleoptarg	369
COMMAND\l@dlabel@parse	310
COMMAND\l@dld@ta	177, 179
COMMAND\l@dld@rbox	325
COMMAND\l@dlsn@te	472
COMMAND\l@dlsnote	476
COMMAND\l@dmake@labels	305, 319
COMMAND\l@dmake@labelsR	319
COMMAND\l@dnumpsstartsL	99, 472
COMMAND\l@dp@rsefootsspec	220
COMMAND\l@dp@rsefootsspec	220
COMMAND\l@dpush@begins	364
COMMAND\l@drd@ta	177, 179
COMMAND\l@dref@undefined	310
COMMAND\l@drrsn@te	472
COMMAND\l@drrsnote	476
COMMAND\l@dtabaddcols	379
COMMAND\l@dtabnoexpands	470
COMMAND\l@dumboxmpfoot	481
COMMAND\l@dunboxmpfoot	471
COMMAND\l@dunhbox@line	165
COMMAND\l@dzeropenalties	472, 477
COMMAND\l@noexpands@edgl	344
COMMAND\l@pb	400
COMMAND\l@prev@nopb	399
COMMAND\l@prev@pb	399
COMMAND\l@reg	407
COMMAND\l@secondoptarg	369
COMMAND\l@wrap@ifnotemptybox	95
COMMAND\label	20, 66, 69, 73, 302, 303, 311
COMMAND\label@refs	302
COMMAND\labelstarttrue	20, 414, 473, 488
COMMAND\labelref@list	302, 305
COMMAND\language	200
COMMAND\lastbox	164
COMMAND\lastskip	138
COMMAND\latex@makecol	297
COMMAND\leavevmode	19, 163
COMMAND\led@check@nopb	400
COMMAND\led@check@pb	400
COMMAND\led@nopb	400, 401
COMMAND\led@nopbnum	400
COMMAND\led@pb	400, 401
COMMAND\led@pb@macro	400
COMMAND\led@pbnum	400
COMMAND\led@reinit@index@fornote	341
COMMAND\led@set@index@fornote	341
COMMAND\ledRflag	338
COMMAND\ledchapter	409, 473
COMMAND\ledfootinsdim	410

COMMAND\ledinnernote	69, 321, 476, 485
COMMAND\ledinnote	335, 408, 415, 480
COMMAND\ledinnotemark	408, 415, 479, 491
COMMAND\ledleftnote	70, 321
COMMAND\ledlinenum	110, 472
COMMAND\ledllfill	170
COMMAND\ledlsnotefontsetup	485
COMMAND\ledlsnotesep	70
COMMAND\ledlsnotewidth	70
COMMAND\lednopb	79, 400
COMMAND\lednopbinverse	401
COMMAND\lednopbinversetrue	61, 79
COMMAND\lednopbnum	400
COMMAND\ledouternote	69, 321, 476, 485
COMMAND\ledpb	79, 400, 409
COMMAND\ledpbnum	400
COMMAND\ledpbsetting	79, 400, 482
COMMAND\ledrightnote	70, 321
COMMAND\ledrsnotefontsetup	485
COMMAND\ledrsnotesep	70
COMMAND\ledrsnotewidth	70
COMMAND\ledsection	409
COMMAND\ledsectnomark	387
COMMAND\ledsectnotoc	387
COMMAND\ledsetnormalparstuff	408, 409, 479
COMMAND\ledsetnormalparstuff@common	249
COMMAND\ledsetnormalparstuffX	408, 409, 479
COMMAND\ledsidenote	69, 70, 321, 323
COMMAND\ledsubsection	409
COMMAND\ledsubsubsection	409
COMMAND\ledxxx	476
COMMAND\left	76
COMMAND\leftctab	381
COMMAND\leftheadline	110
COMMAND\leftlinenum	23, 110, 470, 472
COMMAND\leftlinenumannotation	111
COMMAND\leftltab	380
COMMAND\leftnoteupfalse	70
COMMAND\leftpstartnum	182
COMMAND\leftftab	380
COMMAND\leftsidenote	323
COMMAND\leftskip	196, 199, 201
COMMAND\lemma	3, 30, 33, 35, 36, 38, 141, 144–146, 149, 152, 405, 472, 473, 480, 481, 483
COMMAND\lemmaseparator	411
COMMAND\let	34, 58, 143, 148, 277, 297, 350, 470
COMMAND\letboxXendendlinenum	411
COMMAND\line	205, 208
COMMAND\line@list	113, 115, 131, 148
COMMAND\line@list@stuff	98, 116, 117, 133, 470, 472
COMMAND\line@list@version	119

COMMAND\line@margin	106, 179, 320
COMMAND\line@num	112, 114, 177, 470
COMMAND\line@set	150
COMMAND\lineannot	27, 65, 151
COMMAND\lineannot@set	151
COMMAND\lineation	22, 105, 488–490
COMMAND\linebreak	44
COMMAND\lineneation	484
COMMAND\linenum	27, 30, 33, 65, 67, 141, 150, 151, 308, 311, 318, 405
COMMAND\linenum@out	132, 302, 305
COMMAND\linenumannotation	25, 174, 488, 489
COMMAND\linenumannotationothersidefalse	27
COMMAND\linenumannotationothersidettrue	27, 491
COMMAND\linenumberlist	22, 96, 178, 470
COMMAND\linenumberstyle	24, 109, 470, 487
COMMAND\linenumincrement	22, 472
COMMAND\linenummargin	23, 106, 320
COMMAND\linenumr@p	109, 470, 472
COMMAND\linenumrep	109, 472
COMMAND\linenumsep	23, 70, 110, 321
COMMAND\linerangesep@	293
COMMAND\lineref	302, 308, 318, 411, 478
COMMAND\list@clear	103
COMMAND\list@clearing@reg	472
COMMAND\list@create	103
COMMAND\lock@disp	108
COMMAND\lock@off	127
COMMAND\lock@on	126
COMMAND\lockdisp	24, 108
COMMAND\loop	186, 187, 345
COMMAND\ltab	380, 381, 385
COMMAND\ltabtext	386
COMMAND\m@mmf@prepare	230
COMMAND\mainmatter	488, 489
COMMAND\makeatletter	168
COMMAND\makehboxoffhboxes	201, 203
COMMAND\makeindex	71, 337
COMMAND\makelabel	318
COMMAND\manage stanza@modulo	346
COMMAND\marginpar	69, 80, 319, 320, 471
COMMAND\marginparwidth	70, 321
COMMAND\markboth	29, 168
COMMAND\mathchardef	346
COMMAND\maxhXnotes	411
COMMAND\maxhnotesX	57, 81, 410, 475, 476, 478, 479
COMMAND\maxlinesinpar@list	117
COMMAND\measurebody	382
COMMAND\measuretbody	383
COMMAND\memorybreak	21
COMMAND\metaXXXvfootnote	228

COMMAND\microtype	165
COMMAND\morenoexpands	81, 82, 141, 143
COMMAND\morethantwolines	411
COMMAND\mpfnpos	43, 251, 475, 486
COMMAND\mpnormalfootgroup	470
COMMAND\mpnormalvfootnote	471
COMMAND\msdata	39, 40, 353, 354, 356, 487, 489
COMMAND\msdataposition	41
COMMAND\multfootsep	41, 230
COMMAND\multiplefootnotemarker	230
COMMAND\musixtex	475
COMMAND\n@num	472, 479
COMMAND\n@num@ref	479
COMMAND\new@line	134, 471
COMMAND\new@series	143
COMMAND\newcommand	34, 83, 100, 229, 305
COMMAND\newcommandx	34
COMMAND\newhookarg@specific	292
COMMAND\newhookcommand@series	285, 286, 479, 491
COMMAND\newhookcommand@series@reload	286
COMMAND\newhookcommand@toggle@reload	286, 477
COMMAND\newhooktoggle@series	286, 479
COMMAND\newhooktoggle@specific	291
COMMAND\newif	479
COMMAND\newline	44
COMMAND\newlinechar	279
COMMAND\newseries	42, 406, 409, 410
COMMAND\newseries@	269, 283
COMMAND\newverse	61, 63, 409, 476, 487
COMMAND\next	345
COMMAND\next@action	117
COMMAND\next@actionline	117
COMMAND\next@insert	183
COMMAND\next@line@list@stuff	133, 488
COMMAND\nl@regR	119
COMMAND\no@expands	81, 143, 149, 277, 344, 470
COMMAND\noXlemmaseparator	411
COMMAND\nobreak	219
COMMAND\nocritical	270
COMMAND\noeledsec	409
COMMAND\noendnotes	409
COMMAND\noexpand	407
COMMAND\nofamiliar	289
COMMAND\noindent	19, 20, 61, 62, 163, 482
COMMAND\noindentX	482
COMMAND\nomk@	293
COMMAND\nonbreakableafternumber	411
COMMAND\nonoteX	45, 491
COMMAND\nonumberinfootnote	411
COMMAND\norelax	58

COMMAND\normal@footnotemarkX	233
COMMAND\normal@page@break	399
COMMAND\normal@pars	249
COMMAND\normalbfnoteX	472
COMMAND\normalbodyfootmarkX	234
COMMAND\normalfont	486
COMMAND\normalfootfmt	58, 195, 202, 213, 255, 485
COMMAND\normalfootfmtX	235
COMMAND\normalfootfootmarkX	235
COMMAND\normalfootgroup	197
COMMAND\normalfootgroupX	236
COMMAND\normalfootnoterule	192
COMMAND\normalfootstart	196, 199
COMMAND\normalfootstartX	235
COMMAND\normalvfootnote	193, 195
COMMAND\normalvfootnote@inserted	194, 195
COMMAND\normalvfootnoteX	234
COMMAND\notbool	388
COMMAND\notfontsetup	410
COMMAND\notfontsizeX	52, 410
COMMAND\notenumfont	410
COMMAND\notenumfontX	52, 410
COMMAND\notesXwidthliketwocolumns	411
COMMAND\noteswidthliketwocolumnsX	57, 411, 477, 479
COMMAND\num@lines	158, 185
COMMAND\numberlinefalse	21, 489
COMMAND\numberlinetrue	21, 489
COMMAND\numberonlyfirstinline	283, 411
COMMAND\numberonlyfirstintwolines	411
COMMAND\numberpstartfalse	20
COMMAND\numberpstarttrue	20, 47, 408, 472, 481, 487, 489
COMMAND\numberstanza	48
COMMAND\numberstanzafalse	62
COMMAND\numberstanzatrue	62
COMMAND\numlabfont	23, 58, 110
COMMAND\none@line	158
COMMAND\nonehalfspacing	482
COMMAND\nonlyXpstart	411
COMMAND\nonlysideX	276
COMMAND\npage@action	114, 125
COMMAND\npage@start	114, 472
COMMAND\npagecontents	114
COMMAND\npagenumbering	300
COMMAND\npageparbreak	409
COMMAND\npageref	66, 308
COMMAND\npar	32, 44, 163, 164, 249
COMMAND\npar@line	158, 185
COMMAND\npar@patch@pagenumbering	300
COMMAND\npar@patch@thepage	134, 300
COMMAND\npara@footgroup	199

COMMAND\para@footgroupX	247
COMMAND\para@footsetup	199, 470
COMMAND\para@footsetupX	244, 470, 477
COMMAND\para@vfootnoteX	245
COMMAND\parafootfmt	201, 202, 485
COMMAND\parafootfmtX	246
COMMAND\parafootftm	204
COMMAND\parafootftmX	248
COMMAND\parafootftmsep	410
COMMAND\parafootsep	411, 475, 480
COMMAND\parafootsepX	55, 116, 410, 411, 415, 485, 490, 491
COMMAND\parafootstart	199
COMMAND\parafootstartX	245
COMMAND\paravfootnote	200, 203
COMMAND\parfillskip	202
COMMAND\parindent	482
COMMAND\parindentX	53, 413, 482, 484, 490
COMMAND\parse@annot	177
COMMAND\parshape	81
COMMAND\parskip	20, 61, 62, 164
COMMAND\pause@numbering	101
COMMAND\pausenumbering	21, 101, 102, 117, 118, 164, 475, 477, 484
COMMAND\pausenumbering@page@num	101
COMMAND\penalty	202
COMMAND\pend	2, 7, 18–21, 23, 24, 78, 139, 141, 144, 151, 158, 159, 161–165, 182, 183, 352, 407, 475, 476, 485–488
COMMAND\preXnotes	479, 485
COMMAND\prenotesX	56, 254, 479
COMMAND\prepare@Xgroupbyline	228, 229
COMMAND\prepare@Xprenotes	252
COMMAND\prev@nopb	400
COMMAND\prev@pb	400
COMMAND\prevlineX	116
COMMAND\prevpageX@num	116
COMMAND\print@Xfootnoterule	479
COMMAND\print@Xnotes	295, 296
COMMAND\print@Xnotes@forpages	478
COMMAND\print@eledsection	167
COMMAND\print@footnoteXrule	479
COMMAND\print@leftmargin@eledsection	388
COMMAND\print@lemma	214
COMMAND\print@line	165
COMMAND\print@notesX@forpages	478
COMMAND\print@rightmargin@eledsection	388
COMMAND\printendlines	260, 268, 316, 413, 470, 472
COMMAND\printlinefootnote	215, 216, 478
COMMAND\printlinefootnotearea	217, 219, 478
COMMAND\printlinefootnotenumbers	215
COMMAND\printlines	195, 213, 219, 220, 260, 315, 413, 470, 472, 478, 483
COMMAND\printnnum	408, 413

COMMAND\printpstart	214
COMMAND\protect	143, 407
COMMAND\providecommand	229, 470
COMMAND\pstarL	158
COMMAND\pstart	2, 7, 18–21, 23, 24, 77, 78, 124, 139, 144, 147, 151, 158, 159, 161–164, 167, 183, 352, 393, 407, 408, 472–474, 476, 477, 479–481, 484–491
COMMAND\pstartinfootnote	411
COMMAND\pstartinfootnoteeverytime	411
COMMAND\pstartnum	182
COMMAND\pstartref	64, 302, 309, 475, 487
COMMAND\pstarts	472, 473
COMMAND\pstarts@readL	158
COMMAND\pstarts@typeset@L	158
COMMAND\raggedX	55
COMMAND\raggedleft	54
COMMAND\raggedright	54
COMMAND\raw@text	158, 159
COMMAND\rbracket	51
COMMAND\read@linelist	116, 118
COMMAND\ref	66, 69, 73
COMMAND\reformatted@	315
COMMAND\reformattedwithpage	315
COMMAND\relax	20, 124, 148, 171, 183, 350, 369, 407
COMMAND\renewcommand	81, 410, 413
COMMAND\reset@current@annot	176
COMMAND\reset@msd@options@iffullpage	362
COMMAND\resetlinenumannotation	26, 176
COMMAND\resetprevline@	116
COMMAND\resetprevpage@	116
COMMAND\resumenumbering	21, 97, 98, 102, 103, 117, 118, 164, 472, 476, 477, 484
COMMAND\right	76
COMMAND\rightctab	381
COMMAND\rightlinenum	23, 110, 470, 472
COMMAND\rightlinenumannotation	111
COMMAND\rightltab	381
COMMAND\rightnoteupfalse	70
COMMAND\rightrtab	381
COMMAND\rightrightnote	323
COMMAND\rightrightskip	196, 199, 201, 202
COMMAND\rightstartnum	182
COMMAND\rigidbalance	205, 206, 208, 413, 483
COMMAND\rigidbalanceX	205, 413, 483
COMMAND\robustify	45, 48
COMMAND\roman	374, 483
COMMAND\rtab	380–382, 385
COMMAND\rtabtext	382, 386
COMMAND\sameparallelpagenum	134
COMMAND\sameword	17, 34–36, 38, 132, 151–154, 156, 415, 478, 480, 482, 485, 488, 490
COMMAND\sameword@inedtext	152, 153
COMMAND\saweword	152

COMMAND\scriptsize	110
COMMAND\section	77, 472, 486
COMMAND\section@num	98
COMMAND\sectionmark	387
COMMAND\select@lemmafont	58, 189
COMMAND\series	269
COMMAND\series@	269
COMMAND\seriesatbegin	42, 282, 479
COMMAND\seriesatend	42, 282, 479
COMMAND\set@Xtxtbeforenotes	187
COMMAND\set@continuousnumberingforl	103
COMMAND\set@line	148
COMMAND\set@line@action	114, 125
COMMAND\set@txtbeforenotesX	187
COMMAND\setSErefonlypageprefixmore	67, 315, 484, 487
COMMAND\setSErefonlypageprefixsingle	67, 315, 484, 487
COMMAND\setSErefprefixmore	67
COMMAND\setSErefprefixsingle	67
COMMAND\setapprefprefixmore	67, 410
COMMAND\setapprefprefixsingle	67, 69, 410, 483
COMMAND\setcommand@series	284
COMMAND\sethangindentX	212
COMMAND\sethangingsymbol	61, 344, 410, 482
COMMAND\sethanginsymbol	59
COMMAND\setistwofollowinglines	225
COMMAND\setl@dlprbox	325
COMMAND\setline	24, 115, 119, 124, 139, 143, 481
COMMAND\setlinenum	24, 119, 124, 139, 162, 470, 488
COMMAND\setlinenumannotationsep	28, 414, 489
COMMAND\setmsdatalabel	40
COMMAND\setmsdataposition	487
COMMAND\setmsdataseries	40
COMMAND\setprintendlines	260, 265, 472
COMMAND\setprintlines	220, 225, 260, 472
COMMAND\setsidenotesep	70
COMMAND\setsidenotsep	410
COMMAND\setstanzaindent	347
COMMAND\setstanzaindents	60, 346, 407, 491
COMMAND\setstanzapenalties	346
COMMAND\setstanzavalues	346
COMMAND\settoggle@series	283, 473, 477
COMMAND\showlemma	142, 470, 471
COMMAND\showwordrank	38, 153
COMMAND\sidenote@margin	471
COMMAND\sidenotemargin	69, 471, 476
COMMAND\sidenotesep	410
COMMAND\sidepstartnumtrue	20
COMMAND\skip	196
COMMAND\skipnumbering	25, 127, 128, 140, 472, 479, 480
COMMAND\skipnumbering@reg	479

COMMAND\small	52
COMMAND\somemacro	38
COMMAND\special	14
COMMAND\splitmaxdepth	191, 207
COMMAND\splitoff	205
COMMAND\splittopskip	191, 207, 208
COMMAND\stanza	24, 25, 61–63, 97, 350, 410, 412, 482, 487
COMMAND\stanza@hang	349
COMMAND\stanza@line	349
COMMAND\stanzaindent	60, 347, 478
COMMAND\stanzaindent*	60
COMMAND\stanzaindentbase	346
COMMAND\stanzanumwrapper	62
COMMAND\startlock	22, 24, 114, 139, 350
COMMAND\startstanzahook	410
COMMAND\startsub	23, 114, 138
COMMAND\stopmsd	356
COMMAND\stopmsdata	40, 354
COMMAND\store@annot@to@absline	176
COMMAND\strip@pt	199
COMMAND\strutbox	207
COMMAND\sub@action	114, 125
COMMAND\sub@lock	113
COMMAND\sub@off	122, 305
COMMAND\sub@on	122, 305
COMMAND\subline@num	112, 114
COMMAND\sublinenum@rep	470
COMMAND\sublinenumberstyle	24, 109, 470
COMMAND\sublinenumincrement	22
COMMAND\sublinenumr@p	109, 470, 472
COMMAND\sublinenumrep	109, 472
COMMAND\sublineref	64, 302, 309
COMMAND\subsectionmark	387
COMMAND\sw@inthisedtext	145
COMMAND\sw@list@inedtext	149, 157
COMMAND\sw@txt	153
COMMAND\swnoexpands	38, 488
COMMAND\symlinenum	411
COMMAND\symplinenum	410
COMMAND\sza@penalty	349
COMMAND>tag	478
COMMAND\text	405
COMMAND\text<language>	53
COMMAND\textcolor	82
COMMAND\textheight	81
COMMAND\textsuperscript	27, 95, 175
COMMAND\the	134, 470
COMMAND\thefootnoteA	41
COMMAND\thefootnoteX	474
COMMAND\thelabidx	339, 340

COMMAND\thepage	119, 134, 300
COMMAND\thepstart	20, 414
COMMAND\thepstartL	473
COMMAND\thepstartR	473
COMMAND\thestanza	62
COMMAND\this@crossref@end	229
COMMAND\this@line@list@version	133
COMMAND\this@section@next@page@num	101
COMMAND\this@section@page@num	115
COMMAND\thisfootnote	237
COMMAND\threecol@begin@insert	208
COMMAND\threecolfootfmt	207, 208, 485
COMMAND\threecolfootfmtX	243
COMMAND\threecolfootgroup	206
COMMAND\threecolfootgroupX	243
COMMAND\threecolfootsetup	206
COMMAND\threecolfootsetupX	242
COMMAND\threecolvfootnote	207
COMMAND\threecolvfootnote@inserted	207
COMMAND\threecolvfootnoteX	242
COMMAND\toendnotes	32, 259, 485
COMMAND\twocolfootfmt	485
COMMAND\twocolfootfmtX	240
COMMAND\twocolfootgroupX	241
COMMAND\twocolfootsetupX	240
COMMAND\twocolvfootnoteX	240
COMMAND\twolines	283, 411
COMMAND\twolines@A	283
COMMAND\twolines@B	283
COMMAND\twolines@C	283
COMMAND\twolinesbutnotmore	411
COMMAND\twolinesonlyinsamepage	411
COMMAND\txtbeforeXnotes	411
COMMAND\txtbeforenotesX	56, 187, 488
COMMAND\txtbeforenotesonlyonceX	56, 488
COMMAND\uline	42
COMMAND\unhbox	165, 200
COMMAND\unpenalty	201–203
COMMAND\unskip	202
COMMAND\unvxh	202, 412
COMMAND\unvxhX	412
COMMAND\upbracefill	380
COMMAND\usingcritext	406, 409
COMMAND\usingdtext	406, 409
COMMAND\vAfootnote	191
COMMAND\vadjust	136
COMMAND\variant	33
COMMAND\vbox	161, 163, 200, 205, 252
COMMAND\vfootnote	191, 196, 200, 207
COMMAND\vl@dbfnote	231, 471

COMMAND\vl@disnote	324
COMMAND\vl@dlsnote	324
COMMAND\vl@dosnote	324
COMMAND\vl@drsnote	324
COMMAND\vnumfootnoteX	472
COMMAND\vsizer	56, 57, 81
COMMAND\vsplit	185
COMMAND\waklam	380
COMMAND\waklamec	380
COMMAND\wapunktel	380
COMMAND\wastricht	380
COMMAND\widthX	57, 413, 484
COMMAND\wrap@edcrossref	308, 477
COMMAND\wrapcontentX	53, 485
COMMAND\wrapped@bodyfootmarkX	249
COMMAND\wrapped@footfootmarkX	248
COMMAND\x...	65
COMMAND\xabslineref	309
COMMAND\xannotationref	65
COMMAND\xdef	104, 350
COMMAND\xflagref	65, 310, 413, 483
COMMAND\xleft@appenditem	104, 142
COMMAND\xlineref	65, 413, 483
COMMAND\xpageref	65
COMMAND\xparse	39
COMMAND\xpstartref	65, 475, 488
COMMAND\xr	69
COMMAND\xright@appenditem	104
COMMAND\xsublineref	65
COMMAND\xxref	65, 311, 318, 475, 478, 485
COMMAND\xxxmatter	300
ENVIRONMENTastanza	98, 483
ENVIRONMENTedarrayc	385
ENVIRONMENTedarrayl	385
ENVIRONMENTedarrayr	385
ENVIRONMENTedtabular	490
ENVIRONMENTedtabularc	386
ENVIRONMENTedtabularl	386
ENVIRONMENTedtabularr	386
ENVIRONMENTledgroup	87, 330, 413, 483
ENVIRONMENTledgroupsize	331
PACKAGE(r)(e)ledmac	42
PACKAGEEledmac	114, 336, 408, 409, 478, 479
PACKAGEEledpar	479
PACKAGEEtoolbox	86
PACKAGEParallel	417
PACKAGEREledmac	62, 412, 413
PACKAGEamsgen	363
PACKAGEamsmath	363
PACKAGEbabel	53, 82, 374, 483

PACKAGEbiblatex	80
PACKAGEbidi	52, 54, 88, 482, 490
PACKAGEcalc	491
PACKAGEccaption	96
PACKAGEcolor	82
PACKAGEcsquotes	487
PACKAGEedmac	1, 6, 11, 13–15, 83, 219, 229, 302, 346, 405, 417, 470
PACKAGEedstanza	1, 14, 15, 344
PACKAGEeledmac	1, 11, 15, 17, 72, 229, 332, 336, 367, 389, 402, 406, 408–410, 474, 476, 478
PACKAGEeledpar	191, 417, 472, 476–478
PACKAGEetex	482
PACKAGEetoolbox	103, 151, 269, 283, 294, 323, 388, 399
PACKAGEfancyhdr	297, 486
PACKAGEfloatrow	80, 413
PACKAGEfootmisc	41, 82, 88, 229, 417, 486
PACKAGEgeometry	16
PACKAGEglossaries	73, 341, 483
PACKAGEhandout	477
PACKAGEhyperlink	277
PACKAGEhyperref	65, 144, 248, 249, 303, 338, 339, 392, 402, 475–477, 484, 487, 490, 491
PACKAGEifluatex	86
PACKAGEifxetex	86
PACKAGEimakeidx	71, 80, 87, 88, 332, 336, 409, 474–476, 478
PACKAGEindextols	341
PACKAGEindextool	409
PACKAGEindextools	71, 80, 87, 88, 97, 332, 336, 341, 409, 478, 483, 486
PACKAGEledarab	82
PACKAGEledmac	1, 11, 14, 15, 82, 103, 336, 405, 406, 409, 412
PACKAGEledpar	82
PACKAGEMemoir	87, 336, 409, 417, 476, 481
PACKAGEMorewrites	80
PACKAGEMusixtex	476
PACKAGEperpage	483
PACKAGEpolyglossia	51, 82, 190, 213, 483, 490
PACKAGERagged2e	54, 86
PACKAGERELEDmac	1, 2, 6, 11–13, 15–18, 21, 22, 25, 28, 29, 31, 33–35, 39, 41–43, 45, 46, 48, 50, 53, 54, 56, 58, 61, 63, 65–67, 69, 71–74, 78, 80–84, 105, 107, 113, 115, 118, 119, 122, 132–134, 143, 146, 148, 151, 183, 192, 196, 201, 213, 229, 255, 270, 274, 275, 283, 294, 299, 308, 309, 312, 318, 319, 336, 359, 363, 367, 387, 388, 401, 402, 409, 410, 412–415, 481, 484, 486
PACKAGERELEDpar	1, 4, 6, 9, 17, 20, 22, 42, 57, 65, 67, 68, 78–80, 82, 84, 96, 105, 117, 122, 133, 134, 145, 148, 158, 193, 196, 250, 251, 269, 276, 293, 295, 296, 299, 332, 344, 387, 482, 483, 485, 486, 488, 489
PACKAGESuffix	86
PACKAGETabmac	1, 14, 15, 417
PACKAGETHALIE	63
PACKAGEulem	42
PACKAGEuninormalize	35
PACKAGExargs	34, 86
PACKAGExkeyval	84, 293
PACKAGEXparse	86, 489

PACKAGEX	5, 69, 319, 484
PACKAGEXref	318
PACKAGEXstring	86, 339

A

\absline@num	1
Abu Kamil Shuja' b. Aslam	14
\actionlines@list	1
\actions@list	1
\add@hookarg@specific@to@cs	1
\add@hooktoggle@specific@to@cs	1
\add@inserts	1
\add@inserts@next	1
\add@msd@	1
\add@msdata	1
\add@msdata@firstlineofpage	1
\add@penalties	1
\add@Xgroupbyline	1
\addtol@denvbody	1
Adelard II	14
\advancelabel@refs	1
\advanceline	1, 24
\Aendnote	31
\affixline@num	1
\affixpstart@num	1
\affixside@note	1
\Afootnote	30
\afterantilabe	63
\afternoteX	55
\afterruleX	56
\ampersand	1, 63
\annot@list	1
\annotationref	1, 64
\antilabe	1, 62
\append@notesX	1
\append@Xnotes	1
\applabel	1, 67
\appref	1, 67
\apprefwithpage	1, 67
\arrangementX	1, 44
\arrangementX@normal	1
\arrangementX@paragraph	1
\arrangementX@threecol	1
\arrangementX@twocol	1
\at@every@pend	1
\AtEndEveryPend	1, 20
\AtEveryPend	1, 20
\AtEveryPstart	1, 20
\AtEveryStanza	1, 61
\AtEveryStopStanza	1, 62

\AtStartEveryPstart	1, 20
\AtStartEveryStanza	1, 61
\autopar	1, 19

B

\ballast	81
\ballast@count	1
Beeton, Barbara Ann Neuhaus Friend	19
\beforeantilabe	63
\beforeeiledchapter	1
\BeforeEveryStopStanza	1, 62
\beforeinsertingX	54
\beforeinsertion@X	1
\beforenotesX	56
\beginnumbering	1, 18
\Bendnote	31
\Bfootnote	30
\bhookgroupX	56
\bhooknoteX	54
\bodyfootmarkA	41
\boxfootnotenumbers	1
Bredon, Simon	14
Breger, Herbert	14, 367
Brey, Gerhard	14
Busard, Hubert L. L.	14
\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	31
\Cfootnote	30
\ch@ck@l@ck	1
\ch@cksub@l@ck	1
\chapter	1
\check@pb@in@verse	1
Chester, Robert of	14
Claassens, Geert H. M.	14
\colalignX	54
\continuousnumberingwithcolumns@sync@linenumber@singletext	1
Copernicus, Nicolaus	14
\critext	405

<code>\ctab</code>	1
<code>\ctabtext</code>	1
<code>\current@annot</code>	1

D

Dekker, Dirk-Jan	82
<code>\Dendnote</code>	31
<code>\Dfootnote</code>	30
<code>\disable@endnotes</code>	1
<code>\disable@familiarnotes</code>	1
<code>\disable@notes</code>	1
<code>\disable@sidenotes</code>	1
<code>\disablel@dtabfeet</code>	1
<code>\do@actions</code>	1
<code>\do@actions@fixedcode</code>	1
<code>\do@actions@next</code>	1
<code>\do@ballast</code>	1
<code>\do@feet@custom@order</code>	1
<code>\do@feetX</code>	1
<code>\do@insidelinehook</code>	1
<code>\do@line</code>	1
<code>\do@linehook</code>	1
<code>\do@lockoff</code>	1
<code>\do@lockoffL</code>	1
<code>\do@lockon</code>	1
<code>\do@lockonL</code>	1
<code>\do@Xfeet</code>	1
<code>\doedindexlabel</code>	1
<code>\doendnotes</code>	1, 31
<code>\doendnotesbysection</code>	1, 31
<code>\doinsidelinehook</code>	1, 28
<code>\doinsidethislinehook</code>	1, 29
<code>\dolinehook</code>	1, 28
<code>\dosplits</code>	1
Downes, Michael	81, 200, 202
<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, 76, 380
<code>edarrayc</code> (environment)	74
<code>edarrayl</code> (environment)	74
<code>edarrayr</code> (environment)	74
<code>\edatleft</code>	1, 76
<code>\edatright</code>	1, 76
<code>\edbeforetab</code>	1, 76, 380
<code>\edfilldimen</code>	1
<code>\edfont@info</code>	1

<code>\edindex</code>	1, 71
<code>\edindexlab</code>	1, 73
<code>\EDLABEL</code>	1
<code>\edlabel</code>	1, 64
<code>\edlabelE</code>	1, 66
<code>\edlabelS</code>	1, 66
<code>\edlabelSE</code>	1, 66
<code>\edlineref</code>	1, 64
<code>\edmakelabel</code>	1, 66
<code>\edpageref</code>	1, 64
<code>\edrowfill</code>	1, 75
<code>\EDTAB</code>	1
<code>\edtabcolsep</code>	1, 74
<code>\EDTABINDENT</code>	1
<code>\edtabindent</code>	1
<code>\EDTABtext</code>	1
<code>edtabularc (environment)</code>	74
<code>edtabularl (environment)</code>	74
<code>edtabularr (environment)</code>	74
<code>\EDTEXT</code>	1
<code>\edtext</code>	1, 30
<code>\edvertdots</code>	1, 76
<code>\edvertline</code>	1, 76
<code>\Eendnote</code>	31
<code>\Efootnote</code>	30
<code>\eled@chapter</code>	1
<code>\eled@section</code>	1
<code>\eled@sectioning@out</code>	1
<code>\eled@subsection</code>	1
<code>\eled@subsubsection</code>	1
<code>\eledchapter</code>	1
<code>\eledchapter*</code>	1
<code>\eledsection</code>	1
<code>\eledsection*</code>	1
<code>\eledsubsection</code>	1
<code>\eledsubsection*</code>	1
<code>\eledsubsubsection</code>	1
<code>\eledsubsubsection*</code>	1
<code>\enablel@dtabfeet</code>	1
<code>\end@lemmas</code>	1
<code>\endashchar</code>	1
<code>\endline@num</code>	1
<code>\endlock</code>	1, 24
<code>\endminipage</code>	1
<code>\endnumbering</code>	1, 18
<code>\endpage@num</code>	1
<code>\endprint</code>	1
<code>\endquotation</code>	1
<code>\endquote</code>	1
<code>\endsub</code>	1, 23

<code>\endsubline@num</code>	1
environments:	
<code>edarrayc</code>	74
<code>edarrayl</code>	74
<code>edarrayr</code>	74
<code>edtabularc</code>	74
<code>edtabularl</code>	74
<code>edtabularr</code>	74
<code>ledgroup</code>	63
<code>ledgroupsize</code>	64
<code>minipage</code>	63
Euclid	14
<code>\extensionchars</code>	1, 79

F

<code>\f@x@l@cks</code>	1
Fairbairns, Robin	41
<code>\first@linenum@out@false</code>	1
<code>\first@linenum@out@true</code>	1
<code>\firsthalf@antilabe@wd</code>	1
<code>\firstlinenum</code>	1, 22
<code>\firstseriesX@</code>	1
<code>\firstsublinenum</code>	1, 22
<code>\firstXseries@</code>	1
<code>\flag@end</code>	1
<code>\flag@end@later</code>	1
<code>\flag@end@RTL</code>	1
<code>\flag@start</code>	1
<code>\flag@start@later</code>	1
<code>\flag@start@RTL</code>	1
<code>\flagstanza</code>	1, 63
<code>\flush@notes</code>	1
<code>\fnpos</code>	1, 42
Folkerts, Menso	14
<code>\footfootmarkA</code>	41
<code>\footfudgefiddle</code>	1, 81
<code>\footnote</code>	1
<code>\footnoteA</code>	41
<code>\footnoteB</code>	41
<code>\footnoteC</code>	41
<code>\footnoteD</code>	41
<code>\footnoteE</code>	41
<code>\footnotelang@lua</code>	1
<code>\footnotelang@poly</code>	1
<code>\footnoteoptions@</code>	1
<code>\footnoteXmark</code>	42
<code>\footnoteXtext</code>	42
<code>\footsplitskips</code>	1
<code>\fullstop</code>	1

G

Gädeke, Nora	14
\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	53
\hidenumbering	1, 25
\hidenumberingonleftpage	1, 25
\hidenumberingonrightpage	1
\Hilfsbox	1
\hilfsbox	1
\hilfscount	1
\HILFSskip	1
\Hilfsskip	1
\hilfsskip	1
\hookarg@specific	1
\hooktoggle@specific	1
\hsizethreecolX	55
\hsizetwocolX	54
\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@Footnote	1

<code>\if@RTL</code>	1
<code>\ifafterendnumberingR</code>	1
<code>\ifantilabe@</code>	1
<code>\ifat@every@pend@star@</code>	1
<code>\ifat@every@pstart@star@</code>	1
<code>\ifautopar@pause</code>	1
<code>\ifbypage@</code>	1
<code>\ifbypage@R</code>	1
<code>\ifbypstart@</code>	1
<code>\ifbypstart@R</code>	1
<code>\ifeledmaccompat@</code>	1
<code>\iffirst@linenum@out@</code>	1
<code>\ifinastanzaL</code>	1
<code>\ifinastanzaR</code>	1
<code>\ifindtl@innote</code>	1
<code>\ifindtl@notenumber</code>	1
<code>\ifinserthangingsymbol</code>	1
<code>\ifinstanza</code>	1
<code>\ifinstanzaL</code>	1
<code>\ifinstanzaR</code>	1
<code>\ifl@d@dash</code>	1
<code>\ifl@d@elin</code>	1
<code>\ifl@d@esl</code>	1
<code>\ifl@d@pnum</code>	1
<code>\ifl@d@ssub</code>	1
<code>\ifl@d@Xendlinenumber</code>	1
<code>\ifl@d@Xlinenumber</code>	1
<code>\ifl@d@Xmorethantwolines</code>	1
<code>\ifl@d@Xtwolines</code>	1
<code>\ifl@dend@X</code>	1
<code>\ifl@dhidenumber</code>	1
<code>\ifl@dmemoir</code>	1
<code>\ifl@dpaging</code>	1
<code>\ifl@dpairing</code>	1
<code>\ifl@dprintingcolumns</code>	1
<code>\ifl@dprintingpages</code>	1
<code>\ifl@dskipnumber</code>	1
<code>\ifl@dskipversenumber</code>	1
<code>\ifl@dstartendok</code>	1
<code>\ifl@footmisc</code>	1
<code>\ifl@imakeidx</code>	1
<code>\ifl@indextools</code>	1
<code>\ifledfinal</code>	1, 79
<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>\ifparapparatus@</code>	1
<code>\ifparledgroup</code>	1
<code>\ifprevpgnotnumbered</code>	1
<code>\ifpst@rtedL</code>	1
<code>\ifresumenumbering@start</code>	1
<code>\ifsameparallelpagenumbers</code>	1
<code>\ifseriesbefore</code>	1
<code>\ifsidepstartnum</code>	1
<code>\ifstopmsdata@inserted@</code>	1
<code>\ifsublines@</code>	1
<code>\ifsublinesR</code>	1
<code>\ifwidthliketwocolumns</code>	1
<code>\ifXendinsertsep@</code>	1
<code>\ifxindy@</code>	1
<code>\ifxindyhyperref@</code>	1
<code>\initnumbering@quote</code>	1
<code>\initnumbering@reg</code>	1
<code>\innotemarkX</code>	71
<code>\insert@count</code>	0, 1
<code>\insert@msdata</code>	1
<code>\insert@txtbeforenotesX</code>	1
<code>\insert@Xtxtbeforenotes</code>	1
<code>\insertthangingsymbol</code>	1
<code>\insertlines@list</code>	1
<code>\insertparafootsepX</code>	1
<code>\inserts@list</code>	1

J

Jayaditya	14
-----------------	----

K

Kabelschacht, Alois	187
<code>\keep@this@crossref@forinserting</code>	1

L

<code>\l@advance@parledgroup@beforenormalnotes</code>	1
<code>\l@d@add</code>	1
<code>\l@d@nums</code>	1
<code>\l@d@section</code>	1

<code>\l@d@set</code>	1
<code>\l@d@Xend</code>	1
<code>\l@dampcount</code>	1
<code>\l@dbfnote</code>	1
<code>\l@dcheckcols</code>	1
<code>\l@dcheckstartend</code>	1
<code>\l@dchset@num</code>	1
<code>\l@dcolcount</code>	1
<code>\l@dcollect@body</code>	1
<code>\l@dcollect@body</code>	1
<code>\l@dcolwidth</code>	1
<code>\l@dcsnote</code>	1
<code>\l@dcsnotetext</code>	1
<code>\l@dcsnotetext@l</code>	1
<code>\l@dcsnotetext@r</code>	1
<code>\l@ddodoreinextrafeet</code>	1
<code>\l@ddoxtrafeet</code>	1
<code>\l@dedbeginmini</code>	1
<code>\l@dedendmini</code>	1
<code>\l@emptyd@ta</code>	1
<code>\l@dend@close</code>	1
<code>\l@dend@open</code>	1
<code>\l@dend@stuff</code>	1
<code>\l@dend@Xfalse</code>	1
<code>\l@dend@Xtrue</code>	1
<code>\l@denvbody</code>	1
<code>\l@desnote</code>	1
<code>\l@dfambeginmini</code>	1
<code>\l@dfamendmini</code>	1
<code>\l@dfeetbeginmini</code>	1
<code>\l@dfeetendmini</code>	1
<code>\l@dgetline@margin</code>	1
<code>\l@dgetlock@disp</code>	1
<code>\l@dgetref@num</code>	1
<code>\l@dgetsidenote@margin</code>	1
<code>\l@dobbleoptarg</code>	1
<code>\l@disnote</code>	1
<code>\l@dlabel@parse</code>	1
<code>\l@dld@ta</code>	1
<code>\l@dlp@rbox</code>	1
<code>\l@dlsn@te</code>	1
<code>\l@dlsnote</code>	1
<code>\l@dmake@labels</code>	1
<code>\l@dmodfordtext</code>	1
<code>\l@dnullfills</code>	1
<code>\l@dnumpstartsL</code>	1
<code>\l@doldold@footnotetext</code>	1
<code>\l@dp@rsefootspec</code>	1
<code>\l@dparsedendline</code>	1
<code>\l@dparsedendpage</code>	1

<code>\l@dparsedendsub</code>	1
<code>\l@dparsedstartline</code>	1
<code>\l@dparsedstartpage</code>	1
<code>\l@dparsedstartsub</code>	1
<code>\l@dparsefootspec</code>	1
<code>\l@dpush@begins</code>	1
<code>\l@drd@ta</code>	1
<code>\l@dref@undefined</code>	1
<code>\l@drestorefills</code>	1
<code>\l@drestoreforedtext</code>	1
<code>\l@drp@rbox</code>	1
<code>\l@drsn@te</code>	1
<code>\l@drsnote</code>	1
<code>\l@dsetmaxcolwidth</code>	1
<code>\l@dskipnumberfalse</code>	1
<code>\l@dskipnumbertrue</code>	1
<code>\l@dtabaddcols</code>	1
<code>\l@dtabnoexpands</code>	1
<code>\l@dunboxmpfoot</code>	1
<code>\l@dunhbox@line</code>	1
<code>\l@dzeropenalties</code>	1
<code>\l@noexpands@edgl</code>	1
<code>\l@secondmandarg</code>	1
<code>\l@wrap@ifnotemptybox</code>	1
<code>\l@wrapcs@ifnotemptybox</code>	1
<code>\label</code>	66
<code>\labelpstartfalse</code>	1
<code>\labelpstarttrue</code>	1, 20
<code>\labelref@list</code>	1
<code>\labelrefsparseline</code>	1
<code>\labelrefsparsesubline</code>	1
<code>\last@page@num</code>	1
Lavagnino, John	13
<code>\led@check@nopb</code>	1
<code>\led@check@pb</code>	1
<code>\led@err@AutoparNotNumbered</code>	1
<code>\led@err@BadAction</code>	1
<code>\led@err@edtextoutsidepstart</code>	1
<code>\led@err@EdtextWithoutFootnote</code>	1
<code>\led@err@FootnoteNotInSecondArgEdtext</code>	1
<code>\led@err@HighEndColumn</code>	1
<code>\led@err@LineationInNumbered</code>	1
<code>\led@err@LowStartColumn</code>	1
<code>\led@err@ManyLeftnotes</code>	1
<code>\led@err@ManyRightnotes</code>	1
<code>\led@err@ManySidenotes</code>	1
<code>\led@err@NumberingNotStarted</code>	1
<code>\led@err@NumberingShouldHaveStarted</code>	1
<code>\led@err@NumberingStarted</code>	1
<code>\led@err@NumberingWithoutPstart</code>	1

<code>\led@err@PendInEdtext</code>	1
<code>\led@err@PendNoPstart</code>	1
<code>\led@err@PendNotNumbered</code>	1
<code>\led@err@PstartInEdtext</code>	1
<code>\led@err@PstartInPstart</code>	1
<code>\led@err@PstartNotNumbered</code>	1
<code>\led@err@ReverseColumns</code>	1
<code>\led@err@samewordRTL</code>	1
<code>\led@err@toendnotes@outsidenumbering</code>	1
<code>\led@err@TooManyColumns</code>	1
<code>\led@err@UnequalColumns</code>	1
<code>\led@error@fail@patch@@doclearpage</code>	1
<code>\led@error@fail@patch@@iiiminipage</code>	1
<code>\led@error@fail@patch@@makecol</code>	1
<code>\led@error@fail@patch@@reinserts</code>	1
<code>\led@error@fail@patch@addtocounter</code>	1
<code>\led@error@fail@patch@endminipage</code>	1
<code>\led@error@fail@patch@setcounter</code>	1
<code>\led@error@PackageAfterEledmac</code>	1
<code>\led@mess@NotesChanged</code>	1
<code>\led@mess@SectionContinued</code>	1
<code>\led@nopb</code>	1
<code>\led@nopbnum</code>	1
<code>\led@pb</code>	1
<code>\led@pb@setting</code>	1
<code>\led@pbnum</code>	1
<code>\led@reinit@index@fornote</code>	1
<code>\led@set@index@fornote</code>	1
<code>\led@toksa</code>	1
<code>\led@toksb</code>	1
<code>\led@warn@AppLabelOutSecondArgEdtext</code>	1
<code>\led@warn@BadAction</code>	1
<code>\led@warn@BadAdvancelineLine</code>	1
<code>\led@warn@BadAdvancelineSubline</code>	1
<code>\led@warn@BadHookSeries</code>	1
<code>\led@warn@BadLineation</code>	1
<code>\led@warn@BadLinenummargin</code>	1
<code>\led@warn@BadLockdisp</code>	1
<code>\led@warn@BadSetline</code>	1
<code>\led@warn@BadSetlinenum</code>	1
<code>\led@warn@BadSidenotemargin</code>	1
<code>\led@warn@BadSublockdisp</code>	1
<code>\led@warn@DuplicateLabel</code>	1
<code>\led@warn@edinde@outsidenumbering</code>	1
<code>\led@warn@LineFileObsolete</code>	1
<code>\led@warn@NoFile</code>	1
<code>\led@warn@NoIndexFile</code>	1
<code>\led@warn@NoMarginpars</code>	1
<code>\led@warn@RefUndefined</code>	1
<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</code> (environment)	63
<code>ledgroupsize</code> (environment)	64
<code>\ledinnernote</code>	1, 69
<code>\ledinnote</code>	1
<code>\ledinnotemark</code>	1
<code>\ledleftnote</code>	1, 69
<code>\ledlinenum</code>	1
<code>\ledllfill</code>	1
<code>\ledlsnotefontsetup</code>	1, 70
<code>\ledlsnotesep</code>	1, 70
<code>\ledlsnotewidth</code>	1, 70
<code>\lednopb</code>	1, 78
<code>\lednopbinversetrue</code>	79
<code>\lednopbnum</code>	1
<code>\ledouternote</code>	1, 69
<code>\ledpb</code>	1, 78
<code>\ledpbnum</code>	1
<code>\ledpbsetting</code>	1, 79
<code>\ledrightnote</code>	1, 69
<code>\ledrlfill</code>	1
<code>\ledrsnotefontsetup</code>	1, 70
<code>\ledrsnotesep</code>	1, 70
<code>\ledrsnotewidth</code>	1, 70
<code>\ledsectnomark</code>	1
<code>\ledsectnotoc</code>	1
<code>\ledsetnormalparstuff@common</code>	1
<code>\ledsetnormalparstuffX</code>	1
<code>\ledsidenote</code>	1, 69
<code>\leftctab</code>	1
<code>\leftlinenum</code>	1, 23
<code>\leftlinenumannotation</code>	1
<code>\leftltab</code>	1
<code>\leftnoteupfalse</code>	70
<code>\leftpstartnum</code>	1
<code>\leftrtab</code>	1
<code>Leibniz</code>	14
<code>\lemma</code>	1, 33
<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, 27
<code>\lineannot@set</code>	1

<code>\lineation</code>	1, 22
<code>\linenum</code>	1, 33
<code>\linenum@out</code>	1
<code>\linenumannotation</code>	1, 25
<code>\linenumannotationothersidettrue</code>	27
<code>\linenumberlist</code>	1, 22
<code>\linenumberstyle</code>	1, 24
<code>\linenumincrement</code>	1, 22
<code>\linenummargin</code>	1, 23
<code>\linenumr@p</code>	1
<code>\linenumrep</code>	1
<code>\linenumsep</code>	1, 23
<code>\linerrangesep@</code>	1
<code>\list@clear</code>	1
<code>\list@clearing@reg</code>	1
<code>\list@create</code>	1
<code>\lock@disp</code>	1
<code>\lock@off</code>	1
<code>\lock@on</code>	1
<code>\lockdisp</code>	1, 24
Lorch, Richard	14
<code>\ltab</code>	1
<code>\ltabtext</code>	1
Luecking, Dan	86

M

<code>\m@mmf@check</code>	1
<code>\m@mmf@prepare</code>	1
<code>\M@sect</code>	1
<code>\makehboxofhboxes</code>	1
<code>\manage stanza@modulo</code>	1
<code>\maxhnotesX</code>	57
Mayer, Gyula	14
<code>\measurebody</code>	1
<code>\measuremcell</code>	1
<code>\measuremrow</code>	1
<code>\measuretbody</code>	1
<code>\measuretcell</code>	1
<code>\measuretrow</code>	1
Middleton, Thomas	14, 112
<code>minipage (environment)</code>	63
Mittelbach, Frank	13, 14
<code>\morenoexpands</code>	1, 81
<code>\mp@append@notesX</code>	1
<code>\mp@append@Xnotes</code>	1
<code>\mpfnpos</code>	1, 42
<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, 39
<code>\msdata@c</code>	1
<code>\msdata@cR</code>	1
<code>\multfootsep</code>	1, 41
<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>	45
<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>	52
<code>\notenumfontX</code>	52
<code>\noteschanged@false</code>	1
<code>\noteschanged@true</code>	1
<code>\noteswidthliketwocolumnsX</code>	57
<code>\nulledindex</code>	1
<code>\nullsetzen</code>	1
<code>\num@lines</code>	1
<code>\numberedpar@false</code>	1
<code>\numberedpar@true</code>	1
<code>\numberingfalse</code>	1
<code>\numberingtrue</code>	1
<code>\numberlinefalse</code>	21
<code>\numberlinetrue</code>	21
<code>\numberpstartfalse</code>	1, 20
<code>\numberpstarttrue</code>	1, 20
<code>\numberstanzafalse</code>	62
<code>\numberstanzatrue</code>	62
<code>\numlabfont</code>	1, 58

O

<code>\old@hsize</code>	1
<code>\one@line</code>	1
<code>optionafter</code>	27
<code>optionantilabe</code>	62, 352
<code>optionauxdir</code>	16, 485
<code>optionbefore</code>	27
<code>optioncontinuousnumberingwithcolumns</code>	97, 167, 174, 484, 489
<code>optiondraft</code>	79
<code>optionfinal</code>	79
<code>optioninnnote</code>	483
<code>optioninnote</code>	483
<code>optionlinrangesep</code>	293
<code>optionminus</code>	484
<code>optionnocritical</code>	481, 483
<code>optionnoeledsec</code>	78, 393, 485
<code>optionnoend</code>	483
<code>optionnofamiliar</code>	481, 489
<code>optionnopenalties</code>	81, 485
<code>optionnoprefix</code>	69, 490
<code>optionnoresetlinenumannotation</code>	25, 48, 489
<code>optionnotenumber</code>	483, 486
<code>optionplus</code>	484
<code>optionprefixmore</code>	69, 490
<code>optionsameparallelpagenummer</code>	489
<code>optionswcaseinsensitive</code>	34, 488

P

<code>\page@action</code>	1
<code>\page@num</code>	1
<code>\pagelinesep</code>	1, 72
<code>\pageref</code>	66
<code>\par@line</code>	1
<code>\par@patch@pagenumbering</code>	1
<code>\par@patch@thepage</code>	1
<code>\para@footgroupX</code>	1
<code>\para@footsetup</code>	1
<code>\para@footsetupX</code>	1
<code>\para@vfootnoteX</code>	1
<code>\parafootfmt</code>	1
<code>\parafootfmtX</code>	1
<code>\parafootgroup</code>	1
<code>\parafootsepX</code>	55
<code>\parafootstart</code>	1
<code>\parafootstartX</code>	1
<code>\paravfootnote</code>	1
<code>\parindentX</code>	53
<code>\parse@annot</code>	1
<code>\pausenumbering</code>	1, 21
<code>\pausenumbering@page@num</code>	1
<code>\pend</code>	1, 18
Plato of Tivoli	14
<code>\postbodyfootmark</code>	1
<code>\prebodyfootmark</code>	1
<code>\prenotesX</code>	56
<code>\prepare@edindex@fornote</code>	1
<code>\prepare@prenotesX</code>	1
<code>\prepare@Xgroupbyline</code>	1
<code>\prepare@Xprenotes</code>	1
<code>\prev@nopb</code>	1
<code>\prev@pb</code>	1
<code>\prevpage@num</code>	1
<code>\print@eledsection</code>	1
<code>\print@footnoteXrule</code>	1
<code>\print@leftmargin@eledsection</code>	1
<code>\print@lemma</code>	1
<code>\print@line</code>	1
<code>\print@notesX</code>	1
<code>\print@rightmargin@eledsection</code>	1
<code>\print@Xfootnoterule</code>	1
<code>\print@Xnotes</code>	1
<code>\printendlines</code>	1
<code>\printlineendnote</code>	1
<code>\printlineendnotearea</code>	1
<code>\printlinefootnote</code>	1
<code>\printlinefootnotearea</code>	1
<code>\printlinefootnotenumbers</code>	1

<code>\printlines</code>	1
<code>\printnpnum</code>	1
<code>\printpstart</code>	1
<code>\printsymlineendnotearea</code>	1
<code>\printsymlinefootnotearea</code>	1
<code>\printXafternumber</code>	1
<code>\printXbeforenumber</code>	1
<code>\pstart</code>	1, 18
<code>\pstarteref</code>	1
<code>\pstartnum</code>	1
<code>\pstartref</code>	64
<code>\pstarts@read@L</code>	1
<code>\pstarts@typeset@L</code>	1

Q

<code>\quotation</code>	1
<code>\quote</code>	1

R

<code>\raggedX</code>	55
<code>\raw@text</code>	1
<code>\rbracket</code>	1
<code>\read@linelist</code>	1
<code>\ref</code>	66
<code>\ref@reg@later</code>	1
<code>\Relax</code>	1
<code>\reledmac@error</code>	1
<code>\reledmac@warning</code>	1
<code>\removehboxes</code>	1
<code>\reset@current@annot</code>	1
<code>\reset@msd@options@iffullpage</code>	1
<code>\resetlinenumannotation</code>	1
<code>\resetprevline@</code>	1, 116
<code>\resetprevpage@</code>	1
<code>\resetprevpage@num</code>	116
<code>\restore@endnotes</code>	1
<code>\restore@familiarnotes</code>	1
<code>\restore@notes</code>	1
<code>\restore@sidenotes</code>	1
<code>\resumenumbering</code>	1, 21
<code>\rightctab</code>	1
<code>\rightlinenum</code>	1, 23
<code>\rightlinenumannotation</code>	1
<code>\rightltab</code>	1
<code>\rightnoteupfalse</code>	70
<code>\rightrtab</code>	1
<code>\rightstartnum</code>	1
<code>\rigidbalance</code>	1
<code>\rigidbalanceX</code>	1
<code>\rtab</code>	1

\rtabtext 1

S

Sacrobosco 14
 \sameword 1, 34
 \sameword@inedtext 1
 Schöpf, Rainer 14
 \section@num 1
 \select@lemmafont 1
 \select@lemmafont 1, 58
 \Seref 1, 66
 \Serefonlypage 66
 \Serefwithpage 1, 66
 \series 1
 \seriesatbegin 1, 42
 \seriesatend 1, 42
 \set@continuousnumberingforL 1
 \set@line 1
 \set@line@action 1
 \set@this@c@page 1
 \set@txtbeforenotesX 1
 \set@Xtxtbeforenotes 1
 \setapprefprefixmore 67
 \setapprefprefixsingle 67
 \setcommand@series 1
 \sethangindentX 1
 \sethangingsymbol 1, 61
 \setistwofollowinglines 1
 \setl@dlp@rbox 1
 \setl@drpr@box 1
 \setline 1, 24
 \setlinenum 1, 24
 \setlinenumannotationsep 1, 28
 \setmcellcenter 1
 \setmcellleft 1
 \setmcellright 1
 \setmrowcenter 1
 \setmrowleft 1
 \setmrowright 1
 \setmsdatalabel 1, 40
 \setmsdataposition 1, 41
 \setmsdataseries 1, 40
 \setnoteswidthliketwocolumnsX@ 1
 \setnotesXpositionliketwocolumns@ 1
 \setparindentX 1
 \setprintendlines 1
 \setprintlines 1
 \setSerefonlypageprefixmore 67
 \setSerefonlypageprefixsingle 67
 \setSerefprefixmore 67

<code>\setSErefprefixsingle</code>	67
<code>\setsidenotesep</code>	70
<code>\setstanzaindents</code>	1, 58
<code>\setstanzapenalties</code>	1, 60
<code>\setstanzavalues</code>	1
<code>\settccllcenter</code>	1
<code>\settccllleft</code>	1
<code>\settccllright</code>	1
<code>\settoggle@series</code>	1
<code>\setthrowcenter</code>	1
<code>\setthrowleft</code>	1
<code>\setthrowright</code>	1
<code>\setXnotespositionliketwocolumns@</code>	1
<code>\setXnoteswidthliketwocolumns@</code>	1
<code>\showlemma</code>	1, 79
<code>\showwordrank</code>	1, 38
<code>\sidenote@margin</code>	1
<code>\sidenotemargin</code>	1, 69
<code>\sidepstartnumtrue</code>	20
<code>\skip@lockoff</code>	1
<code>\skipnumbering</code>	1, 25
<code>\splitoff</code>	1
<code>\spreadmath</code>	1, 75
<code>\spreadtext</code>	1, 75
<code>\stanza</code>	1, 58
<code>\stanza@count</code>	1
<code>\stanza@chang</code>	1
<code>\stanza@indent</code>	1
<code>\stanza@line</code>	1
<code>\stanzaindent</code>	1, 60
<code>\stanzaindent*</code>	1, 60
<code>\stanzaindentbase</code>	1, 58
<code>\stanzanumwrapper</code>	1, 62
<code>\startlock</code>	1, 24
<code>\startsub</code>	1, 23
<code>\stepl@dcclcount</code>	1
<code>\stopmsdata</code>	1, 40
<code>\store@annot@to@absline</code>	1
<code>\strip@szacnt</code>	1
<code>\sub@action</code>	1
<code>\sub@lock</code>	1
<code>\sub@off</code>	1
<code>\sub@on</code>	1
<code>\subline@num</code>	1
<code>\sublinenumberstyle</code>	1, 24
<code>\sublinenumincrement</code>	1, 22
<code>\sublinenumr@p</code>	1
<code>\sublinenumrep</code>	1
<code>\sublineref</code>	1, 64
<code>\sublines@false</code>	1

\sublines@true	1
\sublock@disp	1
\sublockdisp	1
Sullivan, Wayne	14, 81, 96, 101, 200, 201, 302, 344
\sw@noexpand	1
\swnoexpands	38
\sza@penalty	1

T

\tabHilfbox	1
\tabhilfbox	1
\theadcolcount	1
\theadtext	1
\theendpageline	1
\thefootnoteA	41
Theodosius	14
\thepageline	1
\thepar@page	1
\thepstart	1, 20
\thestanza	1, 62
\thestartpageline	1
\this@line@list@version	1
\this@section@last@page@num	1
\this@section@page@num	1
\threecol@begin@insert	1
\threecolfootfmt	1
\threecolfootfmtX	1
\threecolfootgroup	1
\threecolfootgroupX	1
\threecolfootsetup	1
\threecolfootsetupX	1
\threecolvfootnote	1
\threecolvfootnote@inserted	1
\threecolvfootnoteX	1
\toendnotes	1, 32
\toendnotes*	1
\twocolfootfmt	1
\twocolfootfmtX	1
\twocolfootgroup	1
\twocolfootgroupX	1
\twocolfootsetup	1
\twocolfootsetupX	1
\twocolvfootnote	1
\twocolvfootnote@inserted	1
\twocolvfootnoteX	1
\txtbeforenotesonlyonceX	56
\txtbeforenotesX	55

U

\unvxhX	1
---------	---

V

Vamana	14
\variab	<u>1</u>
\vbfnoteX	<u>1</u>
\vl@dbfnote	<u>1</u>
\vl@dcsnote	<u>1</u>
\vl@disnote	<u>1</u>
\vl@dlsnote	<u>1</u>
\vl@dosnote	<u>1</u>
\vl@drsnote	<u>1</u>
\vnumfootnoteX	<u>1</u>

W

Whitney, Ron	14
\widthX	57
\wrap@edcrossref	<u>1</u>
\wrapcontentX	53
\wrapped@bodyfootmarkX	<u>1</u>
\wrapped@footfootmarkX	<u>1</u>
Wujastyk, Dominik	13

X

\X@atbegininsertion	<u>1</u>
\X@beforeinsertion	<u>1</u>
\X@doreinfeet	<u>1</u>
\xabslineref	<u>1</u>
\Xafterlemmaseparator	51
\Xafternote	55
\Xafternumber	49
\Xafterrule	56
\Xaftersymlinenum	49
\xannotationref	<u>1</u> , 65
\Xarrangement	<u>1</u> , 44
\Xarrangement@normal	<u>1</u>
\Xarrangement@paragraph	<u>1</u>
\Xarrangement@threecol	<u>1</u>
\Xarrangement@twocol	<u>1</u>
\Xbeforeinserting	54
\Xbeforelemmaseparator	51
\Xbeforenotes	56
\Xbeforenumber	49
\Xbeforesymlinenum	49
\Xbhookgroup	56
\Xbhooknote	54
\Xboxlinenum	50
\Xboxlinenumalign	50
\Xboxsymlinenum	50
\Xcolalign	54
\xedindex	<u>1</u>
\xedlabel	<u>1</u>

\xeditext	1
\Xendafterlemmaseparator	51
\Xendafternote	57
\Xendafterpagenumber	46
\Xendaftersymlinenum	49
\Xendaftrenumber	49
\Xendahookinplaceofnumber	53
\Xendahooklinenum	53
\Xendbeforelemmaseparator	51
\Xendbeforenumber	49
\Xendbeforepagenumber	46
\Xendbeforesymlinenum	49
\Xendbhookinplaceofnumber	53
\Xendbhooklinenum	53
\Xendbhooknote	54
\Xendboxendlinenumalign	50
\Xendboxlinenum	50
\Xendboxlinenumalign	50
\Xendboxstartlinenumalign	50
\Xendboxsymlinenum	50
\Xendhangindent	53
\Xendinplaceofflemmaseparator	51
\Xendinplaceofnumber	49
\Xendinplaceofpagenumber	46
\Xendlemmadisablefontselection	52
\Xendlemmafont	52
\Xendlemmaseparator	51
\Xendlinenumannotationonlyfirst	48
\Xendlinenumannotationonlyfirstintwo	48
\Xendlinenumannotationposition	27
\Xendlineprefixmore	46
\Xendlineprefixsingle	46
\Xendlinerangeseparator	46
\Xendmorethantwolines	47
\Xendnoidenticallinenumannotation	28
\Xendnolinenum	48
\Xendnolinenumifannotation	48
\Xendnonumber	47
\Xendnotefontsize	52
\Xendnotenumfont	51
\Xendnumberonlyfirstinline	45
\Xendnumberonlyfirstintwolines	45
\Xendpagenumberonlyfirst	45
\Xendparagraph	57
\Xendsep	57
\Xendstorelineinfo	1
\Xendsublinesep	49
\Xendsymlinenum	45
\Xendsymlinenumannotation	48
\Xendsympagenum	46

\Xendtwolines	47
\Xendtwolinesbutnotmore	47
\Xendtxtbeforenotes	56
\Xendwrapcontent	53
\xflagref	<u>1</u>
\Xgroupbyline	55
\Xgroupbylineseparetwolines	55
\Xhangindent	53
\Xhsizethreecol	54
\Xhsizetwocol	54
\Xinnotemark	71
\Xinplaceoflemmaseparator	51
\Xinplaceofnumber	49
\Xinsertparafootsep	<u>1</u>
\Xledsetnormalparstuff	<u>1</u>
\xleft@appenditem	<u>1</u>
\Xlemmadisablefontselection	52
\Xlemmafont	52
\Xlemmaseparator	51
\Xlinenumannotationonlyfirst	48
\Xlinenumannotationonlyfirstintwo	48
\Xlinenumannotationposition	27
\Xlinenumannotationposition@side	<u>1</u>
\Xlinenumannotationpositionside	27
\Xlinrangeseparator	46
\xlineref	<u>1</u> , 65
\Xmaxhnotes	56
\Xmorethantwolines	46
\Xnoidenticallinenumannotation	28
\Xnolemmaseparator	<u>1</u> , 51
\Xnolinenum	48
\Xnolinenumifannotation	48
\Xnonbreakableafternumber	49
\Xnonote	45
\Xnonumber	47
\Xnotefontsize	52
\Xnotenumfont	51
\Xnoteswidthliketwocolumns	57
\Xnumberonlyfirstinline	45
\Xnumberonlyfirstintwolines	45
\Xonlypstart	47
\Xpagelinesep	49
\xpageref	<u>1</u> , 65
\Xparafootsep	55
\Xparindent	53
\Xprenotes	<u>1</u> , 56
\Xprenotes@	<u>1</u>
\Xpstart	47
\Xpstarteverytime	47
\Xpstartonlyfirst	47

\xpstartref	1, 65
\Xpstartseparator	47
\XR@test	1
\XR@test@mac	1
\XR@test@mac@test	1
\Xragged	55
\xright@appenditem	1
\Xrigidbalance	1
\Xsethangindent	1
\Xsetparindent	1
\Xstanza	48
\Xstanzaonlyfirst	48
\Xstanzaseparator	48
\Xstorelineinfo	1
\xsublineref	1, 65
\Xsublinesep	23, 49
\Xsublinesep side	23
\Xsymlinenum	45
\Xsymlinenumannotation	48
\Xtoendnotes	32
\Xtwolines	46
\Xtwolinesonlyinsamepage	47
\Xtxtbeforenotes	55
\Xtxtbeforenotesonlyonce	56
\Xtxtbeforenumber	46
\Xunvxh	1
\Xwidth	57
\Xwrapcontent	53
\Xwrapendlemma	52
\Xwraplemma	52
\Xwraplinenumannotation@ref	1
\Xwraplinenumannotation@side	1
\xxref	1, 65

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	87
\morenoexpands: Added \l@dtabnoexpands to \no@expands	143
\reledmac@error: Added \eledmac@error and replaced error messages	88
General: Added tabmac code, and extended indexing	1
v0.2.1.	
\@lab: Removed page setting from \@lab	305
\edlabel: Tweaked \edlabel to get correct page numbers	302
\l@ddodoreintrafeet: Renamed \dodoreintrafeet to \l@ddodoreintrafeet	295
\l@ddoxtrafeet: Renamed \doxtrafeet to \l@ddoxtrafeet	294
\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	143
General: Added text about normal labeling	66
Bug fixes and match with mempatch v1.8	1
Major changes to insert code when memoir is loaded	297
v0.2.2.	
\footfudgefiddle: Added \footfudgefiddle	198
\next@line@list@stuff: Added initial write of page number in \line@list@stuff	133
\para@footsetup: Added \footfudgefiddle to \para@footsetup	199
\para@footsetupX: Added \footfudgefiddle to \para@footsetupX	244
General: Improved paragraph footnotes	1
New Dekker example	1
Used \providecommand for \@gobblethree to avoid clash with the amsfonts package	95
v0.3.0.	
\@lab: Replaced \the\line@num by \linenumr@p\line@num in \@lab, and similar for sub-lines	305
\@nl@reg: Added a bunch of code to \@nl for handling \setlinenum	119
\ledlinenum: Added \linenumr@p and \sublinenumr@p to \leftlinenum and \rightlinenum	110
\linenumberlist: Added \linenumberlist mechanism	96
\printendlines: Added \linenumr@p and \sublinenumr@p to \printendlines	265
\printlines: Added \linenumr@p and \sublinenumr@p to \printlines	226
\sublinenumr@p: Added \linenumberstyle and \sublinenumberstyle	109
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	329
\Xarrangement@normal: Added minpage footnote setup to \footnormal	193
\edtext: Added \showlemma to \edtext	144
\l@dfeetendmini: Added \l@dfeetbeginmini, \l@dfeetendmini and all their supporting code	327
\mpnormalfootgroup: Added \mpnormalfootgroup	197

\mpnormalvfootnote: Added \mpnormalvfootnote	195
\showlemma: Added \showlemma	96
General: Added final/draft options	84
Added ledgroup environment	330
Added ledgroupsize environment	331
Added minipage, etc., support	1
v0.4.1.	
\do@Xfeet: Changed \do@Xfeet code for easier extensions	294
\edindex: Let eledmac take advantage of memoir's indexing	336
\print@Xnotes: Added \opXfeet	295
General: Added code for changing \doclearpage	298
Not released. Minor editorial improvements and code tweaks	1
Only change \@footnotetext and \@footnotemark if memoir not used	230
v0.5.0.	
\@footnotetext: Enabled regular \footnote in numbered text	231
\@xympar: Eliminated \marginpar disturbance	319
General: Added left and right sidenotes	320
Added sidenotes, familiar footnotes in numbered text	1
v0.5.1.	
\affixline@num: Changed \affixline@num to cater for sidenotes	177
\l@getsidenote@margin: Added \sidenotemark and \sidenote@margin	320
General: Added moveable sidenote	320
Fixed right line numbers killed in v0.5	1
Only change \hsize in ledgroupsize environment otherwise page number can be in wrong place	331
v0.6.0.	
\@lopR: Added \@pend, \@pendR, \@lopL and \@lopR in anticipation of parallel processing	122
\@nl@reg: Extended \@nl to include the page number	119
\get@thisfootnote: Changed \l@dbfnote and \v1@dbfnote as originals could give incorrect markers in the footnotes	231
\set@thiscc@page: Extended \new@line to output page numbers	134
General: Fixed long paragraphs looping	1
Fixed minor typos	1
Prepared for eledpar package	1
v0.7.0.	
\@nl@reg: Added \@nl@reg	119
\@ref@reg: Added \@ref@reg	130
\affixline@num: Added skipnumering to \affixline@num	177
\do@actions@fixedcode: Added \do@actions@fixedcode	173
\do@actions@next: Added number skipping to \do@actions	171
\do@insidelinehook: Added \do@linehook for use in \do@line	168
\endnumbering: Changed \endnumbering for eledpar	100
\fx@l@cks: Added \ch@cksub@l@ck, \ch@ck@l@ck and \fx@l@cks	180
\footsplitskips: Added \footsplitskips for use in many footnote styles	191
\get@linelistfile: Added \get@linelistfile	118
\initnumbering@reg: Added \initnumbering@reg	99
\l@advance@parledgroup@beforenormalnotes: Added \l@dunboxmpfoot containing some common code	330
\l@dcsnotetext@r: Added \l@emptyd@ta	169

\l@dgetline@margin: Added \l@dgetline@margin	106
\l@dgetlock@disp: Added \l@dgetlock@disp	108
\l@dgetsidenote@margin: Added \l@dgetsidenote@margin	320
\l@dnumpstartsL: Added \l@dnumpstartsL, \ifl@dpairing and \ifpst@rted for/from eledpar	96
\l@drsn@te: Added \l@dlsn@te and \l@drsn@te for use in \do@line	169
\l@dzeropenalties: Added \l@dzeropenalties	163
\ledlinenum: Added \ledlinenum for use by \leftlinenum and \rightlinenum	110
\list@clearing@reg: Added \list@clearing@reg	118
\n@num: Added \n@num	127
\next@line@list@stuff: Deleted \page@start from \line@list@stuff	133
\normalbfnoteX: Removed extraneous space from \normalbfnoteX	236
\pausenumbering@page@num: Changed \resumenumbering for eledpar	102
\setprintlines: Added \setprintlines for use by \printlines	220
\skipnumbering: Added \skipnumbering and supports	140
\sublinenumincrement: Added \firstlinenum, \linenumincrement, \firstsublinenum and \linenumincrement	107
\sublinenumr@p: Using \linenumrep instead of \linenumr@p	109
Using \sublinenumrep instead of \sublinenumr@p	109
\vnumfootnoteX: Removed extraneous space from \vnumfootnoteX	238
General: Added \setprintendlines for use by \printendlines	260
eledmac having been available for 2 years, deleted the commented out original edmac texts	1
Maïeul Rouquette new maintainer	1
Made macros of all messages	88
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	96
General: For compatibility with eledpar, possibility to use \autopar on the right side.	1
Possibility to number \pstart.	20
Possibility to number the pstart with the commands \numberpstarttrue.	1
v0.12.1.	
General: Do not number \pstarts of stanza.	1

The numbering of <code>\pstart</code> restarts on each <code>\beginnumbering</code> .	1
v0.13.0.	
<code>\managestanza@modulo</code> : New <code>stanzaindentsrepetition</code> counter to repeat stanza indents every n verses.	346
General: New <code>stanzaindentsrepetition</code> counter to repeat stanza indents every n verses.	59
New <code>stanzaindentsrepetition</code> counter: to repeat stanza indents every n verses.	1
v0.13.1.	
General: <code>\thepstartL</code> and <code>\thepstartR</code> use now <code>\bfseries</code> and not <code>\bf</code> , which is deprecated and makes conflicts with <code>memoir</code> class.	1
v0.14.0.	
<code>\edlabel</code> : Tweaked <code>\edlabel</code> to get correct line number if the command is first element of a paragraph.	302
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.	177
<code>\ifinserthangingsymbol</code> : New management of <code>hangingsymbol</code> insertion, preventing undesirable insertions.	345
<code>\printlines</code> : Line numbering can be reset at each <code>pstart</code> .	226
General: Line numbering can be reset at each <code>pstart</code> .	104
Possibility to print <code>\pstart</code> number at the sides of the text.	20
v0.17.0.	
<code>\ifinserthangingsymbol</code> : New new management of <code>hangingsymbol</code> insertion, preventing undesirable insertions.	345
v1.0.0.	
<code>\morenoexpands</code> : Change to be compatible with new features	143
General: <code>\lemma</code> can contain commands.	33
Debug in lineation command	22
New generic commands to customize footnote display.	43, 283
Options <code>nonum</code> and <code>nosep</code> in <code>\Xfootnote</code> .	30
Options of <code>\Xfootnotes</code> .	189
Possibility to have commands in sidenotes.	69
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.	45
v1.1.0.	
<code>\Xprenotes</code> : New skip <code>\Xprenotes@</code>	253
<code>\settoggle@series</code> : <code>\settoggle@series</code> switch the global value of the toggle, not only the local value.	283
General: Add <code>\labelpstarttrue</code> .	20
Add <code>\Xnumberonlyfirstintwolines</code>	45
Add <code>\Xpstart</code> and <code>\Xonlypstart</code>	47
New hook to add arbitrary code at the beginning of the notes	53
New options for block of notes.	55
New package option: <code>parapparatus</code> .	1
New tools to change order of series	282
Sectioning commands.	77
v1.2.0.	
<code>\Xprenotes</code> : Debug in familiar footnotes (bug introduced by v1.1).	253
<code>\endquote</code> : Compatibility of <code>\ledchapter</code> with the <i>memoir</i> class.	386

v1.3.0.	
\endquote:	<i>Quotation</i> and quote environment inside numbered sections. 386
v1.4.0.	
\edtext:	Compatibility of \edtext with the right-to-left direction (with Polyglossia). 144
\ledsetnormalparstuffX:	Direction of footnotes with polyglossia. 249
\newseries@:	Remembers the language of the lemma, in order to create a correct direction for the footnote separator. 272
\rbracket:	Switch the right bracket to a left bracket when the lemma is RTL (needs polyglossia or LuaTeX). 213
General:	Compatibility with LuaTeX of RTL notes. 1
v1.4.1.	
\affixside@note:	Remove spurious spaces. 326
\endquote:	New option <i>noquotation</i> 386
\get@thisfootnote:	Compatibility of standard footnotes with eledmac when these footnotes contain any commands. 231
\labelrefsparsesubline:	Fix a bug with \edlabel. 304
v1.4.2.	
General:	Debug with some special classes. 1
v1.4.3.	
General:	Add \Xnonbreakableafternumber. 49
	Spurious space after familiar footnotes. 1
v1.4.4.	
General:	Label inside familiar footnotes. 1
v1.4.5.	
General:	Bug with komasscript + eledpar + chapter. 1
v1.4.6.	
General:	Bug with memoir class introduced by 1.4.5. 1
v1.4.7.	
\endquote:	Compatibility of sectioning commands with \autopar. 386
v1.4.8.	
General:	Corrects a bug with parallel texts introduced by 1.1. 1
v1.4.9.	
\normalbfnoteX:	Allow to redefine \thefootnoteX with alph when some packages are loaded. 236
v1.5.0.	
\do@insidelinehook:	Added \do@insidelinehook for use in \do@line 168
\edindex:	Compatibility with imakeidx package, and possibility to use multiple index with \edindex. 336
General:	Correct indexing when the call is made in critical notes. 332
v1.5.1.	
\managestanza@modulo:	Correct stanzaindentsrepetition counter 346
\normalvfootnoteX:	Fix a bug with normal familiar footnotes when mixing RTL and LTR text. 234
v1.6.0.	
\newverse:	Add \falseverse macro. 350
v1.6.1.	
\AtStartEveryPstart:	Spurious space in \pstart. 159
\ifinserthangingsymbol:	Hang verse is now not automatically flush right. 345
\l@dunhbox@line:	Move the call to \inserthangingsymbol to allow use \hfill inside. 165

\pend: Spurious space in \pend.	161
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.	78
v1.8.0.	
\endquote: Correction of sectioning commands in parallel texts.	386
\get@index@command: Debug \get@index@command and compatibility with hyperref package.	335
\newhookcommand@series@reload: Debug \beforenotesX and \maxhnotesX which did not work.	286
\prevpage@num: Correct \parafootsep when using with ledgroup.	204
General: Compatibility with parledgroup option ofeledpar package.	1
If imakeidx and hyperref are loaded, adds hyperref in the index.	332
v1.8.1.	
General: Debug endnotes when more than one series is used (change the position where tools for endnotes are defined).	254
v1.8.2.	
General: Debug compatibility problem with hebrew option of babel package.	1
v1.8.3.	
General: Fix spurious spaces added by v1.7.0.	1
v1.8.5.	
General: Debug indexing in right column, witheledpar.	332
v1.9.0.	
\l@ddoxtrafeet: Add \fnpos to choice the order of footnotes.	294
\l@dfteetendmini: Add \mpfnpos to choice the order of footnotes in minipage / ledgroup.	327
v1.10.0.	
\endquote: Correction of sectioning commands in parallel texts.	386
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.	52
v1.11.1.	
General: Correct a bug when a critical note starts with plus or minus.	1
v1.12.0.	
\@nl@reg: To ensure compatibility with \musixtex, \@l becomes \@l. Consequently, \@l@reg becomes \@nl@reg.	119

\AtStartEveryPstart: New optional argument for \pstart, to execute code before it.	159
\edindex: Use correctly default index when imakeidx is loaded.	336
\endquote: \ledxxx sectioning commands are deprecated and replaced by \eledxxx commands.	386
\initnumbering@reg: \beginnumbering is defined only on eledmac, not on eledpar.	99
\l@dgetsidenote@margin: \sidenotemargin is now directly defined in eledmac to be able to manage eledpar.	320
\l@disnote: \l@dlsnote, \l@drsnote and \l@dcnote defined only one time, in eledmac, including needs for eledpar case.	321
\l@dnumpstartsL: Add \ifledRcol@ for eledpar	96
\l@dunhbox@line: \do@line is split in more little commands.	165
\newhookcommand@series@reload: Debug \beforenotesX and \maxhnotesX which did not work when called after \footparagraphX.	286
Debug \Xbeforenotes and \Xmaxhnotes which did not work when called after \footparagraph.	286
\pend: New optional argument for \pend, to execute code after it.	161
\stanza: &can have an optional argument: content to be printed after.	350
\Stanza can have an optional argument: content to be printed before.	350
Add \newverse macro, \falseverse deprecated.	350
General: Add \ledinnernote and \ledouternote commands.	69
Add \Xendparagraph and related settings.	57
Add hyperlink to crossref (needs hyperref package).	64
Compatibility with musixtex.	1
Debug eledmac sectioning command after using \resumenumbering.	1
Ensure that imakeidx is loaded <i>before</i> eledmac	332
New hooks: \Xafterrule and \afterruleX	56
New options for ragged-paragraph notes	55
New sectioning commands.	77
Optional arguments for \pstart and \pend.	19
v1.12.1.	
\wrap@edcrossref: Fix spurious spaces.	308
v1.12.2.	
\l@dunhbox@line: Fix a bug with critical notes at the tops of pages (added by v12.0.0)	165
v1.12.3.	
\flag@end: \flag@start and \flag@end are now defined only one time for eledmac and eledpar	135
\flag@start send a error message when a \edtext is done without insert (note)	135
\reledmac@error: Replaced error messages	88
General: Add macros for new messages since v0.7	88
Correct bug with side and familiar notes in tabular environments.	1
Debug \eledxxx with some paper size	1
Debug \ledinnernote and \ledouternote commands in the top of pages.	69
Debug left and right notes (bugs added by 1.12.0)	1
Underline lemma in \eledxxx when using draft mode.	1
v1.12.4.	
General: Debug spurious page breaks before \chapter (bug added in 1.12.0)	1
v1.12.5.	
\@edindex@hyperref: Debug \edindex when hyperref is not loaded	338
\@ssect: Debug \eledchapter in parallel with memoir	389
\doinsidelinehook: Added \dolinehook and \doinsidelinehook	168

\endnumbering: Allow to mix parallel columns and normal text when using	
\pausenumbering	100
\l@dobbleoptarg: \l@dobblearg becomes \l@dobbelloptarg	369
\l@restoreforedtext: Debug optional arguments of \Xfootnote in tabular context	370
\pausenumbering@page@num: Debug \resumenumbering	102
v1.12.7.	
\wrap@edcrossref: \wrap@edcrossref is now robust	308
v1.12.8.	
\flag@end: \flag@start do not send a error message when a \edtext is done	
without insert (note) but have a endnote	135
v1.13.0.	
\newhooktoggle@series@reload: Add \newhookcommand@toggle@reload	286
\para@footsetupX: In \para@footsetupX, use \columnwidth instead of \hsize	244
\settoggle@series: \settoggle@series can take an optional arguments to reload	
series setup.	283
General: Add \Xnoteswidthliketwocolumns and \noteswidthliketwocolumnsX	57
Added widthliketwocolumns option	84
v1.13.1.	
\ifat@every@pstart@star@: Add \l@dzeropenalties in \pstart	159
General: Coming back of page and line breaking penalties's management, deleted by	
error in v0.17.	1
Debug quotation environment inside of a \pstart preceded by a sectioning command.	1
v1.13.2.	
\l@dnumpstartsL: Add \ifl@dpaging for eledpar	96
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	336
General: Debug spurious characters before endnotes.	254
Delete previous override of \l@d@@wrindexhyp at the beginning of a document	
when hyperref is not loaded.	340
Move gobbling command	95
Provide \@gobblefour	95
v1.14.1.	
\@ssect: Debug sectioning commands when using both handout and hyperref	
package.	392
v1.14.2.	
\@ssect: Debug \edtext after starred sectioning commands when using memoir class.	389
v1.15.0.	
\@edtext@level: New boolean \if@edtext@.	144
\arrangementX@paragraph: Correct bug with paragraphed familiar footnotes setting.	244
\endsub: Restore subline feature (disabled by mistake in v1.8.0).	138
\if@lemmacommand@: New boolean \iflemmacommand@.	149
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.	20
New tools to prevent ambiguous references in lemma	34

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.
	133
v1.16.0.	
General:	\edtext is now defined only in eledmac, not in eledpar. Debug wrong numbering when using \sameword + eledpar + \tag command.
	144
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.
	308
v1.17.0.	
\edtext:	Error message when calling \edtext outside of a numbered paragraph. . . .
	144
v1.18.0.	
\@edindex@hyperref:	Fix spurious space with \edindex when using imakeidx/indextools + hyperref.
	338
\edlabel:	\edlabel is now defined only one time for both eledmac and eledpar . . .
	302
\l@d@section:	Option parapparatus works for endnotes.
	255
\l@dnumpstartsL:	Add \ifl@dprintingpages and \@dprintingcolumns for eledpar
	96
\print@line:	Compatibility with Lua \TeX RTL languages.
	165
\printlinefootnote:	Code refactoring in \printlinefootnote: the printing of the numbers are factorized in \printlinefootnotearea
	215
\printpstart:	Debug \Xpstart with parallel pages and columns (eledpar)
	214
General:	Add \Xpstarteverytime
	47
Compatibility with Lua \TeX RTL languages.
	1
Debug \Xonlypstart when using \Xnumberonlyfirstinline and the current line number differs from the previous.
	47
v1.19.0.	
\footplitskips:	\footplitskips doesn't set \floatingpenalty to \@MM when processing parallel pages.
	191
\xxref:	\xxref works also with right side numbers, when \@Rlineflag is not empty. . .
	311
General:	\Xmaxhnotes and \maxhnotesX work now for both two-columns and three-columns setting.
	1
Compatibility with eledpar v1.13.0.
	1
v1.19.1.	
General:	Call \correct@footinsX@box and \correct@Xfootins@box directly in \print@notesX@forpages and \print@Xnotes@forpages, that is in eledpar. . .
	1
v1.20.0.	
\printlines:	Added \ifl@d@Xmorethantwolines and \ifl@d@Xmorethantwolines to \printlines
	226
\stanza:	& and \& can be preceded by spaces.
	350
\xxref:	Debug \xxref when not loading eledpar (Fix a bug added in 1.19.0).
	311
General:	Add \Xendboxlinenum
	50
Add \Xtwolines and \Xmorethantwolines hooks
	46
Add series option.
	1
Correct \Xinplaceofnumber hook.
	1
Explicit error message when calling \Xfootnote outside of \edtext.
	1
Fix a bug with line number typesetting direction when using \eledsection and similar commands for RTL texts with Lua \TeX
	1

Fix issues with RTL text in notes when using Lua \TeX .	1
Options fulllines in <code>\Xfootnote</code> .	30
The <code>\newifs</code> are not followed by boolean values set to false, because it is the \TeX default setting.	1
v1.21.0.	
<code>\@edindex@hyperref</code> : Look at the hyperindex option of hyperref before inserting hyperref	338
<code>\l@d@section</code> : <code>\endnotes</code> take five arguments.	255
<code>\ledinnotemark</code> : Add <code>\ledinnotemark</code> .	335
<code>\ledsetnormalparstuffX</code> : <code>\ledsetnormalparstuff</code> is deprecated and becomes <code>\ledsetnormalparstuffX</code> and <code>\Xledsetnormalparstuff</code> .	249
<code>\n@num</code> : <code>\n@num@ref</code> deleted	127
<code>\n@num</code> defined only one time for both <code>Eledmac</code> and <code>Eledpar</code>	127
<code>\newhookcommand@series</code> : <code>\newhookcommand@series</code> can take an optional argument.	285
<code>\newhooktoggle@series</code> : <code>\newhooktoggle@series</code> can take an optional argument.	286
<code>\print@footnotexrule</code> : Code refactoring: the spaces after the footnote rules are directly managed in <code>\print@Xfootnoterule</code> and <code>\print@footnotexrule</code>	252
<code>\seriesatend</code> : Fix spurious space in <code>\seriesatend</code>	282
<code>\skipnumbering</code> : <code>\skipnumbering</code> defined only one time for both <code>Eledmac</code> and <code>Eledpar</code> .	140
Correct <code>\skipnumbering</code> for stanza.	140
Delete <code>\skipnumbering@reg</code> .	140
General: <code>\AtEveryPstart</code> and <code>\AtEveryPend</code> are now compatible with <code>\autopar</code>	1
<code>\Xafterrule</code> and <code>\afterruleX</code> features no longer create problems of overflowing at the bottom of the page.	1
<code>\chapter</code> inside optional argument of <code>\pstart</code> works when typesetting parallel pages	1
<code>\preXnotes</code> and <code>\prenotesX</code> features no longer create problems of overflowing at the bottom of the page.	1
<code>\seriesatbegin</code> and <code>\seriesatbegin</code> more efficient	282
Add <code>\applabel</code> and related	67
Add <code>\beforenotesX</code> and <code>\Xbeforenotes</code> features for notes set in two and three column.	1
Add <code>\hidenumbering</code>	25
Add <code>\Xcolalign</code> and <code>\colalignX</code> hooks	54
Add <code>\Xendtwolines</code> , <code>\Xendmorethantwolines</code> , <code>\Xendtwolinesbutnotmore</code> and <code>\Xendtwolinesonlyinsamepage</code> .	47
Add <code>\Xparindent</code> and <code>\hangindentX</code>	53
Add <code>\Xtwolinesbutnotmore</code> and <code>\Xtwolinesonlyinsamepage</code> .	1
Add <code>nocritical</code> , <code>noend</code> , <code>nofamiliar</code> and <code>noledgroup</code> options.	1
Add <code>noeledsec</code> package option	1
Debug <code>\beforenotesX</code> <code>\maxhnotesX</code> <code>\noteswidthliketwocolumnsX</code> and <code>\afterruleX</code> with footnotes set in two and three columns.	1
Fix a bug when a <code>\Xfootnote</code> follows a <code>\Xendnote</code> in the second argument of <code>\edtext</code> (bug added in <code>eledmac 1.0.0</code>).	1
Fix a bug with <code>\maxhnotesX</code> when using <code>\foottwocolX</code> or <code>\footthreecolX</code> .	1
Fix a bug with space between columns with notes in two columns (bug added in v1.13.0).	1

Fix spurious space after first page number in <code>\doendnotes</code> . <code>oldprintnpnumspace</code> option allows to come back to previous setting	1
<code>parapparatus</code> option works now with familiar footnotes.	1
Provide <code>\@gobblefive</code>	95
v1.22.0.	
<code>\ledinnote</code> : <code>\ledinnote</code> takes a first optional argument, which is the label for hyperlinks.	335
General: Add <code>\doendnotesbysection</code> command.	31
Add option for lemma separator inside endnotes	51
Adds hyperlink for references to notes in indices.	1
Fix conflict between <code>noend</code> package option and <code>edtabularx</code> environments	1
Provides support for <code>xindy</code> .	1
Standardize endnotes handbook.	31
When using <code>hyperref</code> package, internal links in index or with <code>\edlineref</code> are now targeted to the top and not longer to the bottom of the lines they refer to.	1
v1.22.1.	
<code>\prevpage@num</code> : Correct double symbol when using both <code>\parafootsep</code> and <code>\Xsymlinenum</code> .	204
General: Fix a bug (added on v1.22.0) with <code>\Xinplaceofnumber</code> hook.	1
v1.23.0.	
<code>\@edtext@level</code> : The boolean <code>\if@edtext@</code> becomes the counter <code>\edtext@level</code> .	144
<code>\Serefwithpage</code> : Debug <code>\Xendtwolines</code> , <code>\Xendmorethantwolines</code> , <code>\Xendtwolinesbutnotmore</code> and <code>\Xendtwolinesonlyinsamepage</code> when using <code>\apprefwithpage</code> .	315
<code>\lemma</code> : Fix spurious space after <code>\lemma</code> command	149
<code>\newseries@</code> : Prevent spurious spaces when <code>\Afootnote</code> and similar commands are followed by spaces (bug added on 1.0.0).	272
<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.	154
General: Add <code>\Xboxlinenumalign</code> and <code>\Xendboxlinenumalign</code> .	50
Add <code>\Xboxstartlinenum</code> , <code>\Xendboxstartlinenum</code> , <code>\Xboxendlinenum</code> , <code>\Xendboxendlinenum</code> .	50
Allow use of <code>\sameword</code> with <code>inputenc</code> managing of UTF-8.	1
Compatibility between <code>nofamiliar/nocriticals</code> option and <code>minipage/ledgroup</code> .	1
Error message when using <code>\beginnumbering... \endnumbering</code> without <code>\pstart</code> .	1
Fix a bug with <code>\sameword</code> when the lemma overlaps multiple line.	34
Fix a bug with <code>\sameword</code> when the same lemma is used for multiple notes or for nested <code>\edtexts</code> .	34
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.	34
v1.23.1.	
General: Fix a bug with <code>\lemma</code> command in the right side.	1
v1.23.2.	
General: Compatibility with \LaTeX 's release 2015.	1

v1.24.0.	
General: We can reinitialize <code>\AtEveryPstart</code> and <code>\AtEveryPend</code> providing to it an empty argument.	1
v1.24.1.	
General: <code>\lemma</code> is disabled when using <code>nocritical</code> option.	1
v1.24.2.	
General: Fix incompatibility between <code>nofamiliar</code> option and <code>memoir</code> package.	1
v1.24.3.	
General: Restore marginal numbers and notes with sectioning command (bug introduced in v1.21.0)	1
v1.24.4.	
General: Fix spurious space with <code>\edindex</code> when using <code>xindy+hyperref</code> option.	1
v1.24.5.	
General: Fix a bug of indent, when a added in 1.1.0, when a <code>\beginnumbering</code> immediately follow a sectioning command.	1
v2.0.0.	
<code>\@iiiminipage</code> : Patch <code>\@iiiminipage</code> instead of redefining it.	329
<code>\@xympar</code> : Patching <code>\@xympar</code> instead of redefining it	319
<code>\endminipage</code> : Patch <code>\endminipage</code> instead of redefining it.	329
<code>\initnumbering@quote</code> : <code>\initnumbering@sectcmd</code> becomes <code>\initnumbering@quote</code>	386
<code>\l@advance@parledgroup@beforenormalnotes</code> : Some conde of <code>\l@dumboxmpfoot</code> moved to <code>\l@advance@parledegroupp@beforenormalnotes</code>	330
<code>\newseries@</code> : One endnotes file by series.	278
General: <code>\@makecol</code> , <code>\@reinserts</code> and <code>\@doclearpage</code> are patched instead of begin redefined	297
<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>	297
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	230
Suppress indexing command specific to <code>memoir</code>	336
v2.0.1.	
General: Fix a bug in <code>eledmac-compat</code> option	1
Fix incompatibility between optional argument of <code>\pstart</code> and <code>\numberpstarttrue</code>	1
v2.1.0.	
General: Fix a bug with <code>\advanceline</code> at the beginning of a <code>\pstart</code>	1
Fix a bug with <code>\chapter</code> in optional argument of <code>\pstart</code> in parallel typesetting with <code>scrbook</code>	1
Fix a bug with <code>\eledchapter</code> in parallel typesetting with <code>scrbook</code>	1
Fix a bug with <code>\setline</code> at the beginning of a <code>\pstart</code>	1
Fix spacing bug with <code>\Xhooknote</code> and <code>\hooknoteX</code> when using them to insert text and not to execute code.	1

New tools to number stanzas	1
v2.1.1.	
General: Fix a bug with <code>\ledpbsetting{before}</code>	1
v2.1.2.	
General: Fix a bug with lineation by <code>pstart</code> and <code>tabular</code> environments (added in 2.1.0).	1
v2.1.3.	
<code>\ledsetnormalparstuffX</code> : Replaced <code>\noindent</code> with <code>\parindent</code> set to 0pt.	249
General: <code>\Xhangindent</code> and <code>\hangindentX</code> work now with all the paragraphs in the note.	1
<code>\Xnoindent</code> and <code>\noindentX</code> work now again (broken in 2.0.0).	1
Change some internal code in order to provide compatibility with \TeX release of october 2015	1
Fix a bug which inserted double space before paragraphed familiar notes.	1
Fix a bug with <code>\edindex</code> when using not-Latin characters without UTF-8 engines	1
v2.2.0.	
General: Fix a bug with combination of <code>\onehalfspacing</code> and two columns and three columns notes typeset.	1
Fix a bug with some setting command and optimization option.	1
Fix spurious space with paragraphed critical notes when using $\text{Lua}\TeX$	1
Increase line list version number to ensure compatibility with new options of <code>reledpar</code> package.	1
New setting tools for endnotes: <code>\Xendnumberonlyfirstinline</code> , <code>\Xendnumberonlyfirstintwolines</code> , <code>\Xendsymmlinenumber</code> , <code>\Xendbeforenumber</code> , <code>\Xendafternumber</code> , <code>\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 TeXformat 2015/10/01.	1
v2.2.2.	
General: Fix a bug in <code>\sethangingsymbol</code>	1
Fix a bug with old version of <code>etex</code>	1
v2.3.0.	
General: Disable empty lines as paragraph in stanza.	1
Fix incompatibility of paragraphed footnotes with <code>bidi</code> v17.9 and following.	1
Warning message when using some setting commands inside <code>rightside</code> environment (deprecated behavior)	1
v2.3.1.	
General: Fix spurious space when using optional argument of <code>\stanza</code> (introduced in v2.3.0).	1
v2.4.0.	
<code>\footnoteoptions@</code> : First argument of <code>\footnoteoption@</code> is now mandatory, not optional.	189
General: <code>\Xbhooknote</code> and <code>\bhooknoteX</code> work with notes in columns.	1
Fix a bug of <code>\parindentX</code> and <code>\Xparindent</code> with two columns and three columns notes.	1
Fix a bug with <code>\sameword</code> in right side.	1
Fix spurious space in two columns and three columns notes.	1
Fix spurious space when using optional argument of stanza (introduced in v2.3.0).	1
New hooks: <code>\Xlinerangeseparator</code> and <code>\Xendlinerangeseparator</code>	46

Option <code>linrangesep</code> for critical footnotes and endnotes.	46
v2.4.1.	
General: Fix a bug with <code>\appref</code> and <code>\apprefwithpage</code> (introduced in v2.4.0).	1
Fix a bug with tabular environments when using <code>babel</code> or <code>polyglossia</code> languages that override \TeX <code>\roman</code> command, like Greek language.	1
Fix a bug with tabular environments when using <code>babel</code> or <code>polyglossia</code> languages that override \TeX <code>\roman</code> command, like Greek.	1
v2.5.0.	
<code>\SErefwithpage</code> : Debug <code>\setapprefprefixsingle</code>	315
<code>\edlabel</code> : Fix a bug when calling <code>\edlabel</code> in a footnotes of the right side	302
<code>\l@d@section</code> : <code>\endnotes</code> take six arguments.	255
<code>\printlines</code> : <code>\printlines</code> takes an eighth argument: the line flag	226
<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.	308
General: <code>\apprefwithpage</code> and <code>\appref</code> print double quotation mark when the label was not defined.	1
<code>\apprefwithpage</code> and <code>\appref</code> work with right side crossref.	1
<code>\apprefwithpage</code> works also when <code>noend</code> option is enabled.	1
<code>\appref</code> and <code>\apprefwithpage</code> can take <code>linrangesep</code> optional argument.	1
<code>\edlabel</code> works now in <code>\Xfootnote</code>	1
<code>\lemma</code> can be used even when the <code>nocritical</code> is enabled.	1
Compatibility with new hook and tools of <code>reledpar</code> 2.6.0.	1
Fix spurious vertical space in <code>astanza</code> environment (<code>reledpar</code>)	1
Log now states “There were undefined references” when using wrong references in <code>\edlineref</code> or <code>edpageref</code>	1
New hooks to customize page and line number appearance in endnotes.	1
New hooks: <code>\Xhookgroup</code> and <code>\bhookgroupX</code>	1
New tools to easily make cross-reference to a passage defined by a start and an end line	66
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>	205
<code>\l@d@section</code> : <code>\endnotes</code> take seven arguments.	255
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>	57
New hooks: <code>\Xlemmafont</code> and <code>\Xendlemmafont</code>	52

New setting commands: <code>\setSErefonlypageprefixsingle</code> and <code>\setSErefonlypageprefixmore</code>	1
Warning for duplicate and undefined labels are parsable by latexmk	1
Warning for duplicate labels does not send any more a false line and page number . . .	1
When using <code>hyperref</code> package, add link in familiar footnotes between the footnote marks in the text and the footnote marks in the footnote	1
When using <code>hyperref</code> package, add links for <code>\SEref</code> and <code>related</code> , <code>\appref</code> and <code>related</code>	1
When using <code>hyperref</code> package, add links from critical footnotes and critical endnotes to the line of text they refers	1
v2.7.1.	
General: Debug <code>\Xhookgroup</code> hooks executed on columnar footnotes (moved to a larger group, to take effect).	1
v2.7.2.	
General: <code>\Xhsize</code> and <code>\hsizeX</code> become <code>\Xwidth</code> and <code>\widthX</code>	57
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.	255
General: <code>reledmac</code> cross-referencing can take advantage of <code>xr</code> package.	1
More <code>\edgls</code> ... commands.	1
No indentation for paragraphed notes in <code>ledgroup</code> . Can be changed with <code>\Xparindent</code> and <code>\parindentX</code>	1
v2.8.1.	
General: Warnings for undefined labels are really parsable by latexmk	1
v2.8.2.	
General: Fix a bug concerning indent in a paragraph immediately following a sectioning command (bug NOT fixed on <code>reledpar</code>)	1
Fix a bug with <code>\AtEveryPstart</code> added in version 2.0.0.	1
Fix a bug with vertical space after the between-sectioning command as optional argument of a <code>\pstart</code> and <code>\pstart</code> content	1
v2.9.0.	
General: Allow continuing line numbering between normal text and parallel text, using <code>\pausenumbering</code> and <code>\resumenumbering</code> and the <code>continuousnumberingwithcolumns</code> option.	1
Fix a bug when using <code>\lineneation{page}</code> and <code>\pausenumbering</code> ... <code>\resumenumbering</code>	1
Fix a bug with three- and two-column footnote setting (added in v2.4.0).	1
Fix spurious space inside three-column familiar footnote.	1
Write correct metadata in numbered files when using <code>\pausenumbering</code> ... <code>\resumenumbering</code>	1
v2.9.1.	
General: Fix a bug when notes start with <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 “ <code>minus</code> ” (bug introduced in v. 2.7.0).	1
v2.9.3.	
General: Fix a bug with line number position and reset added by v. 2.9.0	1

v2.10.0.	
\print@lemma: Code refactoring between \parafootfmt, \twocolfootfmt, \threecolfootfmt and \normalfootfmt.	214
General: Add \AtEveryStanza and \AtEveryStopStanza.	1
Fix a bug in \ledlsnotefontsetup and \ledrsnotefontsetup which could not handle \color command properly.	1
More specific error messages.	1
New hooks: \Xwrapcontent, \Xendwrapcontent and \wrapcontentX.	53
New hooks: \Xwraplemma and \Xendwraplemma	52
v2.10.1.	
General: Add nopenalties option.	1
Fix a bug introduced in v. 1.4: not paragraphed critical footnotes could prevent marginal line number from being displayed	1
v2.11.0.	
\do@actions@fixedcode: Add action 1010	173
General: Add new tools to produce an apparatus of manuscripts	1
Fix a bug in \Xparafootsep in parallel typesetting	1
Make \parafootsepX work	1
Prevent \Xtxtbeforenotes hook from causing notes to go beyond the bottom margin	1
v2.12.0.	
General: \preXnotes becomes \Xprenotes (naming convention)	1
Add \hidenumberingonleftpage and \hidenumberingonrightpage	1
Add \toendnotes and related.	1
Add auxdir option.	1
Fix a bug in critical and familiar footnotes when using uppercase letters with accent mark	1
Fix a bug when using \chapter in optional argument of \pstart in parallel typesetting in combination with the noeledsec option.	1
Fix a bug with \ledinnernote and \ledouternote in parallel typesetting	1
Fix a bug with familiar footnote number in optional argument of \pstart or \pend in parallel typesetting	1
Fix spurious vertical space in \chapter when used as optional argument of \pstart in parallel typesetting.	1
Make endnote compatible with \sameword mechanism	1
More accurate message to control the position of \Xfootnote and \applabel in the \LaTeX code	1
v2.13.0.	
General: Version 2.13.0 never existed.	1
v2.13.1.	
General: In critical footnotes, the right side flag is printed only if requested explicitly with \Xlineflag (bug added in v. 2.5.0).	1
v2.13.2.	
General: Fix a bug added in v. 11.2 which could make parallel typesetting not work.	1
v2.13.3.	
General: Makes \Xendafterpagenumbe affecting \Serefwithpage	1
v2.14.0.	
General: Hyperref with the line number inside critical footnotes is correct when using \xxref	1
Some internal changes for new features of reledpar.	1

v2.14.1.	
General: Fix a bug when using <code>\footnoteX</code> in the first argument of <code>\edtext</code> .	1
v2.14.1a.	
General: Fix problematic typos in the handbook.	1
v2.15.0.	
General: Add “byline” arrangement.	1
Fix <code>\Xtxtbeforenotes</code> in <code>ledgroup</code> .	1
v2.15.1.	
General: Fix <code>\edindex</code> in tabular environments.	1
v2.15.2.	
General: Fix a bug with <code>fancyhdr</code> package 3.8 and later.	1
v2.15.3.	
General: Fix a bug with <code>\section</code> in optional argument of <code>\pstart</code> and empty line before <code>\pend</code> (bug added in v2.8.2).	1
Simplification of the sectioning command code.	1
v2.16.0.	
General: <code>\Xdo@feet</code> becomes <code>\do@Xfeet</code>	297
Add <code>\Xendpagenumberonlyfirst</code> , <code>\Xendpagenumberonlyfirstifsingle</code> , <code>\Xendpagenumberonlyfirstintwo</code> , <code>\Xendinplaceofpagenumber</code> and <code>\Xendsympagenum</code> hooks.	1
Add <code>\Xpagelinesep</code> hook.	49
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.	297
New options for <code>\fnpos</code> and <code>\mpfnpos</code> to set a customized order for familiar and critical footnotes.	1
When <code>\edindex</code> is called outside of a <code>\beginnumbering... \endnumbering</code> structure, it is automatically switched to <code>\index</code> , with a warning message.	1
When indexing texts in familiar footnotes with <code>\edtext</code> , refer to the line number where the footnote is called.	1
When indexing texts in sidenotes with <code>\edtext</code> , refer to the line number where the sidenote is called.	1
v2.16.1.	
General: Fix a bug with redefinition of the style of the footnote number (bug added in v2.12.0)	1
v2.16.2.	
General: Error message if <code>footmisc</code> is loaded after <code>reledmac</code> .	1
Fix a bug introduced by v2.16.1 when using non-expandable control sequence, like <code>\normalfont</code> , in the footnote number style.	1
v2.16.3.	
General: Fix a bug with <code>\Seref</code> (bug added in v2.7.0).	1
v2.16.4.	
General: Fix a bug with vertical space before sectioning command in optional argument of <code>\pstart</code> (bug added in v2.15.3).	1

v2.16.5.	
General: Fix potential spurious spaces in endnotes.	1
v2.16.6.	
General: Fix a bug with the line number style in <code>\doennotes</code> when referring to right side line in parallel typesetting.	1
Take into account <code>\linenumberstyle</code> when using <code>\edlineref</code>	1
v2.16.7.	
General: Fix a bug with <code>\msdata</code> when using multiple <code>\beginnumbering...</code> <code>\endnumbering</code>	1
Fix a bug with <code>\numberpstarttrue</code> when using multiple <code>\beginnumbering...</code> <code>\endnumbering</code>	1
v2.16.8.	
General: Fix a bug with <code>\edindex</code> in footnotes in parallel typesetting.	1
v2.17.0.	
General: Add <code>\edglsadd</code> command.	1
Add <code>\setmsdataposition</code> setting.	1
v2.17.1.	
General: Fix spurious space in paragraphed footnotes when using <code>Lua\TeX</code> without using Right-To-Left text.	1
v2.17.2.	
General: Change log message when numbered files still don't exist, in order to improve compatibility with <code>latexmk</code>	1
v2.17.3.	
General: Fix a bug with <code>\doendnotesbysection</code> and <code>\doendnotes</code>	1
v2.17.4.	
General: Fix a bug with <code>\setSErefonlypageprefixsingle</code> and <code>\setSErefonlypageprefixmore</code>	1
v2.17.5.	
General: Fix a bug with <code>\pstartref</code> when referring to the left side in parallel typesetting.	1
v2.18.0.	
General: Fix a bug when using a <code>\edtext</code> in two lines or more in right-to-left typesetting with <code>X\TeX</code>	1
Fix a bug when using both <code>\Xnumberonlyfirstintwolines</code> or <code>\Xnumberonlyfirstinline</code> and <code>\Xparafootsep</code> and <code>\Xsymlinenum</code>	1
v2.18.1.	
General: Fix a bug when using <code>\msdata</code> with <code>Lua\TeX</code> or with the <code>hyperref</code> package.	1
v2.19.0.	
General: Add <code>\footnoteXmark</code> and <code>\footnoteXtext</code> commands.	1
Add better compatibility with the <code>csquotes</code> package when using familiar footnotes.	1
Fix a bug with paragraph indent after sectioning command.	1
v2.20.0.	
General: Add <code>\AtStartEveryStanza</code> , <code>\BeforeEveryStopStanza</code> , <code>\AtEndEveryPend</code> , <code>\AtStartEveryPstart</code>	1
Add second optional argument to <code>\pstart</code> , <code>\pend</code> and <code>\stanza</code>	1
Add starred version of <code>\AtEveryPstart</code> , <code>\AtEveryPend</code> , <code>\AtEveryStanza</code> and <code>\AtEveryStopStanza</code>	1
Add third and fourth optional argument to <code>\newverse</code>	1
Fix a bug when using familiar footnotes in <code>\eledsection</code> and related.	1
Reset font specification at the beginning of familiar footnotes.	1

v2.21.0.	
General: Add the possibility of nested <code>\sameword</code> commands.	1
Fix a bug when using formatting command in the argument of <code>\edindex</code> inside <code>\edtext</code> .	1
Now, as explained in the handbook, a <code>\edindex</code> inside <code>\edtext</code> only creates an index reference to the main text, and not to the critical footnote.	1
v2.22.0.	
General: Add <code>\txtbeforenotesonlyonceX</code> and <code>\Xtxtbeforenotesonlyonce</code> hooks.	1
Add <code>\txtbeforenotesX</code> hook.	1
Fix a bug added in v2.16.0 when using <code>\Xtxtbeforenotes</code> with paragraphed or normal footnotes.	1
Fix a bug with three and two columns critical footnotes, broken in v. 2.17.6.	1
v2.22.1.	
General: Compatibility with new version of <code>reledpar</code> .	1
Fix a bug with some commands inside <code>\sameword</code> .	1
v2.22.2.	
General: Do not print footnotes at the first run.	1
v2.23.0.	
General: Add <code>swcaseinsensitive</code> option	1
Optimisation of the code added on v2.22.2 to not print footnotes at the first run.	1
v2.24.0.	
<code>\next@line@list@stuff</code> : Add <code>\next@line@list@stuff</code> internal hook.	133
General: Add <code>\swnoexpands</code> macro to avoid problems with not fully expandable macro inside <code>\sameword</code> .	1
Fix a bug with <code>\edlabel</code> at the beginning of a <code>\pstart</code> .	1
Fix spurious space with <code>\labelpstarttrue</code>	1
When a <code>\setlinenum</code> is used, it is stronger than the <code>\lineation{page}</code> setting.	1
v2.24.1.	
General: Compatibility with <code>reledpar 2.20.2</code>	1
v2.24.2.	
General: Fix bug between tabular environments and endnotes	1
v2.25.0.	
General: Add <code>\Xendtxtbeforenotes</code> hook	1
v2.26.0.	
General: Add a mechanism for annotation of stanza line numbering	1
Fix bug in <code>\xpstartref</code>	1
v2.26.1.	
General: Fix bug with <code>\linenumannotation</code>	1
v2.26.10.	
General: Add error message when insertion a <code>\pstart</code> or <code>\pend</code> inside <code>\edtext</code> arguments	1
Really fix bug with page number in footnote after <code>\mainmatter</code> (bug introduced in v2.26.7)	1
v2.26.11.	
General: Allow to use <code>\edtext</code> inside <code>\sameword</code>	1
v2.26.12.	
General: Precise minimal version of <code>xparse</code> package required.	1
v2.26.2.	
General: Fix bug with <code>\linenumannotation</code> in parallel typesetting	1
Fix bug with <code>\Xgroupbyline</code> and <code>\Xparafootsep</code> .	1

Fix bug with <code>\Xgroupbyline</code> and optional argument of critical footnotes.	1
Fix bug with <code>\Xgroupbyline</code> for the series after the first one.	1
v2.26.3.	
General: Fix bug added by v2.26.0 with notes in parallel typesetting	1
v2.26.4.	
General: Fix bug when using <code>\numberpstarttrue</code> and alternating between manual <code>\pstart</code> and automatic <code>\pstart</code> produced by <code>\autopar</code>	1
v2.26.5.	
General: Check that users have a enough recent version of <code>xparse</code>	1
Fix bug when using <code>\eledsection</code> with the <code>continuousnumberingwithcolumns</code> of <code>reledpar</code>	1
Fix bug with <code>continuousnumberingwithcolumns</code>	1
Fix bug with using <code>\eledsection</code> and related when alternate normal typesetting and parallel typesetting	1
v2.26.6.	
General: Fix bug with <code>\lineation</code> by page and nexted <code>\edtext</code> between two pages (added in v. 2.24.0)	1
v2.26.7.	
General: Fix page number in footnote and endnotes when using <code>sameparallelpagenumber</code> option of <code>reledpar</code> and typesetting critical text in single mode after critical text in double page mode	1
v2.26.8.	
General: Fix bug in some specific cases, introduced by v.2.27.7	1
v2.26.9.	
General: Fix bug with critical footnotes line number when alternating between <code>\numberlinefalse</code> and <code>\numberlinetrue</code>	1
Fix bug with page number in footnote after <code>\mainmatter</code> (bug introduced in v2.26.7)	1
Workaround with a bug of L ^A T _E X 2018/04/01 patch level 3 to generate the .sty file from a .ins file	1
v2.27.0.	
General: Add <code>\doinsidethislinehook</code>	1
v2.27.1.	
General: Fix spurious space in <code>\hidenumbering</code>	1
v2.28.0.	
General: Familiar footnotes can take an optional argument to manually define the footnote mark.	1
v2.28.1.	
General: Fix bug with the alignment of the columns of two and three columns critical footnotes.	1
v2.29.0.	
General: Add <code>noresetlinenumannotation</code>	1
Fix bug with <code>\linenumannotation</code> and <code>nofamiliar</code> option.	1
Fix little bug with <code>\msdata</code>	1
Really reset line number annotation at each line	1
v2.30.0.	
General: Add <code>\Xnoidenticallylinenumannotation</code> and <code>\Xendnoidenticallylinenumannotation</code> hooks	1
Add separator between line number annotations in margin. Add <code>\setlinenumannotationsep</code>	1

v2.31.0.	
General: Add <code>noprefix</code> and <code>prefixmore</code> to <code>\appref</code> and <code>\SEref</code> .	1
v2.31.1.	
General: Fix bug in vertical spacing with <code>\Xgroupbyline</code> .	1
Fix bug with <code>\Xgroupbyline</code> , which could produce spurious blank lines.	1
Fix bug with <code>\Xhangindent</code> and <code>\hangindentX</code> for right-to-left texts.	1
Fix bug with indentation with <code>\Xgroupbyline</code> .	1
Fix bug with multiple footnote layer and <code>\Xgroupbyline</code> .	1
Fix incompatibility between <code>\Xgroupbyline</code> and <code>hyperref</code> package.	1
Fix incompatibility between <code>\Xgroupbyline</code> and optional argument of critical footnotes.	1
v2.31.2.	
General: Fix bug with familiar footnote containing 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 \TeX .	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- $\mathsf{X}\mathsf{f}\mathsf{L}\mathsf{a}\mathsf{T}\mathsf{E}\mathsf{X}$ engines.	1
Fix bug with page number in index with <code>\edindex</code> when indexing a critical notes.	1
v2.32.6.	
General: Add <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.	
\ledinnotemark: \ledinnotemark redefinition is deprecated, use \Xinnotemark and \innotemarkX instead.	335
\newhookcommand@series: \newhookcommand@series can take an second optional argument (third argument).	285
General: Add \Xinnotemark and \innotemarkX hooks.	1
Add antilabe feature.	1
Add startstanzaindentsrepetition feature.	1
v2.33.1.	
General: Fix a spacing bug when the argument of \antilabe contains more than one word.	1
v2.34.0.	
General: Add \Xpstartonlyfirst and \Xstanzaonlyfirst.	1
Add \Xpstartseparator.	1
v2.34.1.	
General: Fix bug with page number when using stanazindentsrepetition, 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 \parafootsepX when the footnotes are called outside of a numbered section.	1
Fix incompatibility between lineation by \pstart and (a)stanza.	1
Make commands like \hidenumbering working at the very beginning of a verse when the first argument of \setstanzaindents is equal to 0.	1
v2.34.4.	
General: Fix bugs with \parafootsepX 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 \apprefwithpage / \Serefwithpage 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 crossreferecing in parallel typesetting (added in v2.34.4)	1
Fix bug with sublineation on right side of parallel typesetting.	1