

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	11
1.1 Aim of the package	11
1.2 History	12
1.2.1 edmac	12
1.2.2 ledmac	14
1.2.3 eledmac	14

*This file (`reledmac.dtx`) has version number v2.26.0, last revised 2017/10/21.

[†]maieul at maieul dot net

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

6.3.2 Case setting	29
6.3.3 Notes about input encoding with UTF-8 processor	29
6.3.4 Use with <code>\lemma</code> command	30
6.3.5 Sameword for a group of words	31
6.3.6 Customizing	32
6.3.7 Problems with some macros	32
6.3.8 Automatic sameword annotation	33
6.4 Apparatus of manuscripts	33
6.4.1 Marking sections of text	34
6.4.2 Layout of the apparatus of manuscripts	34
6.4.3 Settings	35
6.5 Familiar notes	35
6.5.1 Basic use	35
6.5.2 Customizing mark	35
6.5.3 Separator for multiple footnotes	35
6.6 Printing the footnote mark without printing the footnote text	36
6.7 Changing series	36
6.7.1 Create a new series	36
6.7.2 Delete series	36
6.7.3 Series order	36
6.8 Position of critical and familiar footnotes	37
7 Apparatus customization	37
7.1 Notes arrangement in a series	38
7.2 Control line number printing	39
7.2.1 Print line number only at first time	39
7.2.2 Print page number only at first time	39
7.2.3 Arbitrary text before line number	40
7.2.4 Separator for line range	40
7.2.5 Abbreviate line range	40
7.2.6 Disable line number	41
7.2.7 Printing pstart number	41
7.2.8 Printing stanza number	41
7.2.9 Separator between line and subline numbers	42
7.2.10 Separator between page and line numbers	42
7.2.11 Space around number	42
7.2.12 Space around line symbol	42
7.2.13 Space in place of number	42
7.2.14 Boxing line number and line symbol	43
7.3 Arbitrary code around line number	44
7.4 Separator between the lemma and the note	44
7.4.1 For footnotes	44
7.4.2 For endnotes	44
7.5 Font style	45
7.5.1 For line number	45
7.5.2 For the lemma	45

7.5.3 For all notes	45
7.6 Wrapping notes	46
7.6.1 Wrapping lemmas	46
7.6.2 Wrapping contents	46
7.7 Indent of notes content	46
7.8 Arbitrary code at the beginning of notes	46
7.9 Arbitrary code before inserting note	47
7.10 Options for footnotes in columns	47
7.10.1 Alignment	47
7.10.2 Size of the columns	47
7.11 Options for paragraphed footnotes and notes grouped by line	48
7.11.1 Mark separation of notes	48
7.11.2 Ragged text	48
7.12 Options for block of notes	48
7.12.1 Grouping notes by line	48
7.12.2 Text before notes	48
7.12.3 Code before notes	49
7.12.4 Spacing	49
7.12.5 Rule	49
7.12.6 Maximum height	49
7.12.7 Width	50
7.13 Footnotes and the <code>reledpar</code> columns	50
7.14 Line number annotation	50
7.15 Endnotes in one paragraph	50
8 Fonts	51
9 Verse	51
9.1 Basic	51
9.2 Define stanza indents	51
9.3 Repeating stanza indents	52
9.4 Manual stanza indent	53
9.5 Stanza breaking	53
9.6 Hanging symbol	53
9.7 Long verse and page break	54
9.8 Content before/after verses	54
9.9 Numbering stanza	55
9.10 Add annotation to line of verse numbers	55
9.10.1 Modify annotation associated to a specific note	56
9.10.2 Changing the position of the annotation	57
9.10.3 Changing macro in which annotations are wrapped	57
9.11 Various tools	58
9.12 Notes on empty lines	58
10 Grouping	58

11 Cross referencing	59
11.1 Basic use	59
11.2 Cross-referencing to a critical note	59
11.3 Cross-referencing which return a number in any case	60
11.3.1 Cross-referencing in order to define line number of a critical note	60
11.4 Not automatic cross-referencing	61
11.5 Normal L ^A T _E X cross-referencing	61
11.6 References to start and end lines	61
11.6.1 Reference to main text lines	61
11.6.2 References to lines that are commented on in the apparatus	61
11.6.3 Settings	62
11.7 Compatibility with xr package	63
12 Side notes	64
12.1 Basics	64
12.2 Setting	64
12.2.1 Width	64
12.2.2 Vertical position	64
12.2.3 Distance to the main text	64
12.2.4 Font	64
12.2.5 Separator between notes	65
13 Indexing	65
13.1 Basics	65
13.2 Referring to critical notes	65
13.3 Separator between page and line numbers	66
13.4 Using xindy	66
13.5 Advanced setting	67
14 Glossary	67
14.1 Preamble setting	67
14.2 Commands	67
15 Tabular material	68
16 Sectioning commands	71
16.1 Sectioning commands without line numbers or critical notes	71
16.2 Sectioning commands with line numbering and critical notes	71
16.3 Optimization	72
17 Quotation environments	72
18 Page breaks	72
18.1 Control page breaking	72
18.2 Prevent page break in a long verses	73

19 Miscellaneous	73
19.1 Known and suspected limitations	74
19.1.1 Non-standard geometry	74
19.1.2 floatrow package compatibility	74
19.1.3 ‘No room for a new’	74
19.1.4 Marginal notes	74
19.1.5 Paragraph shape	75
19.1.6 Paragraphed footnotes	75
19.1.7 Use with other packages	75
19.2 Parallel typesetting	76
I Implementation overview	77
II Preliminaries	77
II.1 Links with original edmac	77
II.2 Package declaration	77
II.3 Package options	78
II.4 Loading packages	80
II.5 Compatibility with LuaTeX	80
II.6 Boolean flags	81
II.7 Messages	82
II.8 Gobbling	88
II.9 Wrapping content	88
II.10 Miscellaneous commands	88
II.11 Prepare reledpar	89
II.12 Booleans provided by other optional packages which are required in any case	90
III Sectioning commands	90
IV List macros	95
V Line counting	96
V.1 Choosing the system of lineation	96
V.2 Line number margin	98
V.3 Line number initialization and increment	99
V.4 Line number locking	100
V.5 Line number style	101
V.6 Line number printing	102
V.7 Line number counters and lists	103
V.8 Line number locking counter	104
V.9 Line number associated to lemma	104
V.10 Reading the line-list file	108
V.11 Commands within the line-list file	110
V.12 Writing to the line-list file	123

VI Marking text for notes	130
VI.1 <code>\edtext</code> itself	131
VI.2 Substitute lemma	138
VI.3 Substitute line numbers	139
VI.4 Lemma disambiguation	140
VII Paragraph decomposition and reassembly	147
VII.1 Boxes, counters, <code>\pstart</code> and <code>\pend</code>	147
VII.2 Processing one line	153
VII.2.1 General process	153
VII.2.2 Process for “normal” line	154
VII.2.3 Process for line containing <code>\eledsection</code> command	156
VII.2.4 Hooks	157
VII.2.5 Sidenotes and marginal line number initialization	157
VIII Line and page number computation	158
IX Line number annotation	162
X Line number printing	164
XI Pstart number printing in side	168
XII Restoring footnotes and penalties	169
XII.1 Add insertions to the vertical list	169
XII.2 Penalties	171
XII.3 Printing leftover notes	172
XII.4 Text before notes	173
XIII Critical footnotes	175
XIII.1 Fonts	175
XIII.2 Individual note options	175
XIII.3 Notes language	176
XIII.4 General survey of the way we manage notes	177
XIII.5 General setup	177
XIII.6 Footnotes arrangement	178
XIII.6.1 User level macro	178
XIII.6.2 Normal footnote	179
XIII.6.3 Paragraphed footnotes	184
XIII.6.4 Columnar footnotes	191
XIII.7 Critical notes presentation	198
XIII.7.1 Font tools	198
XIII.7.2 Pstart number in footnote	199
XIII.7.3 Lemma printing	199
XIII.7.4 Line number printing	200
XIII.7.5 Footnote grouped by line	210

XIV Familiar footnotes	211
XIV.1 Adjacent footnotes	211
XIV.2 Regular footnotes for numbered texts	213
XIV.3 Footnote formats	215
XIV.4 Footnote arrangement	215
XIV.4.1 User level macro	215
XIV.4.2 Normal footnotes	216
XIV.4.3 Two columns footnotes	221
XIV.4.4 Three columns footnotes	224
XIV.4.5 Paragraphed footnotes	226
XIV.5 Wrapping footnote marks in hyperlink	230
XV Code common to both critical and familiar footnote in normal arrangement	231
XVI Footnotes' width for two columns	232
XVII Footnotes' order	233
XVIII Footnotes' rule	233
XIX Specific skip for first series of footnotes	234
XIX.0.1 Overview	234
XIX.0.2 User level command	234
XIX.0.3 Internal commands	235
XX Endnotes	236
XX.1 Internal commands	236
XX.2 User level commands	240
XX.2.1 Inserting contents to endnotes	240
XX.2.2 Printing endnotes	241
XXI Generate series of notes	248
XXI.1 Test if series is still existing	249
XXI.2 Init specific to reledpar	249
XXI.3 For critical footnotes	249
XXI.3.1 Options	249
XXI.3.2 Create inserts, needed to add notes in foot	251
XXI.3.3 Create commands for critical apparatus, \Afootnote, \Bfootnote etc.	251
XXI.3.4 Set standard display	254
XXI.4 For familiar footnotes	254
XXI.4.1 Options	254
XXI.4.2 Create tools for familiar footnotes (\footnoteX)	255
XXI.5 The endnotes	257
XXI.5.1 The auxiliary file	257
XXI.5.2 The main macro	257

XXI.5.3 Tools	258
XXI.5.4 Internal commands	258
XXI.5.5 The options	259
XXI.6 Init standards series (A,B,C,D,E)	260
XXII Setting series display	261
XXII.1 Change series order	261
XXII.2 Test series order	261
XXII.2.1 Get the first series	261
XXII.3 Series setting	262
XXII.3.1 General way of working	262
XXII.3.2 Tools to set options	262
XXII.3.3 Tools to generate options commands	264
XXII.3.4 Options for critical notes	265
XXII.3.5 Options for familiar notes	267
XXII.3.6 Options for endnotes	267
XXII.4 Hooks for a particular footnote	269
XXII.5 Alias	270
XXIII Output routine	270
XXIII.1 Extra footnotes output	270
XXIII.2 Patching standard output's commands	274
XXIV Cross referencing	276
XXIV.1 Compatibility with xref	292
XXV Side notes	293
XXVI Minipages and such	301
XXVII Indexing	306
XXVII.1 Looking on package order	306
XXVII.2 Auxiliary macros for <code>\edindex</code>	306
XXVII.3 Code specific to <code>\edindexin</code> critical footnotes	307
XXVII.4 Analysis of command in indexed text	308
XXVII.5 Code for the formatted index	309
XXVII.6 Main code	309
XXVII.7 Hyperlink	311
XXVII.8 'innote' and 'notenumber' option of <code>indextols</code> package	314
XXVIII Glossaries	315
XXIX Verse	317
XXIX.1 Hanging symbol management	317
XXIX.2 Using <code>&</code> character	317
XXIX.3 Code category setting	318
XXIX.4 Stanza count and indent	318

XXIX.5 Numbering stanza	319
XXIX.6 Stanza number in note	321
XXIX.7 Main work	321
XXIX.8 Restore catcode and penalties	324
XXX Apparatus of Manuscripts	324
XXX.1 User level macro	324
XXX.2 Setting macro	326
XXX.3 Counters and lists	326
XXX.4 Auxiliary file macros	327
XXX.5 Action macro	328
XXX.6 Inserting footnote	333
XXX.7 Other	333
XXXI Arrays and tables	333
XXXI.1 Preamble: macro as environment	333
XXXI.2 Tabular environments	337
XXXI.2.1 Disabling and restoring commands	337
XXXI.2.2 Counters, boxes and lengths	341
XXXI.2.3 Tabular typesetting	344
XXXI.2.4 Environments	356
XXXII Quotation's commands	356
XXXIII Section's title commands	357
XXXIII.1 Commands to disable some feature	357
XXXIII.2 General overview	357
XXXIII.3 <code>\beforeeledchapter</code> command	358
XXXIII.4 Auxiliary commands	359
XXXIII.5 Patching standard commands	359
XXXIII.6 Main code of <code>\eledxxx</code> commands	364
XXXIII.7 Macros written in the auxiliary file	367
XXXIV Page breaking or no page breaking depending of specific lines	369
XXXV Long verse: prevents being separated by a page break	371
XXXVI Tools for hyperref package	372
XXXVII Compatibility with eledmac	372
Appendix A Things to do when changing versions	375
Appendix A.1 Migrating from edmac to ledmac	375
Appendix A.2 Migration from ledmac to eledmac	376
Appendix A.3 Migration to eledmac 1.5.1	377
Appendix A.4 Migration to eledmac 1.12.0	377
Appendix A.5 Migration to eledmac 17.1	378

Appendix A.6 Migration to eledmac 1.21.0	378
Appendix A.6.1 <code>\Xledsetnormalparstuff</code> and <code>\ledsetnormalparstuff</code>	378
Appendix A.6.2 Endnotes	378
Appendix A.7 Migration to eledmac 1.22.0	378
Appendix A.8 Migration to eledmac 1.23.0	378
Appendix A.9 Migration from eledmac to reledmac	379
Appendix A.9.1 Risk of ‘no room for a new’	379
Appendix A.9.2 Multiple indices with memoir	379
Appendix A.9.3 Deprecated commands and options	379
Appendix A.9.4 <code>\renewcommand</code> replaced by <code>command</code>	380
Appendix A.9.5 Commands the names of which have been changed	380
Appendix A.9.6 Endnotes	382
Appendix A.9.7 Z Series	382
Appendix A.9.8 Internal commands	382
Appendix A.10 Migration to reledmac 2.1.0	382
Appendix A.11 Migration to reledmac 2.1.3	382
Appendix A.12 Migration to reledmac 2.3.0	382
Appendix A.13 Migration to reledmac 2.4.0	383
Appendix A.14 Migration to reledmac 2.5.0	383
Appendix A.15 Migration to reledmac 2.7.0	383
Appendix A.16 Migration to reledmac 2.7.2	383
Appendix A.17 Migration to reledmac 2.8.0	383
Appendix A.18 Migration to reledmac 2.13.1	383
Appendix A.19 Migration to reledmac 2.18.0	384
Appendix A.20 Migration to reledmac 2.21.0	384
Appendix A.21 Migration to reledmac 2.24.0	384
Appendix A.22 Migration to reledmac 2.26.0	384
References	385
Index	385
Change History	433

1 Introduction

1.1 Aim of the package

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

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

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

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

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

Any suggestions for new features are welcome.

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

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

1.2 History

1.2.1 `edmac`

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

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

The present version was developed in the summer of 1990, with the intent of adding necessary features, streamlining and documenting the code, and further generalizing it to make it easily adaptable to the needs of editors in different disciplines. John did most of the general reworking and documentation, with the financial assistance of the Division of the Humanities and Social Sciences, California Institute of Technology. Dominik adapted the code to the conventions of Frank Mittelbach’s `doc` option, and added some

documentation, multiple-column footnotes, cross-references, and crop marks.¹ A description by John and Dominik of this version of `edmac` was published as ‘An overview of `edmac`: a PLAIN \TeX format for critical editions’, *TUGboat* 11 (1990), pp. 623–643.

From 1991 through 1994, the macros continued to evolve, and were tested at a number of sites. We are very grateful to all the members of the (now defunct) `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 Quinqueecclesiensi in Regno Ungarie*,¹⁰ the collected letters and papers of Leibniz,¹¹ Theodosius’s *Spherics*, the German *Algorismus* of Sacrobosco, the Sanskrit text of the *Kāśikāvṛtti* of Vāmana and Jayāditya,¹² and the English texts of Thomas Middleton’s collected works.

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).

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: Schiphower en Brinkman, 1993).

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

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

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

¹²Being prepared at Poona and Lausanne Universities.

1.2.2 ledmac

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

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

In March 2003 Peter Wilson started an attempt to port `edmac` from TeX to LaTeX. The starting point was `edmac` version 3.16 as documented on 19 July 1994 (available from CTAN). In August 2003 the `tabmac` functions were added; the starting point for these being version 1.0 of October 1996. The `edstanza` (v0.01) functions were added in February 2004. Sidenotes and regular footnotes in numbered text were added in April 2004. This port was called `ledmac` (L^AT_EX `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. 37). 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 Appendix A.2 p. 376.

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 Appendix A.9 p. 379.

1.3 Bibliography

A collaborative list of works edited with (r)(e)`ledmac` is available at https://www.zotero.org/groups/critical_editions_typeset_with_edmac_ledmac_and_eledmac/items. Please add your own edition made with (r)(e)`ledmac`.

If you write book or article about (r)(e)`ledmac`, please add it on the collaborative bibliography on https://www.zotero.org/groups/articles_and_books_about_reledmac/items.

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

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

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

3 Compatibility warning

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

Please read 7.12.6 p. 49.

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

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

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

4 Options

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

4.1 Specific features

draft underlines lemmas in the main text.

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

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

eledmac-compat help to migrate from eledmac to reledmac (see Appendix A.9.5 p. 380).

swcaseinsensitive make `\sameword` command case insensitive.

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

nopbinverse prevents page break within verse environment.

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

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

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

xindy and `xindy+hyperref` select xindy as the index processor (13.4 p. 66).

4.2 Optimizing package performance

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

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

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

nofamiliar disables tools for familiar footnotes (`\footnoteA`, `\footnoteB` etc.). If you do not need familiar footnotes, this option lets eledmac 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. 58). 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 scope of the `\autopar` command needs to be limited by keeping it within a group, as follows:

```

\begingroup
\beginnumbering
\autopar

A paragraph of numbered text.

Another paragraph of numbered
text.

\endnumbering
\endgroup

```

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

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. 51) or `reledpar` (see 19.2 p. 76).

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

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

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

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

`\AtStartEveryPstart` `\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`)

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 inside. In this case, the line number will be not printed.

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

5.2.6 Languages written in Right to Left

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

5.2.7 Memory limits

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

`\pausenumbering`
`\resumenumbering`

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

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

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

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

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

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

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

5.3 Lineation commands

5.3.1 Disabling lineation

`\numberlinefalse`
`\numberlinetrue`

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

5.3.2 Setting lineation start and step

`\firstlinenum`
`\linenumincrement`

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

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

`\firstsublinenum`
`\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 (7.2.9 p. 42) or by using `Xsublinesepside`. In the second case, it will change the separator only for line numbers in the margins, not in the footnotes.

5.4.2 Locking lineation

`\startlock` The `\startlock` command, used in running text, locks the line number at its current
`\endlock` value, until you insert `\endlock`. It can tell for itself whether you are in a patch of line or sub-line numbering. One use for line-number locking is in printing poetry: there the line numbers should be those of verse lines rather than of printed lines, even when a verse line requires several printed lines. But in this case you may use the `\stanza` mechanism, see 9 p. 51.

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

5.4.3 Setting and changing line number

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

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

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

5.4.4 Line number style

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

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

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

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

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

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

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

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

5.4.5 Skipping and hiding number

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

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

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

5.5 Executing code at each line

`\dolinehook`
`\doinsidelinehook`

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

6 Apparatus commands

6.1 Terminology

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

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

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

So :

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

6.2 Critical notes

6.2.1 The lemma

`\edtext`

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

```
\edtext{⟨lemma⟩}{⟨commands⟩}
```

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

For example:

I am happy :	1	I am happy : I saw my friend Smith on
I saw my friend <code>\edtext{Smith}{</code>	2	Tuesday.
<code>\Afootnote{Jones C, D.}}</code>		
on Tuesday.		_____
		1 Smith] Jones C, D.

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

The $\langle lemma \rangle$ may contain further $\backslash edtext$ 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:

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

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

6.2.2 Footnotes

The second argument of the $\backslash edtext$ macro, $\langle commands \rangle$, may contain a series of subsidiary commands that generate various kinds of notes.

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

If you need more series of critical notes, please look at 6.7.1 p. 36.

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 $\backslash Xtwolines$ and $\backslash Xmorethantwolines$ features for this note (cf. 7.2.5 p. 40).
- `nonum` disables line numbering for this note. A horizontal blank space is added instead. You can use $\backslash Xinplaceoflemmaseparator$ to set it (7.4.1 p. 44).
- `nosep` to disable the lemma separator for this note.
- `linerangesep=\langle c \rangle` to change to $\langle c \rangle$ the separator between start line and end line for this particular note.

Example: $\backslash Afootnote[\textit{nonum}]\{\langle text \rangle\}$.

6.2.3 Endnotes

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

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

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

$\backslash Aendnote[\langle option \rangle]\{\langle text \rangle\}$.

$\langle option \rangle$ can contain a comma-separated list of values. Allowed values are:

- `fulllines` to disable `\xendtwolines` and `\xendmoreethantwolines` features for this particular note (cf. 7.2.5 p. 40).
- `nonum` to disable line number for this particular note.
- `nosep` to disable the lemma separator for this particular note. A horizontal blank space is added instead. You can use `\xendinplaceoflemmaseparator` to set it (7.4.2 p. 45).
- `linrangesep= $\langle c \rangle$` to change to $\langle c \rangle$ 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 $\langle s \rangle$` , where $\langle s \rangle$ 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 $\langle s \rangle$ 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 $\langle s \rangle$` . For each value of $\langle s \rangle$, the first call of the command will print the notes for the first series, the second call will print the notes for the second series etc. For example, do:

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

Note that by default inside endnotes no separator is used between the lemma and the content. However you can use the `\xendlemmaseparator` macro to define one (7.4.2 p. 44).

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

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

`\toendnotes [$\langle series \rangle$] { $\langle content \rangle$ }` where $\langle series \rangle$ is a comma-separated list of the series of endnotes where $\langle content \rangle$ must be inserted. If $\langle series \rangle$ is empty, then $\langle content \rangle$ 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-expanded¹⁴ version of the `<content>`, that is the version where the commands are expanded on the first level. This can be, for example, to get a counter value. Use the starred version in this case. For example:

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

6.2.4 Paragraph in critical apparatus

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

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

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

6.2.5 Change lemma and line number

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

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

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

¹⁴The 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.

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

However, you can retain the value computed by `reledmac` for any number by simply omitting it; and you can omit a sequence of vertical bars at the end of the argument. For example, `\linenum{|||23}` changes only the ending page number of the current lemma.

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

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

6.2.6 Changing the names of commands for critical apparatus

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

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

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

¹⁵We 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

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 $X_{\text{E}}\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ or $\text{L}^{\text{u}}\text{a}\text{T}_{\text{E}}\text{X}$, there should not be any glitches. However, pay attention to how characters are encoded. Similar-looking characters may be represented differently in unicode numbering.

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

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

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

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

- If you use $X_{\text{E}}\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$, add this line in your preamble: `\XeTeXinputnormalization 1`.
- If you use $\text{L}^{\text{u}}\text{a}\text{T}_{\text{E}}\text{X}$, use the `uninormalize` package of Michal Hoftich¹⁶ with the `buffer` option set to true.

With these tools, $X_{\text{E}}\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ / $\text{L}^{\text{u}}\text{a}\text{T}_{\text{E}}\text{X}$ will dynamically normalize unicode input when reading the file. Consequently, you will have no problems with the `\sameword` command.

handbook of `xargs` to know more about `\newcommandx`.

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

6.3.4 Use with `\lemma` command

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

For example, in the following example:

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

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

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

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

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

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

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

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

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

<div style="display: flex; align-items: center; justify-content: center;"> 1_{inlemma} <div style="border: 1px solid black; padding: 2px 5px;">2</div> 3₂ </div>	1...3	4	5	1 ₁	1...5
--	-------	---	---	----------------	-------

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

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

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

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

Here, the corresponding code:

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

1	A (1) A (2) A (3) A (4) A (5)	
	1 A ¹ ...A ⁵] level 1	
	1 A ¹ ...A ³] level 2	
	1 A ² (2)] level 3	

6.3.5 Sameword for a group of words

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

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

```
\sameword{per} \sameword{causam}
tamen scire
\edtext{\sameword{causam}}{\Bfootnote{fnote}}
est
```

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

1 per causam tamen scire causam est per causam cognoscere causam

```
1 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
```

6.3.6 Customizing

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

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

6.3.7 Problems with some macros

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


```

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

```

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

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

```

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

```

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

```

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

```

6.3.8 Automatic sameword annotation

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

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

6.4 Apparatus of manuscripts

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

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

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

6.4.1 Marking sections of text

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

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

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

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

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

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

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

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

6.4.2 Layout of the apparatus of manuscripts

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

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

6.4.3 Settings

As the apparatus of manuscripts technically consists of first-level critical notes ('A' series), any setting available for critical notes can be applied (7 p. 37). However, the following *additional* commands are available.

<code>\setmsdataserie</code>	The series used by default for the apparatus of manuscripts is series A. However, you can change it with <code>\setmsdataserie{<series>}</code> .
<code>\setmsdatalabel</code>	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 <code>\setmsdatalabel{<txt>}</code> .
<code>\setmsdataposition</code>	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: <code>\msdataposition{regular-msdata}</code> And restore the default behaviour using <code>\msdataposition{msdata-regular}</code>

6.5 Familiar notes

6.5.1 Basic use

<code>\footnoteA</code>	As well as the standard L ^A T _E X footnotes generated via <code>\footnote</code> , the package also provides five series of additional footnotes called <code>\footnoteA</code> through <code>\footnoteE</code> . 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.
<code>\footnoteB</code>	
<code>\footnoteC</code>	
<code>\footnoteD</code>	
<code>\footnoteE</code>	

6.5.2 Customizing mark

<code>\thefootnoteA</code>	Each series uses a set of macros for styling the marks. The mark numbering scheme of series A is defined by the <code>\thefootnoteA</code> macro; the default is: <code>\renewcommand*{\thefootnoteA}{\arabic{footnoteA}}</code>
<code>\bodyfootmarkA</code>	
<code>\footfootmarkA</code>	
	The appearance of the mark in the text is controlled by <code>\bodyfootmarkA</code> which is defined as: <code>\newcommand*{\bodyfootmarkA}{%</code> <code>\hbox{\textsuperscript{\normalfont\@nameuse{@thefnmarkA}}}}</code>
	The command <code>\footfootmarkA</code> controls the appearance of the mark at the start of the footnote text. It is defined as: <code>\newcommand*{\footfootmarkA}{\textsuperscript{\@nameuse{@thefnmarkA}}}</code>
	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.

<code>\multfootsep</code>	<code>\multfootsep</code> is used as the separator between footnote markers. Its default definition is:
---------------------------	---

`\providecommand*{\multfootsep}{\normalfont,}`
and can be changed if necessary.

6.6 Printing the footnote mark without printing the footnote text

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

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

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

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

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

6.7 Changing series

6.7.1 Create a new series

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

6.7.2 Delete series

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

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

6.7.3 Series order

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

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

6.8 Position of critical and familiar footnotes

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

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

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

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

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

`<series><type>`

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

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

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

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

7 Apparatus customization

Some commands can be used to change the display of the footnotes. All can have an optional argument [`<s>`], which is the letter of the series — or a list of letters separated by comma — depending on which option is applied. If the optional argument is omitted or empty, the setting will apply to the entire series.

When a length, noted $\langle l \rangle$, is used, it can be stretchable: `a plus b minus c`. The final length m is calculated by \TeX to have: $a - c \leq m \leq a + b$. If you use some relative unit¹⁷, 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.

7.1 Notes arrangement in a series

`\Xarrangement`
`\arrangementX`

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

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

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

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

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

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

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

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

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

7.2 Control line number printing

7.2.1 Print line number only at first time

`\Xnumberonlyfirstinline`

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

`\Xnumberonlyfirstintwolines`

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

`\Xsymmlinenum`

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

`\Xendnumberonlyfirstinline`

`\Xendnumberonlyfirstintwolines`

`\Xendsymmlinenum`

For endnotes, `\Xendnumberonlyfirstinline`; `\Xendnumberonlyfirstintwolines` and `\Xendsymmlinenum` are the equivalents of `\Xnumberonlyfirstinline`; `\Xnumberonlyfirstintwolines` and `\Xsymmlinenum`.

7.2.2 Print page number only at first time

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

`\Xendpagenumberonlyfirst`

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

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

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

the starting page number and the ending page number of the preceding lemma respectively.

In any case, you can use:

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

7.2.3 Arbitrary text before line number

- `\Xbeforenumber` `\Xbeforenumber[⟨s⟩]{⟨txt⟩}` allow to insert `⟨txt⟩` before the line number, only when the line number is printed, so taking into account `\Xnumberonlyfirstinline` and similar.
- `\Xendbeforepagenumber` **For endnotes** `\Xendbeforepagenumber[⟨s⟩]{⟨text⟩}` defines the text before the page number in endnotes. Default value is `p.` (“p” followed by a dot).
- `\Xendafterpagenumber` `\Xendafterpagenumber[⟨s⟩]{⟨text⟩}` defines the text after the page number in endnotes. Default value is `)` (open parenthesis followed by a single space).
- `\Xendlineprefixsingle` `\Xendlineprefixsingle[⟨s⟩]` defines the text before the line number in endnotes, when there is only one line. Default value is empty.
- `\Xendlineprefixmore` `\Xendlineprefixmore[⟨s⟩]{⟨text⟩}` defines the text before the line number in endnotes, when there is more than one line. Default value is empty. If you don’t define it, use the value defined by `\Xendlineprefixsingle`.

7.2.4 Separator for line range

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

7.2.5 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 `<text>` argument of `\Xtwolines` will be used for lemmas which fall on three or more lines.

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

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

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

You can disable `\Xtwolines` and related for a specific note by using the ‘[fulllines]’ argument in the note macro cf. 6.2.2 p. 25.

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

7.2.6 Disable line number

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

7.2.7 Printing pstart number

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

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

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

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

7.2.8 Printing stanza number

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

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

`\Xstanzaseparator`

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

7.2.9 Separator between line and subline numbers

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

Employed without optional argument, it also change separator in side number.

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

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

7.2.10 Separator between page and line numbers

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

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

7.2.11 Space around number

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

`\Xafternumber` With `\Xafternumber[⟨s⟩]{⟨l⟩}` you can add some space after the line number in a footnote. If the line number is not printed, the space is not either. The default value is 0.5 em.

`\Xendbeforenumber` `\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.

7.2.12 Space around line symbol

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

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

`\Xendbeforemsymlinenum` `\Xendaftersymlinenum` are the equivalents of `\Xbeforemsymlinenum` and `\Xaftersymlinenum` for the endnotes.

7.2.13 Space in place of number

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

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

7.2.14 Boxing line number and line symbol

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

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

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

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

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

L to align left (default value);

R to align right;

C to center.

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

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

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

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

`\Xendboxlinenum` `\Xendboxlinenum[⟨s⟩]{⟨l⟩}`, `\Xendboxlinenumalign[⟨s⟩]{⟨text⟩}`, `\Xendboxstartlinenum[⟨s⟩]{⟨l⟩}`,
`\Xendboxlinenumalign` `\Xendboxendlinenum[⟨s⟩]{⟨l⟩}` are the same as, respectively, `\Xboxlinenum` and
`\Xendboxstartlinenumalign` `\Xboxlinenumalign`, `\Xboxstartlinenum`, `\Xboxendlinenum` except in endnotes.
`\Xendboxendlinenumalign`

7.3 Arbitrary code around line number

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

7.4 Separator between the lemma and the note

7.4.1 For footnotes

<code>\Xlemmaseparator</code>	By default, in a footnote, the separator between the lemma and the note is a right bracket (<code>\rbracket</code>) ¹⁸ . You can use <code>\Xlemmaseparator [⟨s⟩]{⟨Xlemmaseparator⟩}</code> 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.
<code>\Xbeforelemmaseparator</code>	Using <code>\Xbeforelemmaseparator [⟨s⟩]{⟨l⟩}</code> 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.
<code>\Xafterlemmaseparator</code>	Using <code>\Xafterlemmaseparator [⟨s⟩]{⟨l⟩}</code> 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.
<code>\Xnolemmaseparator</code>	You can suppress the lemma separator, using <code>\Xnolemmaseparator [⟨s⟩]</code> , which is simply a alias of <code>\Xlemmaseparator [⟨s⟩]{}</code> .
<code>\Xinplaceoflemmaseparator</code>	With <code>\Xinplaceoflemmaseparator [⟨s⟩]{⟨l⟩}</code> you can add a space if no lemma separator is printed. The default value is 1 em.

7.4.2 For endnotes

<code>\Xendlemmaseparator</code>	By default, there is no separator inside endnotes between the lemma and the content of the note. You can use <code>\Xendlemmaseparator [⟨s⟩]{⟨Xendlemmaseparator⟩}</code> to change this. The optional argument can be used to specify the series in which it is used. A common value of <code>⟨Xendlemmaseparator⟩</code> is <code>\rbracket</code> . 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.
<code>\Xendbeforelemmaseparator</code>	Using <code>\Xendbeforelemmaseparator [⟨s⟩]{⟨l⟩}</code> 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.
<code>\Xendafterlemmaseparator</code>	Using <code>\Xendafterlemmaseparator [⟨s⟩]{⟨l⟩}</code> you can add some space between

¹⁸For `polyglossia`, when the lemma is RTL, the bracket automatically switches to a left bracket.

the separator and the content of the note. If your lemma separator is empty, this space won't be printed. The default value is 0.5 em.

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

7.5 Font style

7.5.1 For line number

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

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

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

7.5.2 For the lemma

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

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

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

`\Xendlemmafnt`

```
\Xlemmafnt{\bfseries}
```

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

7.5.3 For all notes

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

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

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

7.6 Wrapping notes

7.6.1 Wrapping lemmas

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

7.6.2 Wrapping contents

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

7.7 Indent of notes content

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

7.8 Arbitrary code at the beginning of notes

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

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

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

7.9 Arbitrary code before inserting note

`\Xbeforeinserting` `\beforeinsertingX` and `\beforeinsertingX` `\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 `bidir`, but want your critical footnote be left-to-right, use `\Xbeforeinserting\LTR`. You should also use `\Xwraplemma` to ensure your lemmas are right-to-left in a left-to-right paragraph (7.6.1 p. 46).

Note that the changes are local to the footnote.

7.10 Options for footnotes in columns

7.10.1 Alignment

`\Xcolalign` `\colalignX` By default, text in footnotes of two or three columns are flush left and without hyphenation. However, you can change this with `\Xcolalign` `\colalignX` for critical footnotes, and `\colalignX` `\colalignX` 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.

`\Xhsizetwocol` `\Xhsizetwocol` `\Xhsizetwocol` `\Xhsizetwocol` is used to change width of a column when critical notes are displaying in two columns. Default value is `.45 \hspace`.

`\Xhsizethreecol` `\Xhsizethreecol` `\Xhsizethreecol` `\Xhsizethreecol` is used to change width of a column when critical notes are displaying in three columns. Default value is `.3 \hspace`.

`\hsizetwocolX` `\hsizetwocolX` `\hsizetwocolX` `\hsizetwocolX` is used to change width of a column when familiar notes are displaying in two columns. Default value is `.45 \hspace`.

`\hsizethreecolX` `\hsizethreecolX` `\hsizethreecolX` `\hsizethreecolX` 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

<code>\Xafternote</code>	You can add some horizontal space after a note by using <code>\Xafternote[⟨s⟩]{⟨l⟩}</code> (for critical footnotes) or <code>\afternoteX[⟨s⟩]{⟨l⟩}</code> (for familiar footnotes). The default value is <code>1em plus .4em minus .4em</code> .
<code>\afternoteX</code>	
<code>\Xparafootsep</code>	For paragraphed footnotes (see below), you can choose the separator between each note by using <code>\Xparafootsep[⟨s⟩]{⟨text⟩}</code> for critical notes and <code>\parafootsepX</code> for familiar notes. A common separator is the double pipe (<code> </code>), which you can set by using <code>\Xparafootsep{⟨\$⟩\parallel\$⟨\$⟩}</code> . Note that if the symbol defined by <code>\Xsymllinenum</code> must be used at the beginning of a note, the <code>\Xparafootsep / \parafootsepX</code> is not used before this note.
<code>\parafootsepX</code>	

7.11.2 Ragged text

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

7.12 Options for block of notes

7.12.1 Grouping notes by line

<code>\Xgroupbyline</code>	If you do not use <code>\Xarrangement{paragraph}</code> , you may want to group all the critical footnotes related to the same line in the same paragraph. In this case, use <code>\Xgroupbyline[⟨series⟩]</code> . In many cases, you might like to use it in combination with <code>\Xnumberonlyfirstinline</code> (7.2.1 p. 39). Note that the <code>\Xafternote</code> and <code>\Xparafootsep</code> settings are used to determine space and content between footnotes (7.11 p. 48). Suppose you have two notes on line 1 which overlap lines 1 and 2. This last note will be printed, if you use <code>\Xgroupbyline</code> in the same group as the previous one. In the case you want that note to be distinct, you must use both <code>\Xgroupbyline</code> and <code>\Xgroupbylineseparetwolines[⟨s⟩]</code> . In many cases, you might like to use it in combination with <code>\Xnumberonlyfirstintwolines</code> (7.2.1 p. 39).
<code>\Xgroupbylineseparetwolines</code>	

7.12.2 Text before notes

<code>\Xtxtbeforenotes</code>	You can add text before critical footnotes with <code>\Xtxtbeforenotes[⟨s⟩]{⟨text⟩}</code> . You can add text before familiar footnotes with <code>\txtbeforenotesX[⟨s⟩]{⟨text⟩}</code> .
<code>\txtbeforenotesX</code>	
<code>\Xtxtbeforenotesonlyonce</code>	By default, such texts are inserted at the beginning of the groups of notes on each pages. You can add <code>\Xtxtbeforenotesonlyonce</code> (for critical footnotes) and <code>\txtbeforenotesonlyonceX</code> (for familiar footnotes) to insert them only the first time notes are typeset.
<code>\txtbeforenotesonlyonceX</code>	

7.12.3 Code before notes

`\Xbhookgroup` While `\Xtxtbeforenotes` is for typesetting text before notes, `\Xbhookgroup[⟨s⟩]{⟨code⟩}` and `\bhookgroupX[⟨s⟩]{⟨code⟩}` (respectively for critical and familiar) are for executing code before a groups 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 can't be flexible, and is relative to the the current font. For example, if you want the note to take, at most, 33% of the text height, do `\Xmaxhnotes{.33\textheight}`.

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

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

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

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

7.12.7 Width

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

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

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

Note that changes the width of the block of notes. If you want to change the width of each column when typesetting notes in columns, use `\Xhsizetwocol`, `\Xhsizethreecol`, `\hsizetwocolX`, `\hsizethreecolX`, see 7.10.2 p. 47.

7.13 Footnotes and the `reledpar` columns

`\Xnoteswidthliketwocolumns`
`\noteswidthliketwocolumnsX`

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

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

7.14 Line number annotation

The way the annotation to lines number annotations are typeset can be change using hooks described on ?? p. ?? and 9.10.3 p. 57.

7.15 Endnotes in one paragraph

`\Xendparagraph`

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

`\Xendafternote`

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

`\Xendsep`

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

8 Fonts

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

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

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

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

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

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

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

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

9 Verse

9.1 Basic

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

If you need to add brackets directly after `\stanza`, `&` or `\&`, add `\norelax`. Otherwise, the brackets will be interpreted as delimitation of an optional argument (cf. 9.8 p. 54)

9.2 Define stanza indents

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

`\setstanzaindent` In order to use the stanza macros, **one must set the indentation values**. First the

value of `\stanzaindentbase` should be set, unless the default value 20pt is desired. Every stanza line indentation is a multiple of this.

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

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

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

If you want the hanging verse to be flush right, you can use `\sethanginsymbol`: see p. 9.6 p. 53.

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

9.3 Repeating stanza indents

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

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

is like

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

Be careful: the feature is changed in `eledmac` 1.5.1. See Appendix A.3 p. 377.

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

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

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

The macros make no restriction on the number of lines in a stanza. Stanza indentation values (and penalty values) obey \TeX 's grouping conventions, so if one stanza among several has a different structure, its indentations (penalties) may be set within a group; the prior values will be restored when the group ends.

9.4 Manual stanza indent

`\stanzaindent`
`\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`s for this verse is skipped, and `{⟨value⟩}` is used instead.

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

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

9.5 Stanza breaking

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

The command

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

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

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

9.6 Hanging symbol

It is possible to insert a symbol in each line of hanging verse, as in French typography; for example, the opening bracket ‘[’. To insert it in `reledmac`, use macro `\sethangingsymbol{⟨h⟩}` with this code. In the example of French typography, do

`\sethangingsymbol`

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

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

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

9.7 Long verse and page break

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

9.8 Content before/after verses

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

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

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

`\AtEveryStanza`

- Use `\AtEveryStanza{⟨arg⟩}` to automatically add content before the stanza (not in the same paragraph).

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

`\AtStartEveryStanza`

- Use `\AtStartEveryStanza` to automatically add content at the beginning of stanza (in the same paragraph).

- `&` can be replaced by `\newverse` with two optional arguments (in brackets). The first will be printed after the current verse, the second before the next verse.

A `\noindent` is automatically inserted before the contents of these optional arguments.

Use a third and fourth optional argument to not add these `\noindents` (to add content respectively after the current verse / before the next verse).

- Use `\AtEveryPend{⟨arg⟩}` to automatically add content after verses (including the final one) and `\AtEveryPstart{⟨arg⟩}` to automatically add content before verses (including the first one).

- `\&` can take an optional argument (in brackets). Its content will be printed after the stanza.

`\AtEveryStopStanza`

- Use `\AtEveryStopStanza` to automatically add content after the end of stanzas (not in the same paragraph).

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

`\AtStartEveryStanza`

- Use `\AtStartEveryStanza` to automatically add content at the end of stanza (in the same paragraph).

9.9 Numbering stanza

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

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

You can use this feature in combination with `\Xstanza` (7.2.8 p. 41).

`\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}%
}
```

9.10 Add annotation to line of verse numbers

It may happen you want to have two numbers or more associated to a line of verse. For example in the following case:

- You want to split a line of verse in two part depending of some stylistic / rhythmic / linguistic convention.
- You want to add the number of the line of verse that a previous edition of the work used.

In those two cases, you should add the other number manually, as `reledmac` can't determine a general pattern for such numbering, which is too much depending of the edited text.¹⁹

`\linenumannotation` To solve this issue, `reledmac` allow you to add annotation to a line of verse number:

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

The annotation can contain any textuel value (number, text, other thing). For example, the line number in the older edition. Here are some characteristics of a line number annotation:

- A line number annotation is associated to a line of verse. It is reset at each new line of verse.

¹⁹However, you may create your own commands with such pattern which will call `reledmac` commands.

- It is printed along the line number in margin.
- It is printed in critical footnotes and endnotes after line number.
- But here is main interest of this feature: the annotation is printed in the critical note only if the `\edtext` is called after the `\linenumannotation`.
- If two or more annotation are called before a `\edtext`, the last one is used in notes associated.
- If a annotation is called in a `\edtext`, it is printed as an annotation for the end of the lemma.

Here an example, supposing we are on line 1:

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

In the critical notes, the line number will be followed :

- by nothing in the first lemma, as there is still not yet annotation for this line;
- by “A” for the second lemma;
- by “A-B” for the third lemma, as it start when annotation A is put and finish when annotation B is put;
- by “B” for the fourth lemma.

9.10.1 Modify annotation associated to a specific note

`\lineannot` The `\linenum` (6.2.5 p. 27) macro change the line number passed to a note. is the same to change annotation. It takes one argument, contening the beginning annotation and the ending annotation, separated by a pip (|). As for `\linenum`, if you specify to change an annotation, it will use the current one. For example:

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


9.10.2 Changing the position of the annotation

annotationposition By default, the annotation are put, both in margin and in footnote, after the line numbers. You can use one of the following command:

- `\Xlinenumannotationposition[⟨s⟩]{⟨v⟩}` to change the position in critical footnotes. `⟨s⟩` means for the series of notes which is applied (it can be a comma separated list of series). If `⟨s⟩` is empty, it also applies to annotation in the side of the pages, along the line number.
- `\Xlinenumannotationpositionside{⟨v⟩}` to change the position in the side of pages, along the line numbers.

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

- `\Xendlinenumannotationposition[⟨s⟩]{⟨v⟩}` to changes the position in critical endnotes. `⟨s⟩` means for the series of notes which is applied (it can be a comma separated list of series).

In the threes previous commands, `⟨v⟩` can be after or before, relatively to the line number.

9.10.3 Changing macro in which annotations are wrapped

By default, the annotation are wrappe in the `\textsuperscript` macro. You can modify it using one of the following macros:

- `\Xwraplinenumannotation[⟨s⟩]{⟨cmd⟩}` for critical footnotes. `⟨s⟩` means for the series of notes which is applied (it can be a comma separated list of series). If `⟨s⟩` is empty, it also applies to the annotations in the margins of the page, around the line number, and to the annotations referenced using the `\annotationref` command of the crossref mechanism (11 p. 59).

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

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

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

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

- `\Xendwraplinenumannotation[⟨s⟩]{⟨cmd⟩}` for critical endnotes. `⟨s⟩` means for the series of notes which is applied (it can be a comma separated list of series).

`⟨\⟩` must be a command which can take one argument, the antislash is required

For example, if you don't want to have annotation in the margin, and want to have them in bold in critical notes, do:²⁰

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

9.11 Various tools

- `\ampersand` If you need to print an & symbol in a stanza, use the `\ampersand` macro, not `\&` which will end the stanza.
- `\flagstanza` Putting `\flagstanza[⟨len⟩]{⟨text⟩}` at the start of a line in a stanza (or elsewhere) will typeset `⟨text⟩` at a distance `⟨len⟩` before the line. The default `⟨len⟩` is `\stanzaindentbase`.

9.12 Notes on empty lines

Since v2.3.0 of `reledmac`, empty lines when typesetting verses no longer produce new paragraphs, and consequently, do not insert vertical spaces. Use optional argument of `\stanza` or `\newverse` to insert vertical space (9.8 p. 54).

10 Grouping

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

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

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

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

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

```
\begin{ledgroup sized}[⟨pos⟩]{⟨width⟩}.
```

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

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

²⁰The `\@gobble` command take one argument, and return nothing.

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

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

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

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

11 Cross referencing

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

11.1 Basic use

`\edlabel` First you place a label in the text using the command `\edlabel{<lab>}`. `<lab>` can be almost anything you like, including letters, numbers, punctuation, or a combination—anything but spaces; you might type `\edlabel{toves-3}`, for example.²¹

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

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

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

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

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

11.2 Cross-referencing to a critical note

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

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

If you want to refer to starting and ending lines, you should use `\appref` and related tools (11.6.2 p. 61).

²¹More precisely, you should stick to characters in the `TeX` categories of “letter” and “other”.

11.3 Cross-referencing which return a number in any case

`\xpageref`
`\xlineref`
`\xsublineref`
`\xpstartref`
`\xannotref`

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. 27).

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

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

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

`\xxref`

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

It automatically calls `\linenum` (6.2.5 p. 27) and `\lineannot` (?? p. ??) and sets the beginning page, line, subline numbers and line annotations to those of the place where `\edlabel{mouse}` was placed, and the ending ones to those where `\edlabel{elephant}` occurs.

For example, one might use the following:

```
\beginnumbering

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

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

\endnumbering
```

11.4 Not automatic cross-referencing

`\edmakeabel` 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 `\edmakeabel{<lab>}{<numbers>}` macro so that you can ‘roll your own’ label.

For example, if you type `\edmakeabel{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 `\edmakeabel` statements near the top of your document, so that you can see them at a glance.

11.5 Normal L^AT_EX cross-referencing

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

11.6 References to start and end lines

11.6.1 Reference to main text lines

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

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

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

`\edlabelSE` Use `\edlabelSE{<label>}` to mark just one location in the text. Contrary to a 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 endnotes.

`\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. 27) 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`
`\setapprefprefixmore`

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`

`\setSErefprefixsingle` and `\setSErefprefixmore` are similar for `\SEref` command.

`\setSErefonlypageprefixsingle`
`\setSErefonlypageprefixmore`

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

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

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

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

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

These commands are the following:

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

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

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

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

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

When calling `\apprefwithpage`, `\SErefwithpage` or `\SErefonlypage` you can use as a first optional argument, in brackets (`[]`), any optional argument which can be used for critical endnotes (6.2.3 p. 25).

11.7 Compatibility with *xr* package

The `\externaldocument` command of the `\xr` package allows making cross-references from an external document, with the standard \LaTeX 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 Side notes

12.1 Basics

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

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

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

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

`\ledsidenote` `\sidenotemargin{right}`

`\sidenotemargin` 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.

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 text into a box of width `\ledrsnotewidth`. These are initially set to the value of `\marginparwidth`.

`\ledrsnotewidth`

12.2.2 Vertical position

`\rightnoteupfalse` By default, sidenotes are placed to align with the last line of the note to which it refers. 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).

`\leftnoteupfalse`

12.2.3 Distance to the main text

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

`\ledrsnotesep`

12.2.4 Font

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

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


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

These can of course be changed to suit.

12.2.5 Separator between notes

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

13 Indexing

13.1 Basics

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

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

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`.

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

13.2 Referring to critical notes

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

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

If you add `\edindex` inside some `\Xfootnote` command, it will refer to that note, and a suffix n will be appended to the reference. You can redefine this suffix by redefining the command `\ledinnotemark`. Its actual definition is:

```
\newcommand{\ledinnotemark}[1]{#1\emph{n}}
```

Where `#1` stands for the reference.

13.3 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 separator²².

```
page_compositor ":"
```

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

13.4 Using xindy

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

```
pagenumber-linenumber
```

An example of such a file is provided in the "examples" folder. Read the `xindy` handbook to learn how to use it.²³

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:

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

²³Or, 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.²⁴

13.5 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 (13.3 p. 66).

14.2 Commands

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

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

²⁴These 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.

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

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

1	2	3
a	bb	ccc
AAA	BB	C

Entries in the environments are the same as for the normal array and tabular environments but there must be no ending `\\` at the end of the last row. *There must be the same number of column designators (the &) in each row.* There is no equivalent to any line drawing commands (such as `\hrule`). 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 $\langle math \rangle$ but the $\langle math \rangle$ 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}
1 & 2 & 3 & 4 \\
& F + G + C & & \\
a & bb & ccc & dddd
\end{array}
```

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

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

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

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

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

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

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

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

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

1	2	3	4
a	┌		d
A	B	C	D

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

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

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

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

For example:

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

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

`\edvertline` `\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. 18):

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

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

16.2 Sectioning commands with line numbering and critical notes

You have to use the following commands:

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

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

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

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

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

16.3 Optimization

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

17 Quotation environments

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

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

18 Page breaks

18.1 Control page breaking

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

```
\ledpb
\lednopb
```


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

These commands have effect only at the second run.

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

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

18.2 Prevent page break in a long verses

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

`\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]{\textit{#1}}
```

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

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

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

19.1 Known and suspected limitations

19.1.1 Non-standard geometry

If you use classes other than `article` or `book`, or if you use the `geometry` package, you should use `maxhnotesX` and/or `\Xmaxhnotes` as explained in 7.12.6 p. 49 in order to prevent footnotes from overlapping the bottom margin.

19.1.2 floatrow package compatibility

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

19.1.3 ‘No room for a new’

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

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

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

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

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

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

19.1.4 Marginal notes

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

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

19.1.5 Paragraph shape

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

`\ballast`

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

19.1.6 Paragraphed footnotes

The restriction on explicit line-breaking in paragraphed footnotes, mentioned on 7.1 p. 38, and described in more detail on XIII.6.3 p. 187, really is a nuisance if that is something you need to do. There are some possible solutions, described by Michael Downes, but this area remains unsatisfactory.

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

`\footfudgefiddle`

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

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

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

Any settings to ‘geometry’ must be made before `\Xarrangement / \arrangementX`.

Finally, in many cases you should use `\Xmaxhnotes` and / or `\maxhnotesX` (7.12.6 p. 49), 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 this²⁵ 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

²⁵Reported 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 `Eledmac` 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 XXIV) 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}[2017/10/21 v2.26.0 typesetting critical editions
4 ]%
5 %

```

II.3 Package options

```

\ifledfinal Use this to remember which option is used, set and execute the options with final as the
\ifnocritical@ default. We use xkeyval in order to manage options with argument.
\if@noeled@sec \RequirePackage{xkeyval}
\ifnoend@ %
\ifnofamiliar@
\ifnoledgroup@ The parledgroup option is for reledpar. However, it has consequence on reledmac
\ifparapparatus@ internal command. So we need to define the boolean now.
\ifnoquotation@ \newif\ifparledgroup
\iflednopbinverse %
\ifparledgroup
\ifwidthliketwocolumns And now, the options of reledmac.
\ifxindy@ \DeclareOptionX{series}[A,B,C,D,E]{\xdef\default@series{#1}}
\ifxindyhyperref@ \ExecuteOptionsX{series}%
\ifeledmaccompat@
12 \newif\if@noeled@sec%
13 \DeclareOptionX{noeledsec}{\@noeled@sectrue}
14
15 \newif\ifnocritical@%
16 \DeclareOptionX{nocritical}{\nocritical@true}%
17
18 \newif\ifnofamiliar@%
19 \DeclareOptionX{nofamiliar}{\nofamiliar@true}%
20
21 \newif\ifnoledgroup@%
22 \DeclareOptionX{noledgroup}{\noledgroup@true}%
23
24 \newif\ifnoend@%
25 \DeclareOptionX{noend}{%
26 \let\l@dend@open\@gobble%
27 \let\l@dend@close\relax%
28 \global\let\l@dend@stuff=\relax%
29 \noend@true%
30 }%
31
32 \newif\ifnoquotation@
33 \DeclareOptionX{noquotation}{\noquotation@true}
34
35 \newif\ifledfinal
36 \DeclareOptionX{final}{\ledfinaltrue}
37 \DeclareOptionX{draft}{\ledfinalfalse}

```

```

38 \ExecuteOptionsX{final}
39
40 \newif\ifparapparatus@
41 \DeclareOptionX{parapparatus}{\parapparatus@true}
42
43 \newif\iflednopbinverse
44 \DeclareOptionX{nopbinverse}{\lednopbinversetrue}
45
46 \newif\ifwidthliketwocolumns%
47 \DeclareOptionX{widthliketwocolumns}{\widthliketwocolumnstrue}%
48
49 \newif\ifcontinuousnumberingwithcolumns
50 \DeclareOptionX{continuousnumberingwithcolumns}{\
continuousnumberingwithcolumnstrue}%
51
52 \newif\ifxindy@
53 \DeclareOptionX{xindy}[eledmac-markup-attr.xdy]{%
54   \AtBeginDocument{\immediate\openout\eledmac@xindy@out=#1}%
55   \newwrite\eledmac@xindy@out%
56   \xindy@true%
57   \gdef\eledmacmarkuplocdepth{:depth 1}%
58   \AtEndDocument{\immediate\closeout\eledmac@xindy@out}%
59 }%
60
61 \newif\ifxindyhyperref@
62 \DeclareOptionX{xindy+hyperref}{%
63   \xindyhyperref@true%
64 }%
65
66 \newif\ifeledmaccompat@%
67 \DeclareOptionX{eledmac-compat}{%
68   \eledmaccompat@true%
69 }%
70 \DeclareOptionX{nopenalties}{%
71   \AtBeginDocument{\let\add@penalties\relax}%
72 }
73 \def\l@auxdir{}%
74 \DeclareOptionX{auxdir}{%
75   \xdef\l@auxdir{#1}/}%
76 }%
77
78 \newif\ifsw@caseinsensitive%
79 \DeclareOptionX{swcaseinsensitive}{%
80   \sw@caseinsensitivetrue%
81 }%
82 %

```

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.

```
83 \ProcessOptionsX*\relax
84
85 %
```

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 Lua \TeX or X \TeX is running, and `ragged2e` to manage ragged justification for paragraphed notes.

```
86 \RequirePackage{xargs}
87 \RequirePackage{xparse}%
88 \RequirePackage{etoolbox}
89 \@ifl@t@r\fmtversion{2015/10/01}
90 {\ifboolexpr{not test{\@ifl@t@r\fmtversion{2016/03/31}} or (test{\
  ifdefstring{\fmtversion}{2016/03/31}} and test {\ifnumless{\patch@level
  }{3}})}}%
91   {\PackageWarning{reledmac}{You are using a LaTeX version older than
  2016/03/31 patch 3.%
92   \MessageBreak You are strongly encouraged to use a newer version.}}%
93   {}%
94 }%
95 {\RequirePackage{etex}%
96  \csname reserveinserts\endcsname{32}%
97  }%
98 \RequirePackage{suffix}
99 \RequirePackage{xstring}
100 \RequirePackage{ifluatex}
101 \RequirePackage{ragged2e}
102 \RequirePackage{ifxetex}%
103 %
```

II.5 Compatibility with Lua \TeX

Here, we enable some primitives for Lua \TeX .

```
104 \ifx\directlua\undefined\else%
105   \directlua{tex.enableprimitives("",{"textdir","pardir","bodydir"})}
106 \fi
107 %
```


II.6 Boolean flags

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

```
108 \newif\ifl@dmemoir
109 \@ifclassloaded{memoir}{\l@dmemoirtrue}{\l@dmemoirfalse}
110
111 %
```

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

```
112 \newif\if@ledgroup%
113 %
```

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

```
114 \newif\ifl@imakeidx
115 \@ifpackageloaded{imakeidx}{\l@imakeidxtrue}{}%False is the default value
116 %
```

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

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

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.

```
124 \newif\ifl@footmisc
125 \@ifpackageloaded{footmisc}{\l@footmisctrue}{}%False is the default value
126 %
```

`\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.

```
127 \ifdef{\if@RTL}{\newif\if@RTL}
128 %
```

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

```
129 \newif\if@firstlineofpage%
130 \newif\if@firstlineofpageR%
131 %
```

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.

```
132 \newcommand{\reledmac@warning}[1]{\PackageWarning{reledmac}{#1}}
133 %
```

`\reledmac@error` Write an error message.

```
134 \newcommand{\reledmac@error}[2]{\PackageError{reledmac}{#1}{#2}}
135 %
```

```
\led@err@NumberingStarted36 \newcommand*\led@err@NumberingStarted}{%
```

```
\led@err@NumberingNotStarted37 \reledmac@error{Numbering has already been started}{\@ehc}}
```

```
\led@err@NumberingShouldHaveStarted38 \newcommand*\led@err@NumberingNotStarted}{%
```

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

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

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

```
142 %
```

```
\led@err@edtextoutsidestart43 \newcommand*\led@err@edtextoutsidestart}{%
```

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

```
145 %
```

```
\led@mess@NotesChanged46 \newcommand*\led@mess@NotesChanged}{%
```

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

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

```
150 \typeout{ You will need to run LaTeX two more times
151 before the footnote placement}%
```

```
152 \typeout{ and line numbering in this section are
153 correct.}}
```

```
154 %
```

```
\led@mess@SectionContinued55 \newcommand*\led@mess@SectionContinued}[1]{%
```

```
156 \message{Section #1 (continuing the previous section)}}%
```

```
157 %
```

```
\led@err@LineationInNumbered58 \newcommand*\led@err@LineationInNumbered}{%
```

```
159 \reledmac@error{You can't use \string\lineation\space within
160 a numbered section}{\@ehc}}
```

```
161 %
```

```

\led@warn@BadLineation162 \newcommand*{\led@warn@BadLineation}{%
\led@warn@BadLinenummargin163 \reledmac@warning{Bad \string\lineation\space argument}}
\led@warn@BadLockdisp164 \newcommand*{\led@warn@BadLinenummargin}{%
\led@warn@BadSublockdisp165 \reledmac@warning{Bad \string\linenummargin\space argument}}
166 \newcommand*{\led@warn@BadLockdisp}{%
167 \reledmac@warning{Bad \string\lockdisp\space argument}}
168 \newcommand*{\led@warn@BadSublockdisp}{%
169 \reledmac@warning{Bad \string\sublockdisp\space argument}}
170 %

\led@warn@NoFile171 \newcommand*{\led@warn@NoFile}[1]{%
172 \reledmac@warning{File `#1' not found}}
173 %

\led@warn@LineFileObsolete174 \newcommand*{\led@warn@Obsolete}[1]{%
175 \reledmac@warning{Line-list file #1 was obsolete. We have not read it.
Please run LaTeX again.}}
176 %

\led@warn@BadAdvancelineSubline177 \newcommand*{\led@warn@BadAdvancelineSubline}{%
\led@warn@BadAdvancelineLine178 \reledmac@warning{\string\advanceline\space produced a sub-line
179 number less than zero.}}
180 \newcommand*{\led@warn@BadAdvancelineLine}{%
181 \reledmac@warning{\string\advanceline\space produced a line
182 number less than zero.}}
183 %

\led@warn@BadSetline184 \newcommand*{\led@warn@BadSetline}{%
\led@warn@BadSetlinenum185 \reledmac@warning{Bad \string\setline\space argument}}
186 \newcommand*{\led@warn@BadSetlinenum}{%
187 \reledmac@warning{Bad \string\setlinenum\space argument}}
188 %

\led@err@PstartNotNumbered189 \newcommand*{\led@err@PstartNotNumbered}{%
\led@err@PstartInPstart190 \reledmac@error{\string\pstart\space must be used within a
\led@err@PendNotNumbered191 numbered section %
\led@err@PendNoPstart192 (\string\beginnumbering\space ..\space\string\endnumbering)
} {\@ehc}}%
\led@err@AutoparNotNumbered193 \newcommand*{\led@err@PstartInPstart}{%
\err@NumberingWithoutPstart194 \reledmac@error{\string\pstart\space encountered while another
195 \string\pstart\space was in effect}{\@ehc}}
196 \newcommand*{\led@err@PendNotNumbered}{%
197 \reledmac@error{\string\pend\space must be used within a
198 numbered section}{\@ehc}}

```

```

199 \newcommand*\led@err@PendNoPstart}{%
200   \reledmac@error{\string\pend\space must follow a \string\pstart}{\@ehc}}
201 \newcommand*\led@err@AutoparNotNumbered}{%
202   \reledmac@error{\string\autopar\space must be used within a
203     numbered section}{\@ehc}}
204 \newcommand*\led@err@NumberingWithoutPstart}{%
205   \reledmac@error{\string\beginnumbering...\string\endnumbering\space
206     without \string\pstart}{\@ehc}}%

```

```

\led@warn@BadAction207 \newcommand*\led@warn@BadAction}{%
208   \reledmac@warning{Bad action code, value \next@action.}}
209 %

```

```

\led@warn@DuplicateLabel210 \newcommand*\led@warn@DuplicateLabel}[1]{%
\led@warn@AppLabelOutSecondArgEdtext211   \reledmac@warning{Duplicate definition of label `#1'\@gobble}%
\led@warn@RefUndefined212   \@latex@warning@no@line{Label `#1' multiply defined}%
\led@warn@RefUndefined213 }%
214 \newcommand*\led@warn@AppLabelOutSecondArgEdtext}[1]{%
215   \reledmac@warning{\string\applabel\space outside of the second argument
216     of an \string\edtext\space `#1' on page \thepage.}}%
217 \newcommand*\led@warn@RefUndefined}[1]{%
218   \G@refundefinedtrue%
219   \reledmac@warning{Reference `#1' on page \thepage\space undefined.%
220     Using `000'.}%
221   \@latex@warning{Reference `#1' undefined\on@line}%
222 }%
223 \newcommand*\led@warn@pairRefUndefined}[1]{%
224   \G@refundefinedtrue%
225   \reledmac@warning{Reference `#1:start' and/or `#1:end' on page \thepage\
226     space undefined.
227     Using `??'.}%
228   \@latex@warning{Reference `#1:start' and/or `#1:end' undefined\on@line}%
229 }

```

```

\led@warn@NoMarginpars229 \newcommand*\led@warn@NoMarginpars}{%
230   \reledmac@warning{You can't use \string\marginpar\space in numbered text
231   }}

```

```

\led@warn@BadSidenotemargin232 \newcommand*\led@warn@BadSidenotemargin}{%
233   \reledmac@warning{Bad \string\sidenotemmargin\space argument}}
234 %

```

```

\led@warn@NoIndexFile35 \newcommand*{\led@warn@NoIndexFile}[1]{%
236   \reledmac@warning{Undefined index file #1}}
237 %

\led@warn@SeriesStillExist38 \newcommand{\led@warn@SeriesStillExist}[1]{%
239   \reledmac@warning{Series #1 is still existing !}%
240 }%
241 %

\led@err@BadAction42 \newcommand*{\led@err@StanzaIndentNotDefined}{%
243   \reledmac@error{You have not defined the indentation for the line \number
\stanza@count}{\@ehc}}%
244 %

\led@err@ManySidenotes45 \newcommand{\led@err@ManySidenotes}{%
\led@err@ManyLeftnotes46   \ifledRcol{%
\led@err@ManyRightnotes47   \reledmac@warning{\itemcount@space sidenotes on line \the\line@numR\
space p. \the\page@numR}%
248   \else%
249     \reledmac@warning{\itemcount@space sidenotes on line \the\line@num\
space p. \the\page@num}%
250     \fi%
251   }%
252 \newcommand{\led@err@ManyLeftnotes}{%
253   \ifledRcol{%
254     \reledmac@warning{\itemcount@space leftnotes on line \the\line@numR\
space p. \the\page@numR}%
255     \else%
256       \reledmac@warning{\itemcount@space leftnotes on line \the\line@num\
space p. \the\page@num}%
257       \fi%
258     }%
259 \newcommand{\led@err@ManyRightnotes}{%
260   \ifledRcol{%
261     \reledmac@warning{\itemcount@space rightnotes on line \the\line@numR\
space p. \the\page@numR}%
262     \else%
263       \reledmac@warning{\itemcount@space rightnotes on line \the\line@num\
space p. \the\page@num}%
264       \fi%
265     }%
266   %

\led@err@TooManyColumns67 \newcommand*{\led@err@TooManyColumns}{%
\led@err@UnequalColumns68   \reledmac@error{Too many columns}{\@ehc}}
\led@err@LowStartColumn
\led@err@HighEndColumn
\led@err@ReverseColumns

```

```

269 \newcommand*\led@err@UnequalColumns}{%
270   \reledmac@error{Number of columns is not equal to the number
271     in the previous row (or \protect\\ \space forgotten?)}{\@ehc}}
272 \newcommand*\led@err@LowStartColumn}{%
273   \reledmac@error{Start column is too low}{\@ehc}}
274 \newcommand*\led@err@HighEndColumn}{%
275   \reledmac@error{End column is too high}{\@ehc}}
276 \newcommand*\led@err@ReverseColumns}{%
277   \reledmac@error{Start column is greater than end column}{\@ehc}}
278 %

```

```

\led@err@toendnotes@outsidenumbering 279 \newcommand\led@err@toendnotes@outsidenumbering}{%
280   \reledmac@error{\string\toendnotes\space and related commands must be
called inside a numbered texte (\string\beginnumbering\space ... \space\string
\endnumbering)}{\@ehc}%
281 }%
282 %

```

```

\led@err@EdtextWithoutFootnote 283 \newcommand\led@err@EdtextWithoutFootnote}{%
284   \reledmac@error{edtext without Xfootnote. Check syntaxis}{\@ehc}%
285 }%
286 %

```

```

\led@err@FootnoteNotInSecondArgEdtext 287 \newcommand\led@err@FootnoteNotInSecondArgEdtext}[1]{%
288   \reledmac@error{#1footnote outside of the second argument of an edtext.
Check syntax}{\@ehc}%
289 }%
290 %

```

```

\led@error@PackageAfterEledmac 291 \newcommand\led@error@PackageAfterEledmac}[1]{%
292   \reledmac@error{#1 must be loaded before reledmac}{\@ehc}%
293 }%
294 %

```

```

\led@error@fail@patch@makecol 295 \newcommand\led@error@fail@patch@makecol}{%
296   \reledmac@error{Fail to patch \string\makecol\space command}{\@ehc}%
297 }%
298 %

```

```

\led@error@fail@patch@reinserts 299 \newcommand\led@error@fail@patch@reinserts}{%
300   \reledmac@error{Fail to patch \string\reinserts\space command}{\@ehc}%
301 }%
302 %

```

```

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

```

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

```

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

```

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

```

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

```

\newcommand{\led@warning@hsizeX@deprecated}{%
  \reledmac@warning{\string\hsizeX\space command deprecated, use \string\widthX\space instead.}%
}%
%
```

```

\newcommand{\led@warning@Xhsize@deprecated}{%
  \reledmac@warning{\string\Xhsize\space command deprecated, use \string\Xwidth\space instead.}%
}%
%
```

```

\led@warning@msdatawithoutstop331 \newcommand{\led@warning@msdatawithoutstop}{%
332   \reledmac@warning{\string\msdata\space without corresponding \string\
stopmsdata}%
333 }%
334 %

```

```

\led@warning@preXnotes@deprecated335 \newcommand{\led@warning@preXnotes@deprecated}{%
336   \reledmac@warning@preXnotes@deprecated%
337 }%
338 %

```

II.8 Gobbling

Here, we define some commands which gobble their arguments.

```

\@gobblethree339 \providecommand*\@gobblethree}[3]{}
\@gobblefour340 \providecommand*\@gobblefour}[4]{}
\@gobbleseven341 \providecommand*\@gobbleseven}[7]{}
342 %

```

II.9 Wrapping content

The `\l@wrap@ifnotempty` macro wrap its second argument the command passed as first argument, but only if the second argument is not empty. That is useful for example when we need to wrap something on a `\textsuperscript`.

Indeed `ab` does not produce the same typographical result than `ab`.²⁶ The `\l@wrapcs@ifnotempty` does the same, but take as argument a control sequence name, so without the antislash.

```

343 \newcommand{\l@wrap@ifnotempty}[2]{%
344   \edef\@tmpa{#2}%
345   \ifx\@tmpa\empty\else%
346     #1{#2}%
347   \fi%
348 }%
349 \newcommand{\l@wrapcs@ifnotempty}[2]{%
350   \l@wrap@ifnotempty{\csname #1\endcsname}{#2}%
351 }%
352 %

```

II.10 Miscellaneous commands

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

²⁶Compare the distance between letters in the first case (ab) and in the second one (ab).


```

353 \ifledfinal
354   \newcommand*{\showlemma}[1]{#1}
355 \else
356   \newcommand*{\showlemma}[1]{\underline{#1}}
357 \fi
358
359 %

```

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

Initialize it as `\empty`.

```

360 \let\linenumberlist=\empty
361
362 %

```

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

`\@l@tempcntb` \LaTeX already defines `\@tempcnta` and `\@tempcntb` but Peter Wilson found in the past that it can be dangerous to use these (for example one of the AMS packages did something nasty to the `ccaption` package's use of one of these).

```

363 \newcount\@l@tempcnta \newcount\@l@tempcntb
364 %

```

II.11 Prepare reledpar

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

```

\ifl@dpaging
\ifl@dprintingpages
\ifl@dprintingcolumns
365 \newif\ifl@dpairing
\ifpst@rtedL 366 \newif\ifl@dpaging%
\l@dnumstartsL 367 \newif\ifl@dprintingpages%
368 \newif\ifl@dprintingcolumns%
369 \newif\ifpst@rtedL
370 \newcount\l@dnumstartsL
371 %

```

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

```

372 \newif\ifledRcol
373 \newif\ifledRcol@
374 %

```

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

```

375 \newif\ifnumberingR
376 %

```

`\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`.

```
377 \newif\ifafterendnumberingR%
378 %
```

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

```
379 \newif\ifXnote@%
380 %
```

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

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

`\ifindtl@notenumber`

```
381 \providebool{indtl@innote}%
382 \providebool{indtl@notenumber}%
383 %
```

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 `\resumenummering` commands have appeared; it need not be related to the logical divisions of your text.

`\extensionchars` Each section will read and write an associated ‘line-list file’, containing information used to do the numbering; the file will be called `\jobname`.`nn`, where `nn` is the section number. However, you may direct that an extra string be added before the `nn` in that filename, in order to distinguish these temporary files from others: that string is called `\extensionchars`. Initially it’s empty, since different operating systems have greatly varying ideas about what characters are permitted in file names. So `\renewcommand{\extensionchars}{-}` gives temporary files called `jobname.-1`, `jobname.-2`, etc.

```
384 \newcount\section@num
385 \section@num=0
386 \let\extensionchars=\empty
387 %
```

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

```
388 \newif\ifnumbering
389 %
```

`\beginnumbering` `\initnumbering@reg` `\beginnumbering` begins a section of numbered text. When it is executed we increment the section number, initialize our counters, send a message to your terminal, and call macros to start the lineation machinery and endnote files.

The initializations here are trickier than they look. `\line@list@stuff` will use all of the counters that are zeroed here when it assembles the line-list and other lists of information about the lineation. But it will do all of this locally and within a group, and when it is done the lists will remain but the counters will return to zero. Those same counters will then be used as we process the text of this section, but the assignments will be made globally. These initializations actually apply to both uses, though in all other respects there should be no direct interaction between the use of these counters and variables in the two processing steps. For parallel processing :

- zero `\l@dnumpst@startsL` — the number of chunks to be processed.
- set `\ifpst@rtedL` to FALSE.

```
390 \newcommand*{\beginnumbering}{%
391   \ifnumbering
392   \led@err@NumberingStarted
393   \endnumbering
394   \fi
395   \global\numberingtrue
396   \global\afterendnumberingRfalse%
397   \global\advance\section@num \@ne
398   \initnumbering@reg
399   \message{Section \the\section@num }%
400   \line@list@stuff{\jobname.\extensionchars\the\section@num}%
401   \ifcontinuousnumberingwithcolumns%
402     \unless\ifafterendnumberingR%
403       \unless\ifl@dpairing%
404         \ledRcoltrue%
405         \global\advance\section@numR \@ne%
406         \message{Section \the\section@numR R (continuousnumbering)}%
407         \line@list@stuffR{\jobname.\extensionchars\the\section@numR R}%
408         \ledRcolfalse%
409       \fi%
410     \fi%
411   \fi%
412   \l@dend@stuff
413   \setcounter{pstart}{1}
414   \ifl@dpairing
```

```

415 \global\l@dnumpstartsL \z@
416 \global\pst@rtedLfalse
417 %

```

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.

```

418 \else
419 \begingroup
420 \global\@afterindenttrue%In order to reestablish normal feature if the \
beginning was not here
421 \initnumbering@quote
422 \ifwidthliketwocolumns%
423 \setwidthliketwocolumns%
424 \csuse{setpositionliketwocolumns@\columns@position}%
425 \fi%
426 \fi
427 \gdef\eled@sections@{}%
428 \if@noeled@sec\else%
429 \makeatletter%
430 \InputIfFileExists%
431 {\l@auxdir\jobname.eledsec\the\section@num}%
432 {}%
433 {\led@warn@NoFile{\l@auxdir\jobname.eledsec\the\section@num}}%
434 \makeatother%
435 \immediate\openout\eled@sectioning@out=\l@auxdir\jobname.eledsec\the\
section@num\relax%
436 \fi%
437 }
438 \newcommand*{\initnumbering@reg}{%
439 \global\pst@rtedLfalse
440 \global\l@dnumpstartsL \z@
441 \global\absline@num \z@
442 \gdef\normal@page@break{}
443 \gdef\l@prev@pb{}
444 \gdef\l@prev@nopb{}
445 \global\line@num \z@
446 \global\subline@num \z@
447 \global\@lock \z@
448 \global\sub@lock \z@
449 \global\sublines@false
450 \global\let\next@page@num=\relax
451 \global\let\sub@change=\relax
452 \global\last@page@num=-10000%
453 \ifdefined\line@numR%
454 \line@numR=\z@%

```

```

455 \last@page@numR=\z@%
456 \fi%
457 \resetprevline@
458 \resetprevpage@num
459 \global\stopmsdata@inserted@true%
460 \global\let\@msdata@list\relax%
461 \global\csundef{\@msdata@\add@msd@c @data}%
462 }
463
464 %

```

`\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.

```

465 \def\endnumbering{%
466 \ifnumbering
467 \global\numberingfalse
468 \normal@pars
469 \ifnum\l@dnumpstartsL=0%
470 \led@err@NumberingWithoutPstart%
471 \fi%
472 \ifl@dpairing
473 \global\pst@rtedLfalse
474 \else
475 \ifx\insertlines@list\empty\else
476 \global\noteschanged@true
477 \fi
478 \ifx\line@list\empty\else
479 \global\noteschanged@true
480 \fi
481 \fi
482 \ifnoteschanged@
483 \led@mess@NotesChanged
484 \fi
485 \else
486 \led@err@NumberingNotStarted
487 \fi
488 \autoparf@false
489 \if@noeled@sec\else%
490 \immediate\closeout\eled@sectioning@out%
491 \fi%
492 \ifl@dpairing\else
493 \global\l@dnumpstartsL=\z@%
494 \endgroup
495 \fi
496 }
497 %

```

`\pausenumbering` The `\pausenumbering` macro is just the same as `\endnumbering`, but with the
`\resumenumbering` `\ifnumbering` flag set to true, to show that numbering continues across the gap.²⁷
`\ifresumenumbering@start` The `\pausenumbering@page@num` counter stores the `\next@page@num` when the
`\pausenumbering@page@num` `\pause@numbering` is called.

```

498 \newcount\pausenumbering@page@num%
499 \newcommand{\pausenumbering}{%
500   \ifx\next@page@num\relax%
501     \global\pausenumbering@page@num=0%
502   \else%
503     \global\pausenumbering@page@num=\next@page@num%
504   \fi%
505   \ifautopar\global\autopar@pausetrue\fi%
506   \endnumbering\global\numberingtrue}
507 %

```

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.

```

508 \newif\ifresumenumbering@start%
509 \newcommand*{\resumenumbering}{%
510   \ifnumbering
511     \ifautopar@pause\autopar\fi
512     \global\pst@rtedLtrue
513     \global\advance\section@num \@ne
514     \global\resumenumbering@starttrue%
515     \led@mess@SectionContinued{\the\section@num}%
516     \set@continuousnumberingforL%
517     \line@list@stuff{\jobname.\extensionchars\the\section@num}%
518     \ifcontinuousnumberingwithcolumns%
519       \unless\ifafterendnumberingR%
520         \unless\ifl@dpairing%
521           \ledRcoltrue%
522           \global\advance\section@numR \@ne%
523           \message{Section \the\section@numR R (continuousnumbering)}%
524           \line@list@stuffR{\jobname.\extensionchars\the\section@numR R}%
525           \ledRcolfalse%
526         \fi%
527       \fi%
528     \fi%
529     \l@dend@stuff
530   \ifl@dpairing\else%
531     \begingroup%
532     \initnumbering@quote%
533     \ifwidthliketwocolumns%

```

²⁷Peter Wilson's thanks to Wayne Sullivan, who suggested the idea behind these macros.

```

534     \setwidthliketwocolumns%
535     \csuse{setpositionliketwocolumns@\columns@position}%
536     \fi%
537   \fi%
538 \else
539   \led@err@NumberingShouldHaveStarted
540   \endnumbering
541   \beginnumbering
542 \fi}
543
544
545 %

```

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

```

546 \newcommand{\set@continuousnumberingforL}{%
547 \ifcontinuousnumberingwithcolumns%
548   \ifl@dpairing%
549     \unless\ifl@dpageing%
550       \global\c@pstartL=\c@pstart%
551     \fi%
552   \fi%
553 \fi%
554 }%
555 %

```

IV List macros

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

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

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

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

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

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

```

556 \newcommand*{\list@create}[1]{%
557   \global\let#1=\empty%

```

```
558 }%
559 %
```

\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 .

```
560 \newcommand*{\list@clear}[1]{%
561   \global\let#1=\empty%
562 }
563 %
```

\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`.

```
564 \newtoks\led@toksa \newtoks\led@toksb
565 \global\led@toksa={\}
566 \long\def\xright@appenditem#1\to#2{%
567   \global\led@toksb=\expandafter{#2}%
568   \xdef#2{\the\led@toksb\the\led@toksa\expandafter{#1}}%
569   \global\led@toksb={}}
570 %
```

\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`.

```
571 \long\def\xleft@appenditem#1\to#2{%
572   \global\led@toksb=\expandafter{#2}%
573   \xdef#2{\the\led@toksa\expandafter{#1}\the\led@toksb}%
574   \global\led@toksb={}}
575 %
```

\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.

```
576 \def\gl@p#1\to#2{\expandafter\gl@poff#1\gl@poff#1#2}
577 \long\def\gl@poff\#1#2\gl@poff#3#4{\gdef#4{#1}\gdef#3{#2}}
578
579 %
```

V Line counting

V.1 Choosing the system of lineation

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


```

\ifbypstart@ The \ifbypage@ and \ifbypstart@ flag specify the current lineation system:
\byppstart@true
\byppstart@false
  \ifbypage@
  \bypage@true
  \bypage@false
580 \newif\ifbypage@
581 \newif\ifbypstart@
582 %

```

- line-of-page: byppstart@ = false and bypage@ = true.
 - line-of-pstart: byppstart@ = true and bypage@ = false.
- reledmac will use the line-of-section system unless instructed otherwise.

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@R \newif\ifbypage@R
\ifbypstart@R \newif\ifbypstart@R
585 %

```

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

```

586 \newcommand*{\lineation}[1]{
587 %

```

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

```

588 \ifnumbering
589 \led@err@LineationInNumbered
590 \else
591 %

```

If the argument is page.

```

592 \def\@tempa{#1}\def\@tempb{page}%
593 \ifx\@tempa\@tempb
594 \global\byppage@true
595 \global\byppstart@false
596 \unless\ifnocritical@%
597 \Xpstart [] [false]%
598 \fi%
599 %

```

If the argument is pstart.

```

600 \else
601 \def\@tempb{pstart}%
602 \ifx\@tempa\@tempb
603 \global\byppage@false
604 \global\byppstart@true
605 \unless\ifnocritical@%
606 \Xpstart%
607 \fi%
608 %

```

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

```

609     \else
610         \def\@tempb{section}
611         \ifx\@tempa\@tempb
612             \global\bypage@false
613             \global\bystart@false
614             \unless\ifnocritical@%
615                 \Xpstart[] [false]%
616             \fi%
617 %

```

In other case, it is an error.

```

618         \else
619             \led@warn@BadLineation
620         \fi
621     \fi
622     \fi
623 \fi}}
624 %

```

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`

```

625 \newcount\line@margin%
626 \newcount\line@margin@columns%Only for parallel typesetting
627 \line@margin@columns=\m@ne%
628
629 \newcommand*\linenummargin}[1]{%
630     \l@dgetline@margin{#1}%
631     \ifnum\@l@tempcntb>\m@ne
632         \ifledRcol
633             \global\line@marginR=\@l@tempcntb
634             \led@warn@setting@in@rightside{\linenummargin}%
635         \else
636             \global\line@margin=\@l@tempcntb
637         \fi
638     \fi}}
639
640 \newcommand*\l@dgetline@margin}[1]{%
641     \def\@tempa{#1}\def\@tempb{left}%
642     \ifx\@tempa\@tempb
643         \@l@tempcntb \z@
644     \else
645         \def\@tempb{right}%
646         \ifx\@tempa\@tempb

```

```

647     \l@dttempcntb \@ne
648   \else
649     \def\@tempb{outer}%
650     \ifx\@tempa\@tempb
651       \l@dttempcntb \tw@
652     \else
653       \def\@tempb{inner}%
654       \ifx\@tempa\@tempb
655         \l@dttempcntb \thr@@
656       \else
657         \led@warn@BadLinenummargin
658         \l@dttempcntb \m@ne
659       \fi
660     \fi
661   \fi
662 \fi}
663
664 %

```

V.3 Line number initialization and increment

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

```

665 \newcounter{firstlinenum}
666   \setcounter{firstlinenum}{5}
667 \newcounter{linenumincrement}
668   \setcounter{linenumincrement}{5}
669 %

```

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

```

670 \newcounter{firstsublinenum}
671   \setcounter{firstsublinenum}{5}
672 \newcounter{sublinenumincrement}
673   \setcounter{sublinenumincrement}{5}
674
675 %

```

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

```

676
677 \newcommand*\firstlinenum[1]{%
678   \ifledRcol%
679     \setcounter{firstlinenumR}{#1}%

```

```

680 \led@warn@setting@in@rightside{\firstlinenum}%
681 \else%
682 \setcounter{firstlinenum}{#1}%
683 \fi%
684 }
685 \newcommand*\linenumincrement}[1]{%
686 \ifledRcol%
687 \setcounter{linenumincrementR}{#1}%
688 \led@warn@setting@in@rightside{\linenumincrement}%
689 \else%
690 \setcounter{linenumincrement}{#1}%
691 \fi%
692 }
693 \newcommand*\firstsublinenum}[1]{%
694 \ifledRcol%
695 \setcounter{firstsublinenumR}{#1}%
696 \led@warn@setting@in@rightside{\firstsublinenum}%
697 \else%
698 \setcounter{firstsublinenum}{#1}%
699 \fi%
700 }
701 \newcommand*\sublinenumincrement}[1]{%
702 \ifledRcol%
703 \setcounter{sublinenumincrementR}{#1}%
704 \led@warn@setting@in@rightside{\sublinenumincrement}%
705 \else%
706 \setcounter{sublinenumincrement}{#1}%
707 \fi%
708 }
709 %
710 %

```

V.4 Line number locking

`\lockdisp` When line locking is being used, the `\lockdisp{⟨word⟩}` macro specifies whether a line number—if one is due to appear—should be printed on the first printed line or on the last, or by all of them. Its argument is a word, either `first`, `last`, or `all`. Initially, it is set to `first`.

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

```

711 \newcount\lock@disp
712 \newcommand{\lockdisp}[1]{%
713 \l@dgetlock@disp{#1}%
714 \ifnum\l@dtempcntb>\m@ne
715 \global\lock@disp=\l@dtempcntb
716 \else
717 \led@warn@BadLockdisp
718 \fi}}
719 \newcommand*\l@dgetlock@disp}[1]{

```

```

720 \def\@tempa{#1}\def\@tempb{first}%
721 \ifx\@tempa\@tempb
722   \@l@dttempcntb \z@
723 \else
724   \def\@tempb{last}%
725   \ifx\@tempa\@tempb
726     \@l@dttempcntb \@ne
727   \else
728     \def\@tempb{all}%
729     \ifx\@tempa\@tempb
730       \@l@dttempcntb \tw@
731     \else
732       \@l@dttempcntb \m@ne
733     \fi
734   \fi
735 \fi}
736
737 %

```

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

```

738 \newcount\sublock@disp
739 \newcommand{\sublockdisp}[1]{%
740   \l@dgetlock@disp{#1}%
741   \ifnum\@l@dttempcntb>\m@ne
742     \global\sublock@disp=\@l@dttempcntb
743   \else
744     \led@warn@BadSublockdisp
745   \fi}}
746
747 %

```

V.5 Line number style

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

```

748 \newcommand*{\linenumberstyle}[1]{%
749   \def\linenumrep##1{\@nameuse{#1}{##1}}
750 \newcommand*{\sublinenumberstyle}[1]{%
751   \def\sublinenumrep##1{\@nameuse{#1}{##1}}
752 %

```

Initialise the number styles to arabic.

```

753 \linenumberstyle{arabic}
754 \let\linenumr@p\linenumrep
755 \sublinenumberstyle{arabic}
756 \let\sublinenumr@p\sublinenumrep
757
758 %

```

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 subtitle) number.

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

```

759 \newlength{\linenumsep}
760 \setlength{\linenumsep}{1pc}
761 \newcommand*{\numlabfont}{\normalfont\scriptsize}
762 \newcommand*{\ledlinenum}{%
763   \bgroup%
764   \ifluatex%
765     \texdir TLT%
766   \fi%
767   \numlabfont%
768   \ifdefstring{\Xlinenumannotationposition@side}{before}{%
769     \ifcvoid{annot@the\absline@num @the\section@num}%
770       {}%
771       {\Xwraplinenumannotation@side{%
772         \csuse{annot@the\absline@num @the\section@num}%
773       }}%
774   }%
775 }{}%
776 \linenumrep{\line@num}%
777 \ifsublines@
778   \ifnum\subline@num>0\relax
779     \unskip%
780     \Xsublinesep@side%
781     \sublinenumrep{\subline@num}%
782   \fi

```

```

783 \fi%
784 \ifdefstring{\Xlinenumannotationposition@side}{after}{%
785 \ifcvoid{annot@the\absline@num @the\section@num}%
786   }%
787   {\Xwraplinenumannotation@side{%
788     \csuse{annot@the\absline@num @the\section@num}%
789     }%
790   }%
791 }{}%
792 \egroup%
793 }%
794
795 \newcommand*{\leftlinenum}{%
796   \ledlinenum
797   \kern\linenumsep}
798 \newcommand*{\rightlinenum}{%
799   \kern\linenumsep
800   \ledlinenum}
801
802 %

```

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 \TeX 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`.

```

803 \newcount\line@num
804 %

```

`\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.

```

805 \newcount\subline@num
806 %

```

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

`\sublines@true`

`\sublines@false`

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

```
807 \newif\ifsublines@
808 %
```

`\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.

```
809 \newcount\absline@num
810 %
```

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

V.8 Line number locking counter

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

`\sub@lock`

```
811 \newcount\@lock
812 \newcount\sub@lock
813 %
```

V.9 Line number associated to lemma

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

`\insertlines@list`

`\actionlines@list`

`\actions@list`

`\annot@list`

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

1. the starting page,
2. line, and
3. sub-line numbers, followed by the

4. ending page,
5. line, and
6. sub-line numbers, and then the
7. font specifier for the lemma.

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

```
23|35|0|24|3|0|0T1/cm/r/m/n.
```

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

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

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

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

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

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

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

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

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

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

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

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

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

Here are the commands to create these lists:

```

814 \list@create{\line@list}
815 \list@create{\insertlines@list}
816 \list@create{\actionlines@list}
817 \list@create{\actions@list}
818 \list@create{\annot@list}%
819 %

```

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

`\endpage@num`

`\endline@num`

`\endsubline@num`

```

820 \newcount\page@num
821 \newcount\endpage@num
822 \newcount\endline@num
823 \newcount\endsubline@num
824 %

```

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

`\noteschanged@true`

`\noteschanged@false`

```

825 \newif\ifnoteschanged@
826 %

```

`\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@27 \newcommand*\resetprevline@{%
828   \def\do##1{\global\csundef{prevline##1}}%
829   \dolistloop{\@series}%
830 }
831 %

```

`\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@32 \newcommand*\resetprevpage@num{%
833   \def\do##1{%
834     \ifcsdef{prevpage##1@num}{%
835       \global\csname prevpage##1@num\endcsname=\z@%
836       \global\csname prevpage##1@numR\endcsname=\z@%
837     }%
838     {}%
839     \ifcsdef{##1prevpage@num}{%

```

```

840     \global\csname ##1prevpage@num\endcsname=\z@%
841     \global\csname ##1prevpage@numR\endcsname=\z@%
842     }%
843     {}%
844     }%
845     \dolistloop{\@series}%
846 }
847 %

```

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.

```

848 \newread\@inputcheck
849 \newcommand*{\read@linelist}[1]{%
850   \ifledRcol%
851     \list@clearing@regR%
852   \else%
853     \list@clearing@reg%
854   \fi%
855 %

```

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).

```

856   \list@clear{\maxlinesinpar@list}
857 %

```

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 `\resumenumbersing`, those things should still have the values they had when `\pausenumbersing` was executed.

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

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

```

858   \get@linelistfile{#1}%
859   \@stopmsd%Security if last \endms{} is forgotten
860   \unless\ifledRcol%Get the last line of the last page
861     \csnumgdef{\@lastabslines@forpage@the\page@num}{\the\absline@num}%

```

```

862 \csnumgdef{@lastline@forpage@the\page@num}{\the\line@num}%
863 \else%
864 \csnumgdef{@lastabsline@forpageR@the\page@numR}{\the\absline@numR}%
865 \csnumgdef{@lastline@forpageR@the\page@numR}{\the\line@numR}%
866 \fi%
867 \endgroup
868 %

```

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.

```

869 \ifledRcol
870 \global\page@numR=\m@ne
871 \ifx\actionlines@listR\empty
872 \gdef\next@actionlineR{1000000}%
873 \else
874 \glp\actionlines@listR\to\next@actionlineR
875 \glp\actions@listR\to\next@actionR
876 \fi
877 \else
878 \global\page@num=\m@ne
879 \ifx\actionlines@list\empty
880 \gdef\next@actionline{1000000}%
881 \else
882 \glp\actionlines@list\to\next@actionline
883 \glp\actions@list\to\next@action
884 \fi
885 \fi
886 }
887 %

```

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

```

888 \newcommand*{\list@clearing@reg}{%
889 \list@clear{\line@list}%
890 \list@clear{\insertlines@list}%
891 \list@clear{\actionlines@list}%
892 \list@clear{\actions@list}%
893 \list@clear{\linesinpar@listL}%
894 \list@clear{\linesonpage@listL}%
895 }%
896 %

```

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

```

897 \newcommand*{\get@linelistfile}[1]{%
898 \InputIfFileExists{\l@auxdir#1}{%

```

```

899 \global\noteschanged@false
900 \begingroup
901   \catcode`\[=1 \catcode`\]=2
902   \makeatletter \catcode`\^M=9}{%
903 \led@warn@NoFile{\l@auxdir#1}%
904 \global\noteschanged@true
905 \begingroup}%
906 }
907
908 %

```

This version of `\read@linelist` creates list macros containing data for the entire section, so they could get rather large. It would be no more difficult to read the line-list file incrementally rather than all at once: we could read, at the start of each paragraph, only the commands relating to that paragraph. But this would require that we have two line-lists open at once, one for reading, one for writing, and on systems without version numbers we would have to do some file renaming outside of \LaTeX for that to work. We have retained this slower approach to avoid that sort of hacking about, but have provided the `\pausenumbering` and `\resumenumbering` macros to help you if you run into macro memory limitations (see 5.2.7 p. 20 above).

V.11 Commands within the line-list file

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

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

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

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

```

909 \newcommand{\line@list@version}[1]{%
910   \IfStrEq{#1}{\this@line@list@version}%
911   {}%
912   {\ifledRcol%
913     \led@warn@Obsolete{\jobname.\extensionchars\the\section@num}%
914     \else%
915     \led@warn@Obsolete{\jobname.\extensionchars\the\section@num}%
916     \fi%

```

```

917 \endinput%
918 }%
919 }%
920 %

```

`\@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 `\fix@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`.

```

921
922 \newcommand*{\@nl}[2]{%
923   \fix@page{#1}%
924   \ifledRcol%
925     \@nl@regR%
926   \else%
927     \@nl@reg%
928   \fi%
929 }
930 \newcommand*{\@nl@reg}{%
931   \ifx\l@dchset@num\relax \else
932     \reset@current@annot%
933     \advance\absline@num \@ne
934     \csgdef{l@dchset@num@the\absline@num}{%To remember this line have
been marked by a \setlinenum
935     \set@line@action
936     \let\l@dchset@num=\relax
937     \advance\absline@num \m@ne
938     \advance\line@num \m@ne
939   \fi
940 %

```

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

```

941 \advance\absline@num \@ne
942   \ifx\next@page@num\relax \else
943     \page@action
944     \let\next@page@num=\relax
945   \fi

```

```

946     \ifx\sub@change\relax \else
947         \ifnum\sub@change>\z@
948             \sublines@true
949         \else
950             \sublines@false
951         \fi
952     \sub@action
953     \let\sub@change=\relax
954 \fi
955 %

```

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

```

956     \ifcase\@lock
957         \or
958             \@lock \tw@
959         \or \or
960             \@lock \z@
961     \fi
962     \ifcase\sub@lock
963         \or
964             \sub@lock \tw@
965         \or \or
966             \sub@lock \z@
967     \fi
968 %

```

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

```

969     \ifsublines@
970         \ifnum\sub@lock<\tw@
971             \advance\subline@num \@ne
972         \fi
973     \else
974         \ifnum\@lock<\tw@
975             \advance\line@num \@ne \subline@num \z@
976         \fi
977     \fi}
978
979 %

```

`\last@page@num` `\fix@page` basically replaces `\@page`. It determines whether or not a new page has been started, based on the page values held by `\@n1`.

```

980 \newcount\last@page@num
981 \last@page@num=-10000
982
983 \newcommand*{\fix@page}[1]{%
984     \ifledRcol
985         \ifnum #1=\last@page@numR
986         \else

```



```

987 \csnumgdef{@lastabsline@forpage@the\page@numR}{\the\absline@numR}%
988 \csnumgdef{@lastline@forpage@the\page@numR}{\the\line@numR}%
989 \ifbypage@R
990 \ifx\l@dchset@num\relax%Not resetting if preceded by a \setlinenum
991 \ifboolexpr{%
992   bool{resumenumberingR@start}%
993   and test {\ifnumequal{\last@page@numR}{-10000}}%
994 }%
995 }%
996 {%
997   \line@numR \z@%
998   \subline@numR \z@%
999   \global\csdef{reset@lineR\the\numexpr\absline@numR+1\relax @\
the\section@numR}{}%
1000 }%
1001 \fi%
1002 \fi
1003 \global\page@numR=#1%
1004 \global\last@page@numR=#1%
1005 \global\def\next@page@numR{#1}%
1006 \fi
1007 \else
1008 \ifnum #1=\last@page@num
1009 \else
1010 \csnumgdef{@lastabsline@forpage@the\page@num}{\the\absline@num}%
1011 \csnumgdef{@lastline@forpage@the\page@num}{\the\line@num}%
1012 \ifbypage@
1013 \ifx\l@dchset@num\relax%Not resetting if preceded by a \setlinenum
1014 \line@num \z@ \subline@num \z@%
1015 \global\csdef{reset@line\the\numexpr\absline@num+1\relax @the\
section@num}{}%
1016 \fi%
1017 \fi
1018 \global\page@num=#1%
1019 \global\last@page@num=#1%
1020 \global\def\next@page@num{#1}%
1021 \listxadd{\normal@page@break}{\the\absline@num}
1022 \fi
1023 \fi}
1024 %

```

`\@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.

```

1025 \@lopR \newcommand*{\@pend}[1]{}
1026 \newcommand*{\@pendR}[1]{}
1027 \newcommand*{\@lopL}[1]{}
1028 \newcommand*{\@lopR}[1]{}
1029

```

```
1030 %
```

\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 `\@nl` of the necessary action.

\sub@off

```
1031 \newcommand*\sub@on{\ifsublines@
1032   \let\sub@change=\relax
1033   \else
1034   \def\sub@change{1}%
1035   \fi}
1036 \newcommand*\sub@off{\ifsublines@
1037   \def\sub@change{-1}%
1038   \else
1039   \let\sub@change=\relax
1040   \fi}
1041
1042 %
```

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

```
1043
1044 \newcommand*\@adv}[1]{%
1045   \ifsublines@
1046   \ifledRcol
1047     \advance\subline@numR by #1\relax
1048     \ifnum\subline@numR<\z@
1049       \led@warn@BadAdvancelineSubline
1050       \subline@numR \z@
1051     \fi
1052   \else
1053     \advance\subline@num by #1\relax
1054     \ifnum\subline@num<\z@
1055       \led@warn@BadAdvancelineSubline
1056       \subline@num \z@
1057     \fi
1058   \fi
1059   \else
1060   \ifledRcol
1061     \advance\line@numR by #1\relax
1062     \ifnum\line@numR<\z@
1063       \led@warn@BadAdvancelineLine
1064       \line@numR \z@
1065     \fi
1066   \else
1067     \advance\line@num by #1\relax
1068     \ifnum\line@num<\z@
1069       \led@warn@BadAdvancelineLine
1070       \line@num \z@
```

```

1071     \fi
1072     \fi
1073     \fi
1074     \set@line@action}
1075
1076 %

```

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

```

1077
1078 \newcommand*\@set}[1]{%
1079     \ifledRcol
1080         \ifsublines@
1081             \subline@numR=#1\relax
1082         \else
1083             \line@numR=#1\relax
1084         \fi
1085         \set@line@action
1086     \else
1087         \ifsublines@
1088             \subline@num=#1\relax
1089         \else
1090             \line@num=#1\relax
1091         \fi
1092         \set@line@action
1093     \fi}
1094
1095 %

```

\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 `\@n1?` macro. If it is not `\relax` then a linenum change is to be done.

```

1096
1097 \newcommand*\l@d@set}[1]{%
1098     \ifledRcol
1099         \line@numR=#1\relax
1100         \advance\line@numR \@ne
1101         \def\l@dchset@num{#1}
1102     \else
1103         \line@num=#1\relax
1104         \advance\line@num \@ne
1105         \def\l@dchset@num{#1}
1106     \fi}
1107 \let\l@dchset@num\relax
1108
1109 %

```

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

```

1110
1111 \newcommand*{\page@action}{%
1112   \ifledRcol
1113     \xright@appenditem{\the\absline@numR}\to\actionlines@listR
1114     \xright@appenditem{\next@page@numR}\to\actions@listR
1115   \else
1116     \xright@appenditem{\the\absline@num}\to\actionlines@list
1117     \xright@appenditem{\next@page@num}\to\actions@list
1118   \fi}
1119 %

```

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

```

1120
1121 \newcommand*{\set@line@action}{%
1122   \ifledRcol
1123     \xright@appenditem{\the\absline@numR}\to\actionlines@listR
1124     \ifsublines@
1125       \@l@tempcnta=-\subline@numR
1126     \else
1127       \@l@tempcnta=-\line@numR
1128     \fi
1129     \advance\@l@tempcnta by -5000\relax
1130     \xright@appenditem{\the\@l@tempcnta}\to\actions@listR
1131   \else
1132     \xright@appenditem{\the\absline@num}\to\actionlines@list
1133     \ifsublines@
1134       \@l@tempcnta=-\subline@num
1135     \else
1136       \@l@tempcnta=-\line@num
1137     \fi
1138     \advance\@l@tempcnta by -5000\relax
1139     \xright@appenditem{\the\@l@tempcnta}\to\actions@list
1140   \fi}
1141 %

```

`\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.

```

1142
1143 \newcommand*{\sub@action}{%
1144   \ifledRcol
1145     \xright@appenditem{\the\absline@numR}\to\actionlines@listR
1146     \ifsublines@
1147       \xright@appenditem{-1001}\to\actions@listR
1148     \else
1149       \xright@appenditem{-1002}\to\actions@listR

```

```

1150 \fi
1151 \else
1152 \xright@appenditem{\the\absline@num}\to\actionlines@list
1153 \ifsublines@
1154 \xright@appenditem{-1001}\to\actions@list
1155 \else
1156 \xright@appenditem{-1002}\to\actions@list
1157 \fi
1158 \fi}
1159 %

```

`\lock@on` `\lock@on` adds an entry to the action-code list to turn line number locking on. The current setting of the sub-lineation flag tells us whether this applies to line numbers or sub-line numbers.

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.

```

1160 \newcommand*{\lock@on}{\futurelet\next\do@lockon}
1161
1162 \newcommand*{\do@lockon}{%
1163 \ifx\next\lock@off
1164 \global\let\lock@off=\skip@lockoff
1165 \else
1166 \ifledRcol
1167 \do@lockonR
1168 \else
1169 \do@lockonL
1170 \fi
1171 \fi}
1172
1173
1174 \newcommand*{\do@lockonL}{%
1175 \xright@appenditem{\the\absline@num}\to\actionlines@list
1176 \ifsublines@
1177 \xright@appenditem{-1005}\to\actions@list
1178 \ifnum\sub@lock=\z@
1179 \sub@lock \@ne
1180 \else
1181 \ifnum\sub@lock=\thr@@
1182 \sub@lock \@ne
1183 \fi
1184 \fi
1185 \else
1186 \xright@appenditem{-1003}\to\actions@list
1187 \ifnum\@lock=\z@
1188 \@lock \@ne
1189 \else

```

```

1190     \ifnum \@lock=\thr@@
1191         \@lock \@ne
1192     \fi
1193 \fi
1194 \fi}
1195
1196 %

```

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

```

\do@lockoff
\do@lockoffL
\skip@lockoff
1197 \newcommand*{\do@lockoffL}{%
1198     \xright@appenditem{\the\absline@num}\to\actionlines@list
1199     \ifsublines@
1200         \xright@appenditem{-1006}\to\actions@list
1201         \ifnum\sub@lock=\tw@
1202             \sub@lock \thr@@
1203         \else
1204             \sub@lock \z@
1205         \fi
1206     \else
1207         \xright@appenditem{-1004}\to\actions@list
1208         \ifnum \@lock=\tw@
1209             \@lock \thr@@
1210         \else
1211             \@lock \z@
1212         \fi
1213     \fi}
1214
1215 \newcommand*{\do@lockoff}{%
1216     \ifledRcol
1217         \do@lockoffR
1218     \else
1219         \do@lockoffL
1220     \fi}
1221 \newcommand*{\skip@lockoff}{\global\let\lock@off=\do@lockoff}
1222 \global\let\lock@off=\do@lockoff
1223
1224 %

```

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

```

1225 \newcommand*{\n@num}{%
1226     \ifledRcol%
1227         \xright@appenditem{\the\absline@num}\to\actionlines@listR
1228         \xright@appenditem{-1007}\to\actions@listR
1229     \else%
1230         \xright@appenditem{\the\absline@num}\to\actionlines@list%
1231         \xright@appenditem{-1007}\to\actions@list%
1232     \fi%

```

```
1233 }%
1234
1235 %
```

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

```
1236 \newcommand*{\n@num@stanza}{%
1237   \ifledRcol%
1238     \xright@appenditem{\the\absline@numR}\to\actionlines@listR%
1239     \xright@appenditem{-1008}\to\actions@listR%
1240   \else%
1241     \xright@appenditem{\the\absline@num}\to\actionlines@list%%
1242     \xright@appenditem{-1008}\to\actions@list%
1243   \fi%
1244 }
1245 %
```

`\ifl@dhidnumber` `\hidnumbering` hides number in margin. It uses action code 1009. `\hidnumberingonleftpage` and `\hidnumberingonrightpage` are variants, using action code only conditionally

`\hidnumberingonleftpage`
`\hidnumberingonrightpage`

```
1246 \newif\ifl@dhidnumber
1247 \newcommand*{\hidnumbering}{
1248   \ifledRcol%
1249     \write\linenum@outR{\string\hide@num}%
1250   \else%
1251     \write\linenum@out{\string\hide@num}%
1252   \fi%
1253 }%
1254 \newcommand*{\hide@num}{%
1255   \ifledRcol%
1256     \xright@appenditem{\the\absline@numR}\to\actionlines@listR%
1257     \xright@appenditem{-1009}\to\actions@listR%
1258   \else%
1259     \xright@appenditem{\the\absline@num}\to\actionlines@list%%
1260     \xright@appenditem{-1009}\to\actions@list%
1261   \fi%
1262 }
1263 \newcommand*{\hidnumberingonleftpage}{%
1264   \ifledRcol%
1265     \write\linenum@outR{\string\hide@num@left}%
1266   \else%
1267     \write\linenum@out{\string\hide@num@left}%
1268   \fi%
1269 }%
1270
1271 \newcommand*{\hide@num@left}{%
1272   \ifledRcol%
1273     \ifodd\page@numR\else%
```

```

1274 \xright@appenditem{\the\absline@numR}\to\actionlines@listR%
1275 \xright@appenditem{-1009}\to\actions@listR%
1276 \fi%
1277 \else%
1278 \ifodd\page@num\else%
1279 \xright@appenditem{\the\absline@num}\to\actionlines@list%%
1280 \xright@appenditem{-1009}\to\actions@list%
1281 \fi%
1282 \fi%
1283 }%
1284
1285 \newcommand*\hidenumberingonrightpage{%
1286 \ifledRcol%
1287 \write\linenum@outR{\string\hide@num@right}%
1288 \else%
1289 \write\linenum@out{\string\hide@num@right}%
1290 \fi%
1291 }%
1292
1293 \newcommand*\hide@num@right{%
1294 \ifledRcol%
1295 \ifodd\page@numR%
1296 \xright@appenditem{\the\absline@numR}\to\actionlines@listR%
1297 \xright@appenditem{-1009}\to\actions@listR%
1298 \fi%
1299 \else%
1300 \ifodd\page@num%
1301 \xright@appenditem{\the\absline@num}\to\actionlines@list%%
1302 \xright@appenditem{-1009}\to\actions@list%
1303 \fi%
1304 \fi%
1305 }%
1306
1307 %

```

`\@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`.

```

1308 \newcount\insert@count
1309 %

```

- #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.

```
1310 \newcommand*\@dummy@ref}[2]{#2}
1311 %
```

`\@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.

```
1312 \newcommand*\@ref}[2]{%
1313   \ifledRcol%
1314     \@ref@regR{#1}{#2}%
1315   \else%
1316     \@ref@reg{#1}{#2}%
1317   \fi%
1318 }%
1319 \newcommand*\@ref@reg}[2]{%
1320   \global\insert@count=#1\relax
1321   \global\advance\@edtext@level by 1%
1322   \loop\ifnum\insert@count>\z@
1323     \xright@appenditem{\the\absline@num}\to\insertlines@list
1324     \global\advance\insert@count \m@ne
1325   \repeat
1326 %
```

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.

```
1327 \begingroup
1328   \let\@ref=\@dummy@ref
1329   \let\@lopL\@gobble
1330   \let\page@action=\relax
1331   \let\sub@action=\relax
1332   \let\set@line@action=\relax
1333   \let\@lab=\relax
1334   \let\@lemma=\relax%
1335   \let\@sw\@gobblethree%
1336   \let\store@annot@to@absline\@gobble%
1337   #2
1338   \global\endpage@num=\page@num
1339   \global\endline@num=\line@num
1340   \global\endsubline@num=\subline@num
1341   \global\let\endcurrent@annot=\current@annot%
1342 \endgroup
1343 %
```

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

```

1344 \xright@appenditem%
1345   {\the\page@num|\the\line@num|
1346    \ifsublines@ \the\subline@num \else 0\fi|
1347    \the\endpage@num|\the\endline@num|
1348    \ifsublines@ \the\endsubline@num \else 0\fi}\to\line@list
1349 \xright@appenditem%
1350   {\current@annot|\endcurrent@annot}\to\annot@list%
1351 %

```

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

```

1352   \@ref@reg@parse{#2}%
1353 %

```

Decrease edtext level counter.

```

1354   \global\advance\@edtext@level by -1%
1355 }
1356 %

```

`\@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.

```

1357 \newcommand{\@ref@reg@parse}[1]{%
1358   \expandafter\list@create\expandafter{\csname sw@list@edtext@tmp@the\
@edtext@level\endcsname}%
1359   \providebool{lemmacommand@the\@edtext@level}%
1360   \boolfalse{lemmacommand@the\@edtext@level}%
1361 %

```

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

```

1362   #1%
1363 %

```

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.

```

1364   \ifnum\@edtext@level>0%
1365     \def\create@this@edtext@level{\expandafter\list@create\expandafter{\
csname sw@list@edtext@the\@edtext@level\endcsname}}%
1366     \ifcsundef{sw@list@edtext@the\@edtext@level}{\create@this@edtext@level
}\}%
1367     \letcs{\@tmp}{sw@list@edtext@the\@edtext@level}%
1368     \letcs{\@tmpp}{sw@list@edtext@tmp@the\@edtext@level}
1369     \xright@appenditem{\expandonce\@tmpp}\to\@tmp%
1370     \global\cset{sw@list@edtext@the\@edtext@level}{\@tmp}%
1371     \fi%
1372 %

```

```
1373 }
1374
1375 %
```

`\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.

```
1376 \newcommand{\@ref@later}[1]{%
1377   \global\advance\@edtext@level by \@ne%
1378   \ifledRcol%
1379     \@ref@reg@parseR{#1}%
1380   \else%
1381     \@ref@reg@parse{#1}%
1382   \fi%
1383   \global\advance\@edtext@level by -\@ne%
1384 }%
1385 %
```

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`.

```
1386 \newwrite\linenum@out
1387 %
```

`\iffirst@linenum@out@`
`\first@linenum@out>true`
`\first@linenum@out>false`

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

We can manage pretty well by means of the `\iffirst@linenum@out@` flag; its inelegant name suggests the nature of the problem that made its creation necessary. It is set to be `true` before any `\linenum@out` file is opened. When such a file is opened for the first time, it is done using `\immediate`, so that it will at once be safe for the output routine to write to it; we then set this flag to `false`.

```
1388 \newif\iffirst@linenum@out@
1389   \first@linenum@out@true
1390 %
```

`\this@line@list@version` The commands allowed in the line-list file and their arguments can change between two versions of `reledmac`. The `\this@line@list@version` command is upgraded when

it happens. It is written in the file list. If we process a line-list file which used a older version, that means the commands used inside are deprecated, and we can't use them.

```
1391 \newcommand{\this@line@list@version}{7}%
1392 %
```

`\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.

```
1393 \let\next@line@list@stuff\relax%
1394 \newcommand*{\line@list@stuff}[1]{%
1395 %
```

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

```
1396 \global\newtoggle{notfirstrun@#1}%
1397 \IfFileExists{\l@auxdir#1}%
1398   {\global\toggletrue{notfirstrun@#1}}%
1399   {\global\togglefalse{notfirstrun@#1}}%
1400 %
```

A internal hook (not used yet).

```
1401 \next@line@list@stuff%
1402 \global\let\next@line@list@stuff\relax%
1403 %
```

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

```
1404 \read@linelist{#1}%
1405 %
```

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 .

```
1406 \iffirst@linenum@out@
1407   \global\first@linenum@out@false%
1408   \immediate\openout\linenum@out=\l@auxdir#1\relax%
1409   \immediate\write\linenum@out{\string\line@list@version{\
this@line@list@version}}%
1410   \ifl@dpaging%
1411     \immediate\write\linenum@out{\string\@par@sync@option{\
@par@this@sync@option}}%
1412   \fi%
1413 \else
1414 %
```

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.

```

1415 \if@minipage%
1416 \leavevmode%
1417 \fi%
1418 \closeout\linenum@out%
1419 \openout\linenum@out=\l@auxdir#1\relax%
1420 \write\linenum@out{\string\line@list@version{\this@line@list@version}}
%
1421 \ifl@dpaging%
1422 \write\linenum@out{\string\par@sync@option{\@par@this@sync@option}}
%
1423 \fi%
1424 \fi%
1425 }%
1426
1427 %

```

`\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.

```

1428 \newcommand*{\new@line}{%
1429 \IfStrEq{\led@pb@setting}{after}%
1430 {\xifinlist{the\absline@num}{\l@prev@nopb}%
1431 {\xifinlist{the\absline@num}{\normal@page@break}%
1432 {\numdef{\@next@page}{\c@page+\@ne}%
1433 \write\linenum@out{\string\@nl[\@next@page][\@next@page]}}%
1434 }%
1435 {\write\linenum@out{\string\@nl[the\c@page][thepage]}}%
1436 }%
1437 {\write\linenum@out{\string\@nl[the\c@page][thepage]}}%
1438 }%
1439 \IfStrEq{\led@pb@setting}{before}%
1440 {\numdef{\next@absline}{the\absline@num+\@ne}%
1441 \xifinlist{next@absline}{\l@prev@nopb}%
1442 {\xifinlist{the\absline@num}{\normal@page@break}%
1443 {\numdef{\nc@page}{\c@page+\@ne}%
1444 \write\linenum@out{\string\@nl[\nc@page][\nc@page]}}%
1445 }%
1446 {\write\linenum@out{\string\@nl[the\c@page][thepage]}}%
1447 }%
1448 {\write\linenum@out{\string\@nl[the\c@page][thepage]}}%
1449 }%
1450 }%
1451 \IfStrEqCase{\led@pb@setting}{{before}{\relax}{after}{\relax}}{\write\
linenum@out{\string\@nl[the\c@page][thepage]}}%
1452 }
1453
1454 %

```

`\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.

```

1455 \newif\if@noneed@Footnote%
1456
1457 \newcommand*{\flag@start}{%
1458   \ifledRcol%
1459     \edef\next{\write\linenum@outR{%
1460       \string\@ref[\the\insert@countR] []}}%
1461     \next%
1462     \ifnum\insert@countR<1%
1463       \if@noneed@Footnote\else%
1464         \led@err@EdtextWithoutFootnote%
1465       \fi%
1466     \fi%
1467   \else%
1468     \edef\next{\write\linenum@out{%
1469       \string\@ref[\the\insert@count] []}}%
1470     \next%
1471     \ifnum\insert@count<1%
1472       \if@noneed@Footnote\else%
1473         \led@err@EdtextWithoutFootnote%
1474       \fi%
1475     \fi%
1476   \fi}%
1477
1478 \newcommand*{\flag@end}{%
1479   \ifledRcol%
1480     \write\linenum@outR[]}%
1481   \else%
1482     \write\linenum@out[]}%
1483   \fi}%
1484
1485
1486 %

```

`\flag@start@RTL` With $X_{\text{L}}\text{T}_{\text{E}}\text{X}$, there is a problem when using RTL: the writing of a command in the numbered auxiliary files (.1, .2 etc) is reversed when the first argument of `\edtext` is typeset in one line, but it is **not** reversed when this first argument is typeset in two lines or more.²⁸

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:

²⁸This problem is caused by the way $X_{\text{L}}\text{T}_{\text{E}}\text{X}$ manages right-to-left typesetting. David Carlisle explains it on <http://tex.stackexchange.com/a/333373/7712> and provides a potential solution, using `\vadjust`. However in some cases this adds spurious vertical spaces in `reledmac`. That is why we are using the solution explained below.

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

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

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

```

1487 \newcommand{\flag@start@RTL}{%
1488   \edlabel{edtext:start:\csuse{thisedtext@the\@edtext@level}}%
1489   \IfStrEq{\xabslineref{edtext:start:\csuse{thisedtext@the\@edtext@level
}}}%
1490     {000}%
1491     {}%
1492     {%
1493     \ifnumequal%
1494       {\xabslineref{edtext:start:\csuse{thisedtext@the\@edtext@level}}}%
1495       {\xabslineref{edtext:end:\csuse{thisedtext@the\@edtext@level}}}%
1496       {\flag@end}%
1497       {\flag@start}%
1498     }%
1499   }%
1500
1501 \newcommand{\flag@end@RTL}{%
1502   \edlabel{edtext:end:\csuse{thisedtext@the\@edtext@level}}%
1503   \IfStrEq{\xabslineref{edtext:start:\csuse{thisedtext@the\@edtext@level
}}}%
1504     {000}%
1505     {}%
1506     {%
1507     \ifnumequal%
1508       {\xabslineref{edtext:start:\csuse{thisedtext@the\@edtext@level}}}%
1509       {\xabslineref{edtext:end:\csuse{thisedtext@the\@edtext@level}}}%
1510       {\flag@start}%
1511       {\flag@end}%
1512     }%
1513   }%
1514   %

```

`\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

```

1515 \newcommand*{\flag@start@later}{%
1516   \ifledRcol%
1517     \write\linenum@outR{\string\@ref@later []}%
1518   \else%

```

```

1519 \write\linenum@out{\string\@ref@later{}}%
1520 \fi%
1521 }%
1522 \newcommand{\flag@end@later}{%
1523 \ifledRcol%
1524 \write\linenum@outR{}}%
1525 \else%
1526 \write\linenum@out{}}%
1527 \fi%
1528 }
1529 %

```

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

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

```

1530
1531
1532 \newcommand*{\startsub}{\dimen0\lastskip
1533 \ifdim\dimen0>Opt \unskip \fi
1534 \ifledRcol \write\linenum@outR{\string\sub@on}%
1535 \else \write\linenum@out{\string\sub@on}%
1536 \fi
1537 \ifdim\dimen0>Opt \hskip\dimen0 \fi}
1538 \def\endsub{\dimen0\lastskip
1539 \ifdim\dimen0>Opt \unskip \fi
1540 \ifledRcol \write\linenum@outR{\string\sub@off}%
1541 \else \write\linenum@out{\string\sub@off}%
1542 \fi
1543 \ifdim\dimen0>Opt \hskip\dimen0 \fi}
1544
1545 %

```

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

```

1546 \newcommand*{\advanceline}[1]{\leavevmode%
1547 \ifledRcol \write\linenum@outR{\string\adv[#1]}%
1548 \else \write\linenum@out{\string\adv[#1]}%
1549 \fi%

```



```
1550 }
1551 %
```

\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.

```
1552
1553 \newcommand*\setline}[1]{%
1554   \leavevmode%
1555   \ifnum#1<\z@
1556     \led@warn@BadSetline%
1557   \else%
1558     \ifledRcol%
1559       \write\linenum@outR{\string\@set[#1]}%
1560     \else%
1561       \write\linenum@out{\string\@set[#1]}%
1562     \fi%
1563   \fi%
1564 }%
1565 %
1566 %
```

\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.

```
1567
1568 \newcommand*\setlinenum}[1]{%
1569   \ifnum#1<\z@
1570     \led@warn@BadSetlinenum
1571   \else
1572     \ifledRcol \write\linenum@outR{\string\l@d@set[#1]}
1573     \else      \write\linenum@out{\string\l@d@set[#1]} \fi
1574   \fi}
1575 %
1576 %
```

\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.

```
1577
1578 \newcommand*\startlock}{%
1579   \ifledRcol \write\linenum@outR{\string\lock@on}%
1580   \else      \write\linenum@out{\string\lock@on}%
1581   \fi}
1582 \def\endlock{%
1583   \ifledRcol \write\linenum@outR{\string\lock@off}%
1584   \else      \write\linenum@out{\string\lock@off}%
1585   \fi}
1586 %
```

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

```

\ifl@dskipnumber
\l@dskipversenumber
\l@dskipnumbertrue
\l@dskipnumberfalse
\skipnumbering
1587 \newif\ifl@dskipnumber
1588 \newif\ifl@dskipversenumber%
1589 \newcommand*{\skipnumbering}{%
1590   \leavevmode%
1591   \ifledRcol%
1592   \ifinstanza%
1593     \write\linenum@outR{\string\n@num@stanza}%
1594   \else%
1595     \write\linenum@outR{\string\n@num}%
1596   \fi%
1597   \advanceline{-1}%
1598 \else%
1599   \ifinstanza%
1600     \write\linenum@out{\string\n@num@stanza}%
1601   \else%
1602     \write\linenum@out{\string\n@num}%
1603   \fi%
1604   \advanceline{-1}%
1605 \fi%
1606 }%
1607
1608 %

```

VI Marking text for notes

The `\edtext` macro is used to create all footnotes and endnotes, as well as to print the portion of the main text to which a given note or notes is keyed. The idea is to have that lemma appear only once in the `.tex` file: all instances of it in the main text and in the notes are copied from that one appearance.

The `\edtext` macro takes two arguments.

```
\edtext{#1}{#2}
```

- `#1` is the piece of the main text being glossed; it gets added to the main text, and is also used as a lemma for notes to it.
- `#2` is a series of subsidiary macros that generate various kinds of notes.

The `\edtext` macro may be used (somewhat) recursively; that is, `\edtext` may be used within its own first argument. The code would be much simpler without this feature, but nested notes will commonly be necessary: it is quite likely that we will have an explanatory note for a long passage and notes on variants for individual words within that passage. The situation we can't handle is overlapping notes that are not nested: for example, one note covering lines 10–15, and another covering 12–18. You can handle such cases by using the `\lemma` and `\linenum` macros within `#2`: they alter the copy

of the lemma and the line numbers that are passed to the notes, and hence allow you to overcome any limitations of this system, albeit with extra effort.

The recursive operation of `\edtext` will fail if you try to use a copy that is called something other than `\edtext`. In order to handle recursion, `\edtext` needs to redefine its own definition temporarily at one point, and that does not work if the macro you are calling is not actually named `\edtext`. There is no problem as long as `\edtext` is not invoked in the first argument. If you want to call `\edtext` something else, it is best to create instead a macro that expands to an invocation of `\edtext`, rather than copying `\edtext` and giving it a new name; otherwise you will need to add an appropriate definition for your new macro to `\morenoexpands`.

Side effects of our line-numbering code make it impossible to use the usual footnote macros directly within a paragraph whose lines are numbered (see comments to `\do@line`, VII.2.1 p. 153). Instead, the appropriate note-generating command is appended to the list macro `\inserts@list`, and when `\pend` completes the paragraph it inserts all the notes at the proper places.

Note that we do not provide previous-note information, although it is often wanted; your own macros must handle that. We 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 accommodate 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.

```

1609 \list@create{\end@lemmas}
1610 %

```

`\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.

```

1611 \newcommand{\dummy@edtext}[2]{#1}
1612 %

```

`\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`.

```

1613 \newcommand{\dummy@edtext@showlemma}[2]{\showlemma{#1}}%
1614 %

```

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

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

The first class is font-changing macros. We suppress expansion for them by letting them become equal to zero.²⁹ This is done because we want to pass into our notes the generic commands to change to roman or whatever, and not their expansions that will ask for a particular style at a specified size. The notes may well be in a smaller font, so the command should be expanded later, when the note’s environment is in effect.

A second sort to turn off includes a few of the accent macros. Most are not a problem: an accent that is expanded to an `\accent` command may be harder to read but it works just the same. The ones that cause problems are: those that use alignments— \TeX seems to get confused about the difference between alignment parameters and macro parameters; those that use temporary control sequences; and those that look carefully at what the current font is.

(The `\copyright` macro defined in PLAIN \TeX has this sort of problem as well, but is not used enough to bother with. That macro, and any other that causes trouble, will get by all right if you put a `\protect` in front of it in your file.)

We also need to eliminate all `reledmac` macros like `\edlabel` and `\setline` that write things to auxiliary files: that writing should be done only once. And we make `\edtext` itself, if it appears within its own argument, do nothing but copy its first argument.

Finally, we execute `\morenoexpands`. The version of `\morenoexpands` defined here does nothing; but you may define a version of your own when you need to add more expansion suppressions as needed with your macros. That makes it possible to make such additions without needing to copy or modify the standard `reledmac` code. If you define your own `\morenoexpands`, you must be very careful about spaces: if the

²⁹Since ‘control sequences equivalent to characters are not expandable’—*The TeXbook*, answer to Exercise 20.14.

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_TE_X or Xe_LA_TE_X.)

```

1615 \newcommand*{\no@expands}{%
1616   \let\select@lemmafnt=0%
1617   \let\startsub=\relax \let\endsub=\relax
1618   \let\startlock=\relax \let\endlock=\relax
1619   \let\edlabel=\@gobble
1620   \let\setline=\@gobble \let\advanceline=\@gobble
1621   \let\sameword\sameword@inedtext%
1622   \let\edtext=dummy@edtext
1623   \let\edindex\dummy@edindex%
1624   \l@dtabnoexpands
1625   \let\linenumnotation=\@gobble%
1626   \morenoexpands}
1627 \let\morenoexpands=\relax
1628
1629 %

```

`\@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.

```

1630 \newcommand{\@tag}{}
1631 %

```

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

```

1632 \newcount\@edtext@level%
1633 \@edtext@level=0%
1634 %

```

`\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.

```

1635 \newif\if@edtext@secondarg@%
1636 %

```

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

```
1637 \newcounter{edtext}
1638 \renewcommand{\theedtext}{edtxt@arabic{edtext}}%
1639 %
```

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

```
1640 \newcommand{\edtext}[2]{\leavevmode%
1641 %
```

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

```
1642 \ifnumberedpar%
1643 %
```

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

```
1644 \global\advance\@edtext@level by 1%
1645 %
```

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

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`.

```
1646 \stepcounter{edtext}%
1647 \csxdef{thisedtext@the\@edtext@level}{\theedtext}%
1648 %
```

By default, we do not use `\lemma`

```
1649 \global\@lemmacommand@false%
1650 %
```

```
1651 \begingroup%
1652 %
```

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`.

```
1653 \ifledRcol%
1654 \ifcvoid{sw@list@edtextR@the\@edtext@level}%
1655 {\global\let\sw@inthisedtext\empty}%
1656 {\expandafter\gl@p\csname sw@list@edtextR@the\@edtext@level\
endcsname\to\sw@inthisedtext}%
1657 \else%
1658 \ifcvoid{sw@list@edtext@the\@edtext@level}%
1659 {\global\let\sw@inthisedtext\empty}%
1660 {\expandafter\gl@p\csname sw@list@edtext@the\@edtext@level\
endcsname\to\sw@inthisedtext}%
1661 \fi%
1662 %
```

`\@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.

```
1663     \global\renewcommand{\@tag}{-%
1664         \no@expands #1%
1665     }%
1666 %
```

`\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`.

```
1667     \set@line%
1668 %
```

`\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`.

```
1669     \ifledRcol \global\insert@countR \z%
1670     \else      \global\insert@count \z@ \fi%
1671 %
```

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.

```
1672     \@edtext@secondarg@true%
1673     \ignorespaces #2\relax%
1674     \@edtext@secondarg@false%
1675 %
```

With $X_{\text{L}}^{\text{A}}\text{T}_{\text{E}}\text{X}$, 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.

```
1676     \if@RTL%
1677         \flag@start@RTL%
1678     \else%
1679         \flag@start%
1680     \fi%
1681 %
```

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

```

1682     \if@lemmacommand@%
1683     \ifledRcol%
1684         \write\linenum@outR{\string\@lemma}%
1685     \else%
1686         \write\linenum@out{\string\@lemma}%
1687     \fi%
1688     \fi%
1689 %

```

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.

```

1690     \endgroup%
1691     \ifdef{\hypertarget}%
1692     {%
1693     \Hy@raisedlink@left{\hypertarget{\cuse{thisedtext@the\
@edtext@level}:start}{}}%
1694     \showlemma{#1}%
1695     \Hy@raisedlink{\hypertarget{\cuse{thisedtext@the\@edtext@level}:
end}{}}%
1696     }%
1697     {%
1698     \showlemma{#1}%
1699     }%
1700 %

```

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.

```

1701     \ifx@end@lemmas\empty \else%
1702     \gl@p@end@lemmas\to\x@lemma%
1703     \x@lemma%
1704     \global\let\x@lemma=\relax%
1705     \fi%
1706     \if@RTL%
1707     \flag@end@RTL%
1708     \else%
1709     \flag@end%
1710     \fi%
1711 %

```

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`.

```

1712 \global\@noneed@Footnotefalse%
1713 \global\advance\@edtext@level by -1%
1714 \global\@lemmacommand@false%
1715 %

```

We also reset `\@beforeinsertofthisedtext`

```

1716 \global\let\@beforeinsertofthisedtext\relax%
1717 %

```

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

```

1718 \else%
1719 \showlemma{#1} (\textbf{\textsc{Edtext outside numbered paragraph}})\
led@err@edtextoutsidestart%
1720 \fi%
1721 }%
1722
1723
1724 %

```

`\@beforeinsertofthisedtext` `\@beforeinsertofthisedtext` is an internal macro. `reledmac` or `reledpar` can add in this macro any content required to be executed before doing any `\insert` related to a `\edtext`. Its content is `\let` equal to `\relax` at the end of every `\edtext`.

```

1725 \let\@beforeinsertofthisedtext\relax
1726 %

```

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

```

1727 \newif\ifnumberline
1728 \numberlinetrue
1729 %

```

`\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.

```

1730 \newcommand*{\set@line}{%
1731 \ifledRcol
1732 \ifx\line@listR\empty

```

```

1733 \global\noteschanged@true
1734 \xdef\l@d@nums{000|000|000|000|000|000|000|\edfont@info}%
1735 \else
1736 \gl@p\line@listR\to\@tempb
1737 \xdef\l@d@nums{\@tempb|\edfont@info}%
1738 \global\let\@tempb=\undefined
1739 \fi
1740 \ifx\annot@listR\empty%
1741 \xdef\l@current@annotR{|\}%
1742 \else%
1743 \gl@p\annot@listR\to\@tempb%
1744 \xdef\l@current@annotR{\@tempb}%
1745 \global\let\@tempb=\undefined%
1746 \fi%
1747 \else
1748 \ifx\line@list\empty
1749 \global\noteschanged@true
1750 \xdef\l@d@nums{000|000|000|000|000|000|000|\edfont@info}%
1751 \else
1752 \gl@p\line@list\to\@tempb
1753 \xdef\l@d@nums{\@tempb|\edfont@info}%
1754 \global\let\@tempb=\undefined
1755 \fi
1756 \ifx\annot@list\empty%
1757 \xdef\l@current@annot{|\}%
1758 \else%
1759 \gl@p\annot@list\to\@tempb%
1760 \xdef\l@current@annot{\@tempb}%
1761 \global\let\@tempb=\undefined%
1762 \fi%
1763 \fi}
1764
1765 %

```

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

```

1766 \newcommand*\edfont@info{\f@encoding/\f@family/\f@series/\f@shape}
1767
1768 %

```

VI.2 Substitute lemma

`\lemma` The `\lemma{text}` macro allows you to change the lemma that is passed on to the notes. Read about `\@tag` in normal `\edtext` macro for more details about `\sw@list@inedtext` and `\no@expands` (VI.1 p. 135).

```

1769 \newcommand*\lemma}[1]{%
1770 \global\@lemmacommand@true%
1771 \global\renewcommand{\@tag}{#1}%

```

```

1772 \no@expands #1%
1773 }%
1774 \ignorespaces%
1775 }%
1776 %

```

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

```

1777 \newcommand{\@lemma}{%
1778 \booltrue{lemmacommand@the\@edtext@level}%
1779 }%
1780 %

```

\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.

```

1781 \newif\if@lemmacommand@%
1782 %

```

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. 104): the starting page, line, and sub-line numbers, followed by the ending page, line, and sub-line numbers, and then the font specifier for the lemma. However, you can omit any parameters you do not want to change, and you can omit a string of vertical bars at the end of the argument. Hence \linenum{18|4|0|18|7|1|0} is an invocation that changes all the parameters, but \linenum{|3} only changes the starting line number, and leaves the rest unaltered.

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

```

1783 \newcommand*{\linenum}[1]{%
1784 \xdef\@tempa{#1|||||\noexpand\\\l@d@nums}%
1785 \global\let\l@d@nums=\empty
1786 \expandafter\line@set\@tempa|\\\ignorespaces}
1787 %

```

\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.

```

1788 \def\line@set#1|#2|#3|#4\\{%
1789 \gdef\@tempb{#1}%
1790 \ifx\@tempb\empty

```

```

1791     \l@d@add{#3}%
1792 \else
1793     \l@d@add{#1}%
1794 \fi
1795 \gdef\@tempb{#4}%
1796 \ifx\@tempb\empty\else
1797     \l@d@add{|}\line@set#2\|#4\|
1798 \fi}
1799 %

```

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

```

1800 \newcommand{\l@d@add}[1]{\xdef\l@d@nums{\l@d@nums#1}}
1801
1802 %

```

`\lineannot` Last but not least, `\lineannot` allow us to substitute line number annotation. It is different of `\linenum` for ascendant compatibility with older version of `reledmac`. It calls `\lineannot@set` to determine if we must change only one annotation or two, or zero.

`\lineannot@set`

```

1803 \newcommand*{\lineannot}[1]{%
1804     \lineannot@set#1|}%
1805 }%
1806 \def\lineannot@set#1|#2|{%
1807     \expandafter\parse@annot#1|#2|}%
1808 \IfStrEq{#1}{}%
1809     {\let\@tempa\annot@start}%
1810     {\def\@tempa{#1}}%
1811 \IfStrEq{#2}{}%
1812     {\let\@tempb\annot@start}%
1813     {\def\@tempb{#2}}%
1814 \xdef\l@current@annot{\@tempa|\@tempb}%
1815 }%
1816 %

```

VI.4 Lemma disambiguation

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

The general mechanism is the following:

- **At the first run**, each `\sameword` command increments an `etoolbox` counter the name of which contains the argument of the `\sameword` commands.
- Then this counter, associated with the argument of `\sameword` is stored with the `\@sw` command in the auxiliary file of the current `reledmac` section (the `.1`, `.2...` file).

- **When this auxiliary file is read at the second run**, different operations are achieved:

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

```

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

```

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

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

```

bar1 foo1 foo2 bar2 foo3 (A)(B) foo4 bar3 (C) foo5 (D) bar4 (E)

```

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

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

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

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

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

After theory, implementation.

`\get@sw@txt` As the argument of `\sameword` can contain an active character if we use `inputenc` with `utf8` option instead of native UTF-8 engine, we store its detokenized content in a macro in order to allow the dynamic name of macro with `\csname`.³⁰

Because there is a bug with `\detokenize` and $X_{\text{T}}\text{E}_{\text{X}}$ when using non BMP characters³¹, we detokenize only for non- $X_{\text{T}}\text{E}_{\text{X}}$ engines. In any case, in $X_{\text{T}}\text{E}_{\text{X}}\text{a}$ `\csname` construction can contain UTF-8 characters without a problem, as UTF-8 characters are not managed with category codes, but instead read directly as UTF-8 characters.

³⁰See <http://tex.stackexchange.com/q/244538/7712>.

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

```

1817 \newcommand{\get@sw@txt}[1]{%
1818   \begingroup%
1819   \swnoexpands%
1820 }%

```

Using case sensibility option.

```

1821   \ifsw@caseinsensitive%
1822     \def\@tmpa##1{\lowercase{##1}}%
1823   \else%
1824     \def\@tmpa##1{##1}%
1825   \fi%
1826 }%

```

And now, define \sw@txt.

```

1827   \ifxetex%
1828     \@tmpa{\xdef\sw@txt{##1}}%
1829   \else%
1830     \@tmpa{\expandafter\xdef\expandafter\sw@txt\expandafter{\detokenize
1831 {##1}}}%
1832   \fi%
1833 }%
1834 }%

```

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}{%
1836   \let\sameword\@firstofone%Allow to have nested \sameword
1837   \let\emph\@firstofone%
1838   \let\textit\@firstofone%
1839   \let\textbf\@firstofone%
1840   \let\textsc\@firstofone%
1841   \let\framebox\@firstofone%
1842   \RenewExpandableDocumentCommand{\edindex}{om}{-}{%
1843     \ifdefined\index%
1844       \RenewExpandableDocumentCommand{\index}{om}{-}{%
1845         \fi%
1846       \let\selectlanguage\@gobble%
1847       \let\foreignlanguage\@secondoftwo%
1848       \ifdefined\xpg@loaded%
1849         \renewcommand\do[1]{%
1850           \expandafter\RenewExpandableDocumentCommand\expandafter{\csname
1851 text##1\endcsname}{om}{####2}}%
1852         \expandafter\docsvlist\expandafter{\xpg@loaded}%
1853       \fi%
1854     }%
1855 }%

```

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

```
1856 \newcommandx{\sameword}[2][1,usedefault]{%
1857     \leavevmode%
1858     \get@sw@txt{#2}%
1859 %
```

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

```
1860     \unless\ifledRcol%
1861         \csnumgdef{sw@\sw@txt}{\csuse{sw@\sw@txt}+\@ne}%
1862 %
```

Then, write its value to the numbered file.

```
1863     \protected@write\linenum@out{}{\string\@sw{\sw@txt}{\csuse{sw@\sw@txt}
1864 }}{#1}}%
1864 %
```

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

```
1865     \else%
1866         \csnumgdef{sw@\sw@txt}{\csuse{sw@\sw@txt}+\@ne}%
1867         \protected@write\linenum@outR{}{\string\@sw{\sw@txt}{\csuse{sw@\sw@txt}
1868 }}{#1}}%
1868     \fi%
1869 %
```

And print the word.

```
1870     #2%
1871 }%
1872 %
```

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

```
\if@addsw73 \newif\if@addsw%
1874 %
```

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

```
1875 \newcommand{\@sw}[3]{%
1876     \get@sw@txt{#1}%
1877     \unless\ifledRcol%
1878 %
```

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

```
1879     \csxdef{sw@\sw@txt @\the\absline@num @\the\section@num}{#2}%
1880 %
```

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


```

1881 \numdef{\prev@line}{\the\absline@num-1}%
1882 \ifcsundef{sw@sw@txt @\prev@line @\the\section@num}{%
1883   \csnumgdef{sw@sw@txt @\prev@line @\the\section@num}{#2-1}%
1884   }{}%
1885 %

```

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

```

1886 \numdef{\the@sw}{#2-\csuse{sw@sw@txt @\prev@line @\the\section@num}}%
1887 %

```

And do the same thing for the right side.

```

1888 \else%
1889   \csxdef{sw@sw@txt @\the\absline@numR @\the\section@numR @R}{#2}%
1890   \numdef{\prev@line}{\the\absline@numR-1}%
1891   \ifcsundef{sw@sw@txt @\prev@line @\the\section@numR @R}{%
1892     \csnumgdef{sw@sw@txt @\prev@line @\the\section@numR @R}{#2-1}%
1893     }{}%
1894   \numdef{\the@sw}{#2-\csuse{sw@sw@txt @\prev@line @\the\section@numR @R}}%
1895   \fi%
1896 %

```

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

```

1897 \@tempcnta=\@edtext@level
1898 \@whilenum{\@tempcnta>0}\do{%
1899   \ifcsdef{sw@list@edtext@tmp@\the\@tempcnta}%
1900     {%
1901       \@addswfalse%
1902       \notbool{lemmacommand@\the\@tempcnta}%
1903       {\@addswtrue}%
1904       {\IfStrEq{#3}{inlemma}%
1905        {\@addswtrue}%
1906        {%
1907          \def\do##1{%
1908            \ifnumequal{##1}{\the\@tempcnta}%
1909              {\@addswtrue\listbreak}%
1910              }%
1911          }%
1912          \docsvlist{#3}%
1913        }%
1914      }%
1915     \if@addsw%
1916       \letcs{\@tmp}{sw@list@edtext@tmp@\the\@tempcnta}%
1917       \ifledRcol%
1918         \xright@appenditem{\the@sw}{\the\absline@numR}}\to\@tmp%
1919       \else%
1920         \xright@appenditem{\the@sw}{\the\absline@num}}\to\@tmp%
1921       \fi%
1922     \cslet{sw@list@edtext@tmp@\the\@tempcnta}{\@tmp}%

```

```

1923     \fi%
1924     }%
1925     }%
1926     \advance\@tempcnta by -1%
1927 }%
1928 }%
1929 %

```

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

```

1930 \newcommandx{\sameword@inedtext}[2][1,usedefault]{%
1931   \get@sw@txt{#2}%
1932   \unless\ifledRcol@%
1933 %

```

Just a precaution.

```

1934   \ifx\sw@list@inedtext\empty%
1935     \def\the@sw{999}%
1936     \def\this@absline{-99}%
1937   \else%
1938 %

```

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

```

1939     \gl@p\sw@list@inedtext\to\@tmp%
1940     \edef\the@sw{\expandafter\@firstoftwo\@tmp}%
1941     \edef\this@absline{\expandafter\@secondoftwo\@tmp}%
1942     \fi%
1943 %

```

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

```

1944     \ifcsdef{sw@\sw@txt @\this@absline @\the\section@num}{%
1945       \numdef{\prev@line}{\this@absline-1}%
1946       \numdef{\sw@atthisline}{\csuse{sw@\sw@txt @\this@absline @\the\
1947 section@num}-\csuse{sw@\sw@txt @\prev@line @\the\section@num}}%
1948       {\numdef{\sw@atthisline}{0}}%
1949 %

```

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

```

1950     \ifnumgreater{\sw@atthisline}{1}%
1951       {\showwordrank{#2}{\the@sw}}%
1952       {#2}%
1953 %

```

And the same for right side.

```

1954   \else%
1955     \ifx\sw@list@inedtext\empty%

```

```

1956 \def\the@sw{999}%
1957 \def\this@absline{-99}%
1958 \else%
1959 \gl@p\sw@list@inedtext\to\@tmp%
1960 \edef\the@sw{\expandafter\@firstoftwo\@tmp}%
1961 \edef\this@absline{\expandafter\@secondoftwo\@tmp}%
1962 \fi%
1963 \ifcsdef{sw@\sw@txt @\this@absline @\the\section@numR @R}{%
1964 \numdef{\prev@line}{\this@absline-1}%
1965 \numdef{\sw@atthisline}{\csuse{sw@\sw@txt @\this@absline @\the\
section@numR @R}-\csuse{sw@\sw@txt @\prev@line @\the\section@numR @R}}%
1966 }%
1967 {\numdef{\sw@atthisline}{0}}%
1968 \ifnumgreater{\sw@atthisline}{1}%
1969 {\showwordrank{#2}{\the@sw}}%
1970 {#2}%
1971 \fi%
1972 }%
1973 %

```

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

```

1974 \newcommand{\showwordrank}[2]{%
1975 #1\textsuperscript{#2}%
1976 }%
1977 %

```

VII Paragraph decomposition and reassembly

In order to be able to count the lines of text and affix line numbers, we add an extra stage of processing for each paragraph. We send the paragraph into a box register, rather than straight onto the vertical list, and when the paragraph ends we slice the paragraph into its component lines; to each line we add any notes or line numbers, add a command to write to the line-list, and then at last send the line to the vertical list. This section contains all the code for this processing.

VII.1 Boxes, counters, `\pstart` and `\pend`

`\raw@text` Here are numbers and flags that are used internally in the course of the paragraph decomposition.
`\ifnumberedpar@`
`\numberedpar@true` When we first form the paragraph, it goes into a box register, `\raw@text`, instead
`\numberedpar@false` of onto the current vertical list. The `\ifnumberedpar@` flag will be true while a para-
`\num@lines` graph is being processed in that way. `\num@lines` will store the number of lines in the
`\one@line` paragraph when it is complete. When we chop it up into lines, each line in turn goes
`\par@line` into the `\one@line` register, and `\par@line` will be the number of that line within the
paragraph.

```

1978 \newbox\raw@text
1979 \newif\ifnumberedpar@
1980 \newcount\num@lines
1981 \newbox\one@line
1982 \newcount\par@line
1983 %

```

`\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 `\numberstarttrue` `\autopar` command below may be used to insert these commands automatically. `\numberstartfalse` Beware: everything that occurs between `\pstart` and `\pend` is happening within a group; definitions must be global if you want them to survive past the end of the paragraph. `\labelstarttrue` `\labelstartfalse` `\thepstart`

```

1984 \if@every@pstart@star@
1985 \newcommand{\AtStartEveryPstart}[1]{%
1986   \ifstrempy{#1}%
1987   {\gdef\@at@start@every@pstart{}}%
1988   {\gdef\@at@start@every@pstart{#1}}%
1989 }%
1990 \def\@at@start@every@pstart{}%
1991
1992 \newif\if@every@pstart@star@%
1993 \newcommand{\AtEveryPstart}[1]{%
1994   \ifstrempy{#1}%
1995   {\gdef\@at@every@pstart{}}%
1996   {\gdef\@at@every@pstart{\noindent#1}}%
1997   \global\@at@every@pstart@star@false%
1998 }%
1999 \WithSuffix\newcommand\AtEveryPstart*[1]{%
2000   \ifstrempy{#1}%
2001   {\gdef\@at@every@pstart{}}%
2002   {\gdef\@at@every@pstart{#1}}%
2003   \global\@at@every@pstart@star@true%
2004 }%
2005 \def\@at@every@pstart{}%
2006
2007 \newcounter{pstart}
2008 \renewcommand{\thepstart}{\bfseries\@arabic@c@pstart}. }
2009 \newif\ifnumberpstart
2010 \numberstartfalse
2011 \newif\iflabelpstart
2012 \labelstartfalse
2013 \newcommandx*\pstart[2][1,2,usedefault]{%
2014   \normal@pars%
2015   \ifboolexpr{%
2016     test {\ifstrempy{#1}}%
2017     and test {\ifstrempy{#2}}%

```

```

2018 }%
2019   {\at@every@pstart}%
2020   {%
2021   \ifstrempy{#1}{-}{\noindent#1}%
2022   \ifstrempy{#2}{-}{#2}%
2023   }%
2024 \ifautopar%
2025   \autopar%
2026 \fi%
2027 \ifluatex%
2028   \edef\l@luatexttextdir@L{\the\textdir}%
2029 \fi%
2030 \@nobraektrue%
2031 \ifnumbering \else%
2032   \led@err@PstartNotNumbered%
2033   \beginnumbering%
2034 \fi%
2035 \ifnumberedpar@%
2036   \led@err@PstartInPstart%
2037   \pend%
2038 \fi%
2039 \list@clear{\inserts@list}%
2040 \global\let\next@insert=\empty%
2041 \begingroup\normal@pars%
2042 \global\advance \l@dnumpstartsL@ne
2043 \global\setbox\raw@text=\vbox\bgroup%
2044   \if@nobraek%
2045     \if@afterindent\else%
2046       \noindent%
2047       \global\@afterindenttrue%
2048     \fi%
2049   \fi%
2050   \ifautopar\else%
2051   \ifnumberpstart%
2052     \ifinstanza\else%
2053     \ifsidepstartnum\else%
2054       \thepstart%
2055     \fi%
2056   \fi%
2057   \fi%
2058   \fi%
2059 \numberedpar@true%
2060 \iflabelpstart%
2061   \protected@edef\@currentlabel{\p@pstart\thepstart}%
2062 \fi%
2063 \l@dzeropenalties%
2064 \@at@start@every@pstart%
2065 \ignorespaces%because not automatically ignored if an optional argument
is used (classical TeX behavior for space after commands)
2066 }

```

```
2067 %
```

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

```
2068 \newcommand*{\pend}[2][1,2,usedefault]{\ifnumbering \else%
2069   \led@err@PendNotNumbered%
2070   \fi%
2071   \global\l@dskipversenumberfalse%
2072   \ifnumberedpar@ \else%
2073     \led@err@PendNoPstart%
2074     \fi%
2075 %
```

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.

```
2076   \l@dzeropenalties%
2077   \@at@end@every@pend%
2078   \endgraf\global\num@lines=\prevgraf\egroup%
2079   \global\par@line=0%
2080 %
```

We check if lineation is by `pstart`: in this case, we reset the line number, but only in the second line of the `pstart`. We can't reset line number at the beginning of `\pstart`, as `\setline` is parsed at the end of previous `\pend`, and so, we must do it at the end of first line of `pstart`.

```
2081   \csnumdef{pstartline}{0}%
2082   \loop\ifvbox\raw@text%
2083     \csnumdef{pstartline}{\pstartline+\@ne}%
2084     \do@line%
2085     \ifbypstart@%
2086       \ifnumequal{pstartline}{1}{%
2087         \bgroup%
2088         \let\leavevmode\relax%
2089         \setline{1}%
2090         \egroup%
2091         \resetprevline@}{}%
2092       \fi%
2093     \repeat%
2094 %
```

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

```
2095   \flush@notes%
2096   \endgroup%
2097   \ignorespaces%
2098 %
```

Increase pstart counter.

```

2099 \ifnumberpstart%
2100   \global\pstartnumtrue%
2101 \fi%
2102 \addtocounter{pstart}{1}%
2103 \ifcontinuousnumberingwithcolumns%
2104   \addtocounter{pstartL}{1}%
2105   \addtocounter{pstartR}{1}%
2106 \fi%
2107 %

```

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

```

2108 \normal@pars%
2109 \ifboolexpr{%
2110   test {\ifstrempy{#1}}%
2111   and test {\ifstrempy{#2}}%
2112 }%
2113   {\at@every@pend}%
2114   {%
2115   \ifstrempy{#1}{\noindent#1}%
2116   \ifstrempy{#2}{#2}%
2117   }%
2118 %

```

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.

```

2119 \@nobreakfalse%
2120 \ifautopar%
2121   \autopar%
2122 \fi%
2123 }
2124 %

```

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.

```

\AtEveryPend25
\at@every@pend26
\ifat@every@pend@star@27 \newif\ifat@every@pend@star@%
2128 \newcommand{\AtEveryPend}[1]{%
2129   \ifstrempy{#1}%
2130   {\gdef\at@every@pend{}}%
2131   {\gdef\at@every@pend{\noindent#1}}%
2132   \global\at@every@pend@star@false%
2133 }%
2134 \WithSuffix\newcommand\AtEveryPend*[1]{%
2135   \ifstrempy{#1}%

```

```

2136 {\gdef\at@every@pend{}}%
2137 {\gdef\at@every@pend{#1}}%
2138 \global\at@every@pend@star@true%
2139 }%
2140 \xdef\at@every@pend{}%
2141 %
2142 %

```

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

```

2143 \newcommand{\AtEndEveryPend}[1]{%
2144 \ifstrempy{#1}%
2145 {\xdef\@at@end@every@pend{}}%
2146 {\gdef\@at@end@every@pend{#1}}%
2147 }%
2148 \def\@at@end@every@pend{}%
2149 %

```

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

```

2150 \newcommand*\l@dzzeropenalties{%
2151 \brokenpenalty \z@ \clubpenalty \z@
2152 \displaywidowpenalty \z@ \interlinepenalty \z@ \predisplaypenalty \z@
2153 \postdisplaypenalty \z@ \widowpenalty \z@}
2154 %
2155 %

```

\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.

```

2156 \newif\ifautopar
2157 \autoparfalse
2158 \newcommand*\autopar>{
2159   \ifledRcol
2160     \ifnumberingR \else
2161       \led@err@AutoparNotNumbered
2162       \beginnumberingR
2163       \fi
2164     \else
2165       \ifnumbering \else
2166         \led@err@AutoparNotNumbered
2167         \beginnumbering
2168         \fi
2169       \fi
2170   \autopartrue
2171   \everypar{\setbox0=\lastbox
2172     \endgraf \vskip-\parskip
2173     \pstart \noindent \kern\wd0 \ifnumberpstart\ifinstanza\else\thepstart\
2174   fi\fi
2175   \let\par=\pend}%
2176   \ignorespaces}
2177 %

```

`\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.

```

2177 \newcommand*\normal@pars{\everypar{}\let\par\endgraf}
2178
2179 %

```

`\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.

```

2180 \newif\ifautopar@pause
2181 %

```

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.
`\l@dunhbox@line`

```

2182 \newcommand*\l@dunhbox@line}[1]{\unhbox #1}
2183 \newcommand*\do@line}{%

```

```

2184 {\vbadness=10000
2185 \splittopskip=\z@
2186 \do@linehook
2187 \l@emptyd@ta
2188 \global\setbox\one@line=\vsplit\raw@text to\baselineskip}%
2189 \unvbox\one@line \global\setbox\one@line=\lastbox
2190 \getline@num
2191 \IfStrEq{\led@pb@setting}{before}{\led@check@pb\led@check@nopb}{\}
2192 \ifnum\@lock>\@ne
2193 \inserthangingsymboltrue
2194 \else
2195 \inserthangingsymbolfalse
2196 \fi
2197 \check@pb@in@verse
2198 \ifl@dhidenumber%
2199 \global\l@dhidenumberfalse%
2200 \f@x@l@cks%
2201 \else%
2202 \affixline@num%
2203 \fi%
2204 %

```

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

```

2205 \xifinlist{\the\l@dnumpstartsL}{\eled@sections@}%
2206 {\print@eledsection}%
2207 {\print@line}%
2208 \IfStrEq{\led@pb@setting}{after}{\led@check@pb\led@check@nopb}{\}
2209 }%
2210 %

```

VII.2.2 Process for “normal” line

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

```

2211 \def\print@line{
2212 %

```

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

```

2213 \affixpstart@num%
2214 %

```

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

```

2215 \hb@xt@ \linewidth{%
2216 %

```

User hook.

```

2217 \do@insidelinehook%
2218 %

```

Left line number

```
2219 \l@dld@ta%
2220 %
```

Prepare text to be inserted before notes.

```
2221 \if@firstlineofpage%
2222 \set@Xtxtbeforenotes%
2223 \set@txtbeforenotesX%
2224 \global\@firstlineofpagefalse%
2225 \fi%
2226 %
```

Insert footnotes made of manuscripts data and critical footnotes.

```
2227 \ifdefstring{\ms@data@position}{msdata-regular}{%
2228 \insert@msdata%
2229 \add@inserts%
2230 \add@Xgroupbyline%
2231 }{%
2232 \add@inserts%
2233 \add@Xgroupbyline%
2234 \insert@msdata%
2235 }%
2236 %
```

Insert marginal notes.

```
2237 \affixside@note%
2238 %
```

Print left notes.

```
2239 \l@dlsn@te
2240 %
```

Boxes the line, writes information about new line in the numbered file.

```
2241 {\ledllfill\hb@xt@ \wd\one@line{%
2242 \new@line%
2243 %
```

If we use the continuousnumberingwithcolumns, we increase right line number.

```
2244 \ifcontinuousnumberingwithcolumns%
2245 \unless\ifafterendnumberingR%
2246 \new@lineR%
2247 \xappto\next@line@list@stuffR{%
2248 \unexpanded{\global\line@numR=}\the\line@num%
2249 }%
2250 \fi%
2251 \fi%
2252 %
```

If we use Lua^LTeX then restore the direction.

```

2253 \ifluatex%
2254 \textdir\l@luatextextdir@L%
2255 \fi%
2256 %

```

Insert, if needed, the hanging symbol.

```

2257 \inserthangingsymbol%
2258 %

```

And so, print the line.

```

2259 \l@dunhbox@line{\one@line}}%
2260 %

```

Right line number

```

2261 \ledrfill\l@drd@ta%
2262 %

```

Print right notes.

```

2263 \l@drsn@te%
2264 }}%
2265 %

```

And reinsert penalties (for page breaking)...

```

2266 \add@penalties%
2267 }
2268 %

```

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.

```

2269 \def\print@eledsection{%
2270 \if@firstlineofpage%
2271 \set@Xtxtbeforenotes%
2272 \set@txtbeforenotesX%
2273 \global\@firstlineofpagefalse%
2274 \fi%
2275 \ifdefstring{\ms@data@position}{msdata-regular}{%
2276 \insert@msdata%
2277 \add@inserts%
2278 \add@Xgroupbyline%
2279 }{%
2280 \add@inserts%
2281 \add@Xgroupbyline%
2282 \insert@msdata%
2283 }%

```

```

2284 \affixside@note%
2285 \numdef{\temp@}{\l@dnumpstartsL-1}%
2286 \xifinlist{\temp@}{\eled@sections@}{\@nobreaktrue}{\@nobreakfalse}%
2287 \eled@sectioningtrue%
2288 \csuse{eled@sectioning@the\l@dnumpstartsL}%
2289 \eled@sectioningfalse%
2290 \global\csundef{eled@sectioning@the\l@dnumpstartsL}%
2291 \if@RTL%
2292 \hspace{-3\paperwidth}%
2293 {\hbox{\l@dunhbox@line{\one@line}} \new@line}%
2294 \else%
2295 \hspace{3\paperwidth}%
2296 {\new@line \hbox{\l@dunhbox@line{\one@line}}}%
2297 \fi%
2298 \vskip-\baselineskip%
2299 }
2300 %

```

VII.2.4 Hooks

`\do@linehook` `\do@insidelinehook` Two hooks into `\do@line`. The first is called at the beginning of `\do@line`, the second is called in the line box. The second can, for example, have a `\markboth` command inside, the first can not.

```

2301 \newcommand*{\do@linehook}{}
2302 \newcommand*{\do@insidelinehook}{}
2303 %

```

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

```

2304 \newcommand*{\dolinehook}[1]{\gdef\do@linehook{#1}}%
2305 \newcommand*{\doinsidelinehook}[1]{\gdef\do@insidelinehook{#1}}%
2306
2307 %

```

VII.2.5 Sidenotes and marginal line number initialization

`\l@emptyd@ta` `\l@dld@ta` `\l@drd@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
\l@dcsnotetext@l
\l@dcsnotetext@r
2308 \newcommand*{\l@emptyd@ta}{%
2309 \gdef\l@dld@ta{}%
2310 \gdef\l@drd@ta{}%
2311 \gdef\l@dcsnotetext@l{}%
2312 \gdef\l@dcsnotetext@r{}%
2313 \gdef\l@dcsnotetext{}%
2314
2315 %

```

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

`\l@drsn@te`

```

2316 \newcommand{\l@dlsn@te}{%
2317   \ifboolexpr{%
2318     bool {l@dprintingcolumns}%
2319     and bool {ledRcol@}%
2320   }{% If we are on a right column
2321     \@tempdima=\@morespace@leftnote@rightcolumn%
2322   }{%
2323     \@tempdima=\z@%
2324   }%
2325   \hb@xt@ \z@{\hss\box\l@dlp@rbox\kern\ledlnotesep\hskip\@tempdima}%
2326 }%
2327 \newcommand{\l@drsn@te}{%
2328   \ifboolexpr{%
2329     bool {l@dprintingcolumns}%
2330     and not bool {ledRcol@}%
2331   }{% If we are on a left column
2332     \@tempdima=\@morespace@rightnote@leftcolumn%
2333   }{%
2334     \@tempdima=\z@%
2335   }%
2336   \hb@xt@ \z@{\hskip\@tempdima\kern\ledrsnotesep\box\l@drp@rbox\hss}%
2337 }%
2338 %
2339 %

```

`\ledllfill` These macros are called at the left (`\ledllfill`) and the right (`\ledrlfill`) of each numbered line. The initial definitions correspond to the original code for `\do@line`.

`\ledrlfill`

```

2340 \newcommand*\ledllfill{\hfil}
2341 \newcommand*\ledrlfill{}
2342 %
2343 %

```

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.

```

2344 \newcommand*\getline@num{%
2345   \global\advance\absline@num \@ne%
2346   \do@actions
2347   \do@ballast
2348   \ifnumberline
2349     \ifsublines@
2350       \ifnum\sub@lock<\tw@
2351         \global\advance\subline@num \@ne

```

```

2352     \fi
2353     \else
2354         \ifnum\@lock<\tw@
2355             \global\advance\line@num \@ne
2356             \global\subline@num \z@
2357         \fi
2358     \fi
2359 \fi
2360 }
2361 %

```

`\do@ballast` The real work in the macro above is done in `\do@actions`, but before we plunge into that, let's get `\do@ballast` out of the way. This macro looks to see if there is an action to be performed on the *next* line, and if it is going to be a page break action, `\do@ballast` decreases the count `\ballast@count` counter by the amount of `ballast`. This means, in practice, that when `\add@penalties` assigns penalties at this point, \TeX will be given extra encouragement to break the page here (see XII.2 p. 172).

`\ballast@count` First we set up the required counters; they are initially set to zero, and will remain so unless you type `\setcounter{ballast}{(some figure)}` in your document.

```

2362 \newcount\ballast@count
2363 \newcounter{ballast}
2364 \setcounter{ballast}{0}
2365 %

```

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.

```

2366 \newcommand*{\do@ballast}{\global\ballast@count \z@
2367 \begingroup
2368 \advance\absline@num \@ne
2369 \ifnum\next@actionline=\absline@num
2370 \ifnum\next@action>-1001\relax
2371 \global\advance\ballast@count by -\c@ballast
2372 \fi
2373 \fi
2374 \endgroup}
2375 %

```

`\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`.

```

2376 \newcommand*{\do@actions}{%
2377 \global\let\do@actions@next=\relax

```

```

2378 \ifnum\absline@num<\next@actionline\else
2379 %

```

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.

```

2380 \ifnum\next@action>-1001
2381 \global\page@num=\next@action
2382 \ifresumenumbering@start%
2383 \setbox0=\hbox{}%Required to get the correct page number, when the
\resumenumbering is just after a \newpage
2384 \ifnum\pausenumbering@page@num<\page@num%
2385 \global\resumenumbering@startfalse%
2386 \fi%
2387 \fi%
2388 \ifboolexpr{%
2389 bool{resumenumbering@start}%
2390 and test {\ifdimgreater{\pagedepth}{\z@}}%
2391 }%
2392 {}%
2393 {\global\@firstlineofpagetrue}%
2394 \ifcsdef{reset@line\the\absline@num @\the\section@num}%
2395 {%
2396 \global\line@num=\z@ \global\subline@num=\z@%
2397 \resetprevline@%
2398 }%
2399 {}%
2400 \global\resumenumbering@startfalse%
2401 \add@msdata@firstlineofpage%
2402 %

```

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`.)

```

2403 \else
2404 \ifnum\next@action<-4999
2405 \@l@dtmpcnta=-\next@action
2406 \advance\@l@dtmpcnta by -5001
2407 \ifsublines@
2408 \global\subline@num=\@l@dtmpcnta
2409 \else
2410 \global\line@num=\@l@dtmpcnta
2411 \fi
2412 %

```

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

```

2413 \else
2414 \@l@dtmpcnta=-\next@action
2415 \advance\@l@dtmpcnta by -1000
2416 \do@actions@fixedcode

```



```

2417 \fi
2418 \fi
2419 %

```

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.

```

2420 \ifx\actionlines@list\empty
2421 \gdef\next@actionline{1000000}%
2422 \else
2423 \gl@p\actionlines@list\to\next@actionline
2424 \gl@p\actions@list\to\next@action
2425 \global\let\do@actions@next=\do@actions
2426 \fi
2427 \fi
2428 %

```

Make the recursive call, if necessary.

```

2429 \do@actions@next}
2430
2431 %

```

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

```

2432 \newcommand*{\do@actions@fixedcode}{%
2433 \ifcase\@l@dttempcnta
2434 \or% % 1001 = starting sublineation
2435 \global\sublines@true
2436 \or% % 1002 = ending sublineation
2437 \global\sublines@false
2438 \or% % 1003 = starting locking number
2439 \global\@lock=\@ne
2440 \or% % 1004 = ending locking number
2441 \ifnum\@lock=\tw@
2442 \global\@lock=\thr@@
2443 \else
2444 \global\@lock=\z@
2445 \fi
2446 \or% % 1005 = starting locking subnumber
2447 \global\sub@lock=\@ne
2448 \or% % 1006 = ending locking subnumber
2449 \ifnum\sub@lock=\tw@
2450 \global\sub@lock=\thr@@
2451 \else
2452 \global\sub@lock=\z@
2453 \fi
2454 \or% % 1007 = skipping numbering
2455 \l@dskipnumbertrue

```

```

2456 \or% % 1008 = skipping numbering in stanza
2457 \l@dskipversenumbertrue%
2458 \or% % 1009 = hiding number
2459 \l@dhidnumbertrue
2460 \or% % 1010 = inserting msdata
2461 \add@msdata%
2462 \else
2463 \led@warn@BadAction
2464 \fi}
2465
2466
2467 %

```

IX Line number annotation

The `\linenumannotation` allows users to add manual annotation in to line number, for example to let know the line numbers in an older edition.

That implies the 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 to each `\edtext`, when reading the auxiliary numbered files.
- To crossref commands which refers to line numbers (to be completed when I will have solved this issue).

`\linenumannotation` First, the user levels command, which only writes an command in the numbered files, storing the annotation.

```

2468 \newcommand{\linenumannotation}[1]{%
2469 \leavevmode%In case it start a \pstart, ensure the \@annot is written
after \@nl
2470 \ifledRcol%
2471 \write\linenum@outR{\string\@annot[#1]}%
2472 \else%
2473 \write\linenum@out{\string\@annot[#1]}%
2474 \fi%
2475 }%
2476 %

```

`\linenumannotation@side` The `\linenumannotation@side` macro to determine the position of line number annotation in the line printed in the side, relative to the line number position. The default value is after, but it could be change to before, using the `\Xlinenumannotationposition` macro.

```
2477 \def\linenumannotation@side{after}%
2478 %
```

`\wraplinenumannotation@ref` The `\Xwraplinenumannotation@ref` macro is applied to crossref made to line annotation. The `\Xwraplinenumannotation@side` macro is applied to line annotation in side. They must be modified by users using `\Xwraplinenumannotation`. By default, they call `\textsuperscript`.

```
2479 \def\Xwraplinenumannotation@ref{\textsuperscript}%
2480 \def\Xwraplinenumannotation@side{\textsuperscript}%
2481 %
```

`\@annot` Then, the numbered files command `\@annot`, which:

`\store@annot@to@absline`

- Associate the annotation to the absolute lines number.
- Store the current annotation in a macro, which will be used by the `\@ref@reg` macro, which manages all thing related to `\edtext` in numbered files. As we don't want to store it multiple time if we have nested `\edtext`, we use a auxiliary macro, `\store@annot@to@absline`, which is let to `\@gobble` when we parse nested `\edtext` in numbered auxiliary files.

```
2482 \newcommand{\@annot}[1]{%
2483   \store@annot@to@absline{#1}%
2484   \def\current@annot{#1}%
2485 }%
2486 \newcommand{\store@annot@to@absline}[1]{%
2487   \csgappto{annot@the\absline@num @the\section@num}{#1}%
2488 }%
2489 %
```

`\current@annot` By default, there is not annotation to a line number, so we store a empty `\current@annot`

```
2490 \let\current@annot=\empty%
2491 %
```

`\reset@current@annot` `\reset@current@annot` is called at each `\@n1` in numbered auxiliary files. It reset the annotation of line number at each line.

```
2492 \newcommand{\reset@current@annot}{%
2493   \let\current@annot\empty%
2494 }%
2495 %
```

`\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` read the format and define two global macros.

```

2496 \def\parse@annot#1|#2|{%
2497   \gdef\annot@start{#1}%
2498   \gdef\annot@end{#2}%
2499 }%
2500 %

```

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` if required.

To determine whether we need to affix a line number to this line, we compute the following:

$$n = \text{int}((\text{linenum} - \text{firstlinenum}) / \text{linenumincrement})$$

$$m = \text{firstlinenum} + (n \times \text{linenumincrement})$$

(where *int* truncates a real number to an integer). *m* will be equal to *linenum* only if we are to paste a number on here. However, the formula breaks down for the first line to number (and any before that), so we check that case separately: if `\line@num ≤ \firstlinenum`, we compare the two directly instead of making these calculations.

We compute, in the scratch counter `\@l@tempcnta`, the number of the next line that should be printed with a number (*m* in the above discussion), and move the current line number into the counter `\@l@tempcntb` for comparison.

First, the case when we are within a sub-line range.

```

2501 \newcommand*{\affixline@num}{%
2502 %

```

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).

```

2503   \ifledgroupnotesL@else
2504     \ifnumberline
2505       \ifl@dskipnumber
2506         \global\l@dskipnumberfalse
2507     \else
2508       \ifsublines@
2509         \@l@tempcntb=\subline@num
2510         \ifnum\subline@num>\c@firstsublinenum
2511           \@l@tempcnta=\subline@num
2512           \advance\@l@tempcnta by-\c@firstsublinenum
2513           \divide\@l@tempcnta by\c@sublinenumincrement
2514           \multiply\@l@tempcnta by\c@sublinenumincrement
2515           \advance\@l@tempcnta by\c@firstsublinenum
2516         \else
2517           \@l@tempcnta=\c@firstsublinenum
2518       \fi
2519 %

```

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.

```
2520     \ch@cksub@l@ck
2521 %
```

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

```
2522     \else
2523     \@l@tempcntb=\line@num
2524 %
```

Check on the `\linenumberlist` If it is `\empty` use the standard algorithm.

```
2525     \ifx\linenumberlist\empty
2526     \ifnum\line@num>\c@firstlinenum
2527     \@l@tempcnta=\line@num
2528     \advance\@l@tempcnta by-\c@firstlinenum
2529     \divide\@l@tempcnta by\c@linenumincrement
2530     \multiply\@l@tempcnta by\c@linenumincrement
2531     \advance\@l@tempcnta by\c@firstlinenum
2532     \else
2533     \@l@tempcnta=\c@firstlinenum
2534     \fi
2535     \else
2536 %
```

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

```
2537     \@l@tempcnta=\line@num
2538     \edef\rem@inder{\,\linenumberlist,\number\line@num,}%
2539     \edef\sc@n@list{\def\noexpand\sc@n@list
2540     ###1,\number\@l@tempcnta,###2|\def\noexpand\rem@inder
2541     {####2}}}%
2542     \sc@n@list\expandafter\sc@n@list\rem@inder|
2543     \ifx\rem@inder\empty%
2544     \advance\@l@tempcnta\@ne
2545     \fi
2546 %
```

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

```
2547     \ch@ck@l@ck
2548     \fi
2549 %
```

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

```
2550     \ifnum\@l@tempcnta=\@l@tempcntb
2551     \ifl@dskipversenumber\else
2552 %
```

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 \TeX we have to consider two column documents as well. In this case Peter Wilson thought we need to put the numbers at the outside of the column — the left of the first column and the right of the second. Do the `twocolumn` stuff before going on with the original code.

`\l@dld@ta` A left line number is stored in `\l@dld@ta` and a right one in `\l@drd@ta`.

`\l@drd@ta`

```

2553     \if@twocolumn
2554         \if@firstcolumn
2555             \gdef\l@dld@ta{\llap{\leftlinenum}}%
2556         \else
2557             \gdef\l@drd@ta{\rlap{\rightlinenum}}%
2558         \fi
2559     \else
2560         \ifboolexpr{bool {l@dprintingcolumns} and test {\
ifnumgreater{\line@margin@columns}{\m@ne}}%
2561             {\@l@tempcntb=\line@margin@columns}%
2562             {\@l@tempcntb=\line@margin}%
2563             \ifnum\@l@tempcntb>\@ne
2564                 \advance\@l@tempcntb \page@num
2565             \fi
2566         \ifboolexpr{%
2567             bool {l@dprintingcolumns}%
2568             and (%
2569                 (test {\ifdefstring{\linenum@OnlyPages@ForColumns}{left
}}%
2570
2571                 and test {\ifnumodd{\page@num}}%
2572                 )%
2573                 or%
2574                 (test {\ifdefstring{\linenum@OnlyPages@ForColumns}{right
}}%
2575
2576                 and not test {\ifnumodd{\page@num}}%
2577                 )%
2578                 )%
2579             }%
2580             {%
2581                 \ifodd\@l@tempcntb%
2582                     \gdef\l@drd@ta{\rlap{\rightlinenum}}%
2583                 \else%
2584                     \gdef\l@dld@ta{\llap{\leftlinenum}}%
2585                 \fi%
2586             }%
2587         \fi

```

```

2587     \fi
2588     \fi
2589 %

```

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

```

2590     \f@x@l@cks
2591     \fi
2592     \fi
2593     \fi
2594 }
2595
2596 %

```

`\ch@cksub@l@ck` These macros handle line number locking for `\affixline@num`. `\ch@cksub@l@ck`
`\ch@ck@l@ck` checks subline locking. If it fails, then we disable the line-number display by setting the
`\f@x@l@cks` counters to arbitrary but unequal values.

```

2597 \newcommand*{\ch@cksub@l@ck}{%
2598   \ifcase\sub@lock
2599     \or
2600       \ifnum\sublock@disp=\@ne
2601         \@l@tempcntb=\z@ \@l@tempcnta=\@ne
2602       \fi
2603     \or
2604       \ifnum\sublock@disp=\tw@ \else
2605         \@l@tempcntb=\z@ \@l@tempcnta=\@ne
2606       \fi
2607     \or
2608       \ifnum\sublock@disp=\z@
2609         \@l@tempcntb=\z@ \@l@tempcnta=\@ne
2610       \fi
2611   \fi}
2612 %

```

Similarly for line numbers.

```

2613 \newcommand*{\ch@ck@l@ck}{%
2614   \ifcase\@lock
2615     \or
2616       \ifnum\lock@disp=\@ne
2617         \@l@tempcntb=\z@ \@l@tempcnta=\@ne
2618       \fi
2619     \or
2620       \ifnum\lock@disp=\tw@ \else
2621         \@l@tempcntb=\z@ \@l@tempcnta=\@ne
2622       \fi
2623     \or
2624       \ifnum\lock@disp=\z@
2625         \@l@tempcntb=\z@ \@l@tempcnta=\@ne

```

```

2626     \fi
2627     \fi}
2628 %

```

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

```

2629 \newcommand*{\f@x@l@cks}{%
2630   \ifcase\@lock
2631   \or
2632     \global\@lock=\tw@
2633   \or \or
2634     \global\@lock=\z@
2635   \fi
2636   \ifcase\sub@lock
2637   \or
2638     \global\sub@lock=\tw@
2639   \or \or
2640     \global\sub@lock=\z@
2641   \fi}
2642 %
2643 %

```

XI Pstart number printing in side

In side, the printing of pstart number is running like the printing of line number. There is only some differences:

`\affixpstart@num`
`\pstartnum`

- The pstarts counter is upgrade 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.

```

\leftpstartnum44
\rightpstartnum45 \newif\ifsidepstartnum
\ifsidepstartnum46 \newcommand*{\affixpstart@num}{%
2647   \ifsidepstartnum
2648     \if@twocolumn
2649       \if@firstcolumn
2650         \gdef\l@dld@ta{\llap{\leftpstartnum}}}%
2651       \else
2652         \gdef\l@drd@ta{\rlap{\rightpstartnum}}}%
2653     \fi
2654   \else
2655     \@l@dttempcntb=\line@margin%

```



```

2656     \ifnum\@l@tempcntb>\@ne
2657         \advance\@l@tempcntb \page@num
2658     \fi
2659     \ifodd\@l@tempcntb
2660         \gdef\l@drd@ta{\rlap{\rightpstartnum}}}%
2661     \else
2662         \gdef\l@dld@ta{\llap{\leftpstartnum}}}%
2663     \fi
2664 \fi
2665 \fi
2666 }
2667 %
2668 %
2669
2670 \newif\ifpstartnum
2671 \pstartnumtrue
2672 \newcommand*\leftpstartnum{%
2673     \ifpstartnum\thepstart
2674     \kern\linenumsep\fi
2675     \global\pstartnumfalse
2676 }
2677 \newcommand*\rightpstartnum{%
2678     \ifpstartnum
2679     \kern\linenumsep
2680     \thepstart
2681     \fi
2682     \global\pstartnumfalse
2683 }
2684 %

```

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.

```

2685 \list@create{\inserts@list}
2686 %

```

`\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`.

```
2687 \newcommand*{\add@inserts}{%
2688   \global\let\add@inserts@next=\relax
2689   %
```

If `\inserts@list` is empty, there are not any more notes or insertions for this paragraph, and we need not waste our time.

```
2690   \ifx\inserts@list\empty \else
2691   %
```

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.

```
2692   \ifx\next@insert\empty
2693     \ifx\insertlines@list\empty
2694       \global\noteschanged@true
2695       \gdef\next@insert{100000}%
2696     \else
2697       \gl@p\insertlines@list\to\next@insert
2698     \fi
2699   \fi
2700   %
```

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.

```
2701   \ifnum\next@insert=\absline@num
2702     \gl@p\inserts@list\to\@insert
2703     \@insert
2704     \global\let\@insert=\undefined
2705     \global\let\next@insert=\empty
2706     \global\let\add@inserts@next=\add@inserts
2707   \fi
2708 \fi
2709 %
```

Make the recursive call, if necessary.

```
2710 \add@inserts@next}
2711
2712 %
```

`\add@Xgroupbyline` If you use `\Xgroupbyline`, the insertion of the critical footnotes are not made immediately in `\add@inserts`, but the content to be inserted is stored, to be inserted in one block. This insertion in one block is made by `\add@Xgroupbyline`.

```

2713 \newcommand{\add@Xgroupbyline}{%
2714   \unless\ifnocritical@%
2715   \def\do##1{%Looping on the series
2716     \def\do###1{%Looping on the ##1@forinserting command
2717       \ifcsdef{##1@forinserting@###1}{%
2718         \X@beforeinsertion{##1}%
2719         \if@ledgroup%
2720           \global\setbox\@nameuse{mp##1footins}=\vbox%
2721         \else%
2722           \insert\csname ##1footins\endcsname%
2723         \fi%
2724         {%
2725           \ifcsdef{Xhsize\csuse{series@display##1}@##1}%
2726             {\hsize \csuse{Xhsize\csuse{series@display##1}@##1}}%
2727             {%
2728               \if@ledgroup%
2729                 \unvbox\@nameuse{mp##1footins}%
2730               \fi%
2731               \X@atbegininsertion{##1}%
2732               \ifcsstring{series@display##1}%
2733                 {%
2734                   \Xledsetnormalparstuff{##1}%
2735                   \rule\z@\splittopskip%
2736                 }%
2737                 {%
2738                   \csuse{##1@forinserting@###1}%
2739                   \strut\par%
2740                 }%
2741               \global\csundef{##1@forinserting@###1}%
2742             }%
2743           }%
2744         }%
2745       \ifcsdef{##1@forinserting}{%
2746         \dolistcslloop{##1@forinserting}%
2747       }{%
2748         \global\csundef{##1@forinserting}%
2749       }%
2750     \dolistloop{\@series}%
2751   \fi%
2752 }%
2753
2754
2755 %

```

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. 159). Finally, the penalty is checked to see that it does not go below -10000 .

```

2756 \newcommand*{\add@penalties}{\@l@tempcnta=\ballast@count
2757   \ifnum\num@lines>\@ne
2758     \global\advance\par@line \@ne
2759     \ifnum\par@line=\@ne
2760       \advance\@l@tempcnta \clubpenalty
2761     \fi
2762     \@l@tempcntb=\par@line \advance\@l@tempcntb \@ne
2763     \ifnum\@l@tempcntb=\num@lines
2764       \advance\@l@tempcnta \widowpenalty
2765     \fi
2766     \ifnum\par@line<\num@lines
2767       \advance\@l@tempcnta \interlinepenalty
2768     \fi
2769   \fi
2770   \ifnum\@l@tempcnta=\z@
2771     \relax
2772   \else
2773     \ifnum\@l@tempcnta>-10000
2774       \penalty\@l@tempcnta
2775     \else
2776       \penalty -10000
2777     \fi
2778   \fi}
2779
2780 %

```

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

```

2781 \newcommand*{\flush@notes}{%
2782   \iftoggle{notfirstrun@}\jobname.\extensionchars\the\section@num}{%

```

```

2783 \xloop%
2784 \ifx\inserts@list\empty \else%
2785 \glOp\inserts@list\to\@insert%
2786 \@insert%
2787 \global\let\@insert=\undefined%
2788 \repeat%
2789 }{}%
2790 }%
2791
2792
2793 %

```

\xloop `\xloop` is a variant of the PLAIN \TeX `\loop` macro, useful when it's hard to construct a positive test using the \TeX `\if` commands—as in `\flush@notes` above. One types `\xloop ... \if ... \else ... \repeat`, and the action following `\else` is repeated as long as the `\if` test fails. (This macro will work wherever the PLAIN \TeX `\loop` is used, too, so we could just call it `\loop`; but it seems preferable not to change the definitions of any of the standard macros.)

This variant of `\loop` was introduced by Alois Kabelschacht in *TUGboat* 8 (1987), pp. 184–5.

```

2794 \def\xloop#1\repeat{%
2795 \def\body{#1\expandafter\body\fi}%
2796 \body}
2797
2798 %

```

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.

```

2799 \newcommand{\set@Xtxtbeforenotes}{%
2800 \unless\ifnocritical%
2801 \def\do##1{%
2802 \nottoggle{Xtxtbeforenotesonlyonce@##1}{%
2803 \global\togglefalse{Xtxtbeforenotes@##1@typeset}%
2804 }{}%
2805 }%
2806 \dolistloop{\@series}%
2807 \fi%
2808 }%
2809 %

```

\set@txtbeforenotesX The `\set@txtbeforenotesX` does the same for the `\textbeforenotesX`.

```

2810 \newcommand{\set@txtbeforenotesX}{%
2811 \unless\ifnofamiliar%

```

```

2812 \def\do##1{%
2813   \nottoggle{txtbeforenotesonlyonceX@##1}{%
2814     \global\togglefalse{txtbeforenotesX@##1@typeset}%
2815   }{%
2816   }%
2817   \dolistloop{\@series}%
2818   \fi%
2819 }%
2820 %

```

`\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.

```

2821 \newcommand{\insert@Xtxtbeforenotes}[1]{%
2822   \nottoggle{Xtxtbeforenotes@#1@typeset}{%
2823     \global\toggletrue{Xtxtbeforenotes@#1@typeset}%
2824   \ifcsvoid{Xtxtbeforenotes@#1}{%
2825     \ifcsstring{series@display#1}{paragraph}%
2826     {\noindent\csuse{Xtxtbeforenotes@#1}}%
2827     {\expandafter\insert\csname#1footins\endcsname%
2828       \bgroup%
2829         \noindent%
2830         \ifcsdef{\csuse{series@display#1}@begin@insert}{%
2831           \csuse{\csuse{series@display#1}@begin@insert}{#1}%
2832         }{%
2833           \strut\csuse{Xnotefontsize@#1}\csuse{Xtxtbeforenotes@#1}%
2834         \egroup%
2835       }%
2836     }%
2837   }%
2838 }%
2839 }%
2840
2841
2842 \newcommand{\insert@txtbeforenotesX}[1]{%
2843   \nottoggle{txtbeforenotesX@#1@typeset}{%
2844     \global\toggletrue{txtbeforenotesX@#1@typeset}%
2845   \ifcsvoid{txtbeforenotesX@#1}{%
2846     \ifcsstring{series@displayX#1}{paragraph}%
2847     {\noindent\csuse{txtbeforenotesX@#1}}%
2848     {\expandafter\insert\csname footins#1\endcsname%
2849       \bgroup%
2850         \noindent%
2851         \ifcsdef{\csuse{series@displayX#1}@begin@insert}{%
2852           \csuse{\csuse{series@displayX#1}@begin@insert}{#1}%
2853         }{%
2854           \strut\csuse{notefontsizeX@#1}\csuse{txtbeforenotesX@#1}%

```

```

2855     \egroup%
2856     }%
2857   }%
2858 }%
2859 {}%
2860 }%
2861
2862
2863 %

```

XIII Critical footnotes

The footnote macros are adapted from those in PLAIN T_EX, but they differ in these respects: the outer-level commands must add other commands to a list macro rather than doing insertions immediately; there are many separate levels of the footnotes, not just one; and there are options to reformat footnotes into paragraphs or into multiple columns.

XIII.1 Fonts

Before getting into the details of formatting the notes, we set up some font macros. It is the notes that present the greatest challenge for our font-handling mechanism, because we need to be able to take fragments of our main text and print them in different forms: it is common to reduce the size, for example, without otherwise changing the fonts used.

`\select@lemmafont` `\select@lemmafont` is provided to set the right font for the lemma in a note. This macro extracts the font specifier from the line and page number cluster, and issues the associated font-changing command, so that the lemma is printed in its original font.

```

2864 \def\select@lemmafont#1|#2|#3|#4|#5|#6|#7|{\select@lemmafont#7|}
2865 \def\select@@lemmafont#1/#2/#3/#4|{%
2866   {\fontencoding{#1}\fontfamily{#2}\fontseries{#3}\fontshape{#4}%
2867   \selectfont}
2868
2869 %

```

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.

```

2870 \newcommand*{\footnoteoptions@}[3]{%
2871   \def\do##1{%
2872     \ifstrequal{#1}{L}{% In Leftside
2873       \xright@appenditem{\noexpand\setkeys[mac]{#3footnoteoption}{\
unexpanded{##1}}}\to\inserts@list%
2874       \global\advance\insert@count \@one% Increment the left insert
counter.

```

```

2875     }%
2876     {%
2877         \xright@appenditem{\noexpand\setkeys[mac]{#3footnoteoption}{\
unexpanded{##1}}}\to\inserts@listR%
2878         \global\advance\insert@countR \@ne% Increment the right insert
counter insert.
2879     }%
2880 }%
2881 \notblank{#2}{\docsvlist{#2}}{}}% Parsing all options
2882 }
2883 %

```

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.

```

2884 \newcommand*{\footnotelang@lua}[1][1=L,usedefault]{%
2885     \ifstrequal{#1}{L}{%
2886         \xright@appenditem{\csxdef{footnote@luatextdir}{\the\textdir}}\to\
inserts@list%Know the dir of lemma
2887         \global\advance\insert@count \@ne%
2888         \xright@appenditem{\csxdef{footnote@luatexpardir}{\the\pardir}}\to\
inserts@list%Know the dir of lemma
2889         \global\advance\insert@count \@ne%
2890     }%
2891     {%
2892         \xright@appenditem{\csxdef{footnote@luatextdir}{\the\textdir}}\to\
inserts@listR%Know the dir of lemma
2893         \global\advance\insert@countR \@ne%
2894         \xright@appenditem{\csxdef{footnote@luatexpardir}{\the\pardir}}\to\
inserts@listR%Know the dir of lemma
2895         \global\advance\insert@countR \@ne%
2896     }%
2897 }
2898 %

```

`\footnotelang@poly` `\footnotelang@poly` is called to remember the information about the language of a lemma when polyglossia is used.

```

2899 \newcommand*{\footnotelang@poly}[1][1=L,usedefault]{%
2900     \ifstrequal{#1}{L}{%
2901         \if@RTL%
2902             \xright@appenditem{\csxdef{footnote@dir}{@RTLtrue}}\to\
inserts@list%Know the language used in the lemma
2903             \global\advance\insert@count \@ne%
2904         \else
2905             \xright@appenditem{\csxdef{footnote@dir}{@RTLfalse}}\to\
inserts@list%Know the language of lemma

```



```

2906     \global\advance\insert@count \@ne%
2907     \fi%
2908     \xright@appenditem{{\csxdef{footnote@lang}{\expandonce\languagename}}}\
to\inserts@list%Know the language of lemma
2909     \global\advance\insert@count \@ne%
2910     }%
2911     {%
2912     \if@RTL
2913     \xright@appenditem{{\csxdef{footnote@dir}{@RTLtrue}}}\to\
inserts@listR%Know the language of lemma
2914     \global\advance\insert@countR \@ne%
2915     \else
2916     \xright@appenditem{{\csxdef{footnote@dir}{@RTLfalse}}}\to\
inserts@listR%Know the language of lemma
2917     \global\advance\insert@countR \@ne%
2918     \fi
2919     \xright@appenditem{{\csxdef{footnote@lang}{\expandonce\languagename}}}\
to\inserts@listR%Know the language of lemma
2920     \global\advance\insert@countR \@ne%
2921     }%
2922 }
2923 %

```

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 of the footnotes arrangement: “normal”, “paragraphed”, “two columns” or “three columns”.

XIII.5 General setup

`\footsplitskips` Some setup code that is common for a variety of the footnotes. The setup is for:

- `\interlinepenalty`.
- `\splittopskip` (skip before last part of notes that flow from one page to another).

- `\splitmaxdepth`.
- `\floatingpenalty`, that is penalty values being added when a long note flows from one page to another. Here, we let it to 0 when we are processing parallel pages in `eledpar`, in order to allow notes to flow from left to right pages and *vice-versa*. Otherwise, we let it to `\@MM`, which is the standard \TeX `\floatingpenalty`.

```

2924 \newcommand*\footsplitskips}{%
2925   \interlinepenalty=\interfootnotelinepenalty
2926   \unless\ifl@dprintingpages%
2927     \floatingpenalty=\@MM%
2928   \fi%
2929   \splittopskip=\ht\strutbox \splitmaxdepth=\dp\strutbox
2930   \leftskip=\z@skip \rightskip=\z@skip}
2931
2932 %

```

`\normalfootnoterule` `\normalfootnoterule` is a standard footnote-rule macro, for use by a `footstart` macro: just the same as the PLAIN \TeX footnote rule.

```

2933 \let\normalfootnoterule=\footnoterule
2934 %

```

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.

```

2935 \newcommandx{\Xarrangement}[2][1,usedefault]{%
2936   \def\do##1{%
2937     \csname Xarrangement@#2\endcsname{##1}%
2938   }%
2939   \ifstrempy{#1}%
2940     {%
2941       \dolistloop{\@series}%
2942     }%
2943   {
2944     \docsvlist{#1}%
2945   }%
2946 }%
2947 %

```

XIII.6.2 Normal footnote

`\Xarrangement@normal` We can now define all the parameters for the series of footnotes; initially they use the “normal” footnote formatting.

What we want to do here is to insert something like the following for each footnote series. (This is an example, not part of the actual `reledmac` code.)

```
\skip\Afootins=12pt plus5pt minus5pt
\count\Afootins=1000
\dimen\Afootins=0.8\vsiz
\let\vAfootnote=\normalvfootnote \let\Afootfmt=\normalfootfmt
\let\Afootstart=\normalfootstart \let\Afootgroup=\normalfootgroup
\let\Afootnoterule=\normalfootnoterule
```

(Read *The TeXbook* in order to understand what are the counter, skip and dimen associated to an insertion.)

Instead of repeating ourselves, we define a `\Xarrangement@normal` macro that makes all these assignments for us, for any given series letter. This command is called when people use `\Xarrangement[⟨series⟩]{normal}`

Now we set up the `\Xarrangement@normal` macro itself. It takes one argument: the footnote series letter.

```
2948 \newcommand*{\Xarrangement@normal}[1]{%
2949 \csgdef{series@display#1}{normal}
2950 \expandafter\let\csname #1footstart\endcsname=\normalfootstart
2951 \expandafter\let\csname v#1footnote\endcsname=\normalvfootnote
2952 \expandafter\let\csname #1footfmt\endcsname=\normalfootfmt
2953 \expandafter\let\csname #1footgroup\endcsname=\normalfootgroup
2954 \expandafter\let\csname #1footnoterule\endcsname=%
2955 \normalfootnoterule
2956 \count\csname #1footins\endcsname=1000
2957 \dimen\csname #1footins\endcsname=\csuse{Xmaxhnotes@#1}
2958 \skip\csname #1footins\endcsname=\csuse{Xbeforenotes@#1}%
2959 \advance\skip\csname #1footins\endcsname by\csuse{Xafterrule@#1}%
2960 %
```

The `reledpar` provides tools in order to confine notes to one side. The mechanism is explained in the `reledpar`’s handbook. For now, just retain we need to store default value of the counter associated to the notes \TeX ’s inserts.

```
2961 \csxdef{default@#1footins}{1000}%Use this to confine the notes to one
2962 side only
2963 %
```

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).

```
2963 \ifnoledgroup@else%
2964 \expandafter\let\csname mpv#1footnote\endcsname=\mpnormalvfootnote
2965 \expandafter\let\csname mp#1footgroup\endcsname=\mpnormalfootgroup
2966 \count\csname mp#1footins\endcsname=1000
```

```

2967 \dimen\csname mp#1footins\endcsname=\csuse{Xmaxhnotes@#1}
2968 \skip\csname mp#1footins\endcsname=\csuse{Xbeforenotes@#1}%
2969 \advance\skip\csname mp#1footins\endcsname by\csuse{Xafterrule@#1}%
2970 \fi
2971 }
2972
2973 %

```

\normalvfootnote We now begin a series of commands that do ‘normal’ footnote formatting: a format much like that implemented in PLAIN T_EX, in which each footnote is a separate paragraph.

`\normalvfootnote` takes the series letter as #1 and the entire text of the footnote is #2. It does the `\insert` for this note, calling on the `\footfmt` macro for this note series to format the text of the note.

```

2974 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\normalvfootnote}[2]{%
2975 \iftoggle{Xgroupbyline@#1}{%In the case we use \Xgroupbyline, the
insertion is done later, in \add@Xgroupbyline.
2976 \prepare@Xgroupbyline{#1}{#2}{\normalvfootnote@inserted}%
2977 }{%In the case we don't use \Xgroupbyline, the insertion is made directly
2978 \X@beforeinsertion{#1}%
2979 \insert\csname #1footins\endcsname{%
2980 \X@atbegininsertion{#1}%
2981 \normalvfootnote@inserted{#1}{#2}%
2982 }%
2983 }%
2984 }%
2985 %

```

\normalvfootnote@inserted The `\normalvfootnote@inserted` macro is expanded to the content to be add to a `\insert` for normal critical footnote.

```

2986 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\
normalvfootnote@inserted}[2]{%
2987 \nottoggle{Xgroupbyline@#1}{\noindent}{\csuse{Xbhooknote@#1}%
2988 \csuse{Xnotefontsize@#1}%
2989 \footsplitskips
2990 \ifl@dpairing\ifl@dpadding\else%
2991 \setXnoteswidthliketwocolumns@{#1}%
2992 \fi\fi%
2993 \setXnotespositionliketwocolumns@{#1}%
2994 \spaceskip=\z@skip \xspaceskip=\z@skip%
2995 \csname #1footfmt\endcsname #2{#1}%
2996 }%
2997 %

```

```

\X@beforeinsertion#8 \newcommand{\X@beforeinsertion}[1]{%
2999 \if@ledgroup\else%
3000 \insert@Xtxtbeforenotes{#1}%

```

```

3001 \fi%
3002 \cuse{Xbeforeinserting@#1}%
3003 }%
3004 %

```

```

\beforeinsertion@X  \newcommand{\beforeinsertion@X}[1]{%
3006 \if@ledgroup\else%
3007 \insert@txtbeforenotesX{#1}%
3008 \fi%
3009 \cuse{beforeinsertingX@#1}%
3010 }%
3011 %

```

```

\X@atbegininsertion  \newcommand{\X@atbegininsertion}[1]{%
3013 \hspace=\expandafter\dimexpr\cuse{Xwidth@#1}\relax%
3014 }%
3015 %

```

And somewhat different versions of `\normalvfootnote` and `\normalvfootnote@inserted` for minipages.

```

\mpnormalvfootnote  \notbool{parapparatus@}{\newcommand*{\newcommand}{\mpnormalvfootnote}[2]{%
3017 \iftoggle{Xgroupbyline@#1}{%
3018 \prepare@Xgroupbyline{#1}{#2}{\mpnormalvfootnote@inserted}%
3019 }%
3020 {%
3021 \global\setbox\@nameuse{mp#1footins}%
3022 \vbox{%
3023 \unvbox\@nameuse{mp#1footins}%
3024 \mpnormalvfootnote@inserted{#1}{#2}%
3025 }%
3026 }%
3027 }%
3028
3029 %

```

```

\mpnormalvfootnote@inserted  \newcommand{\mpnormalvfootnote@inserted}[2]{%
3031 \noindent\cuse{Xhooknote@#1}%
3032 \cuse{Xnotefontsize@#1}%
3033 \hspace\columnwidth%
3034 \@parboxrestore%
3035 \color@begingroup%
3036 \csname #1footfmt\endcsname #2{#1}\color@endgroup%
3037 }%
3038 %

```

`\normalfootfmt` `\normalfootfmt` is a ‘normal’ macro to take the footnote line and page number information (see V.9 p. 104), 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.

```

3039 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\normalfootfmt}[4]{%
3040   \Xstorelineinfo{#1}{#4}%
3041   \nottoggle{Xgroupbyline@#4}{\Xledsetnormalparstuff{#4}}{ }%
3042   \hangindent=\csuse{Xhangindent@#4}%
3043   \everypar{\hangindent=\csuse{Xhangindent@#4}}%
3044   \nottoggle{Xgroupbyline@#4}{\rule\z@\splittopskip}{ }%
3045   {\printlinefootnote{#1}{#4}}%
3046   \print@lemma{#1}{#2}{#4}%
3047   \csuse{Xwrapcontent@#4}{#3}%
3048   \nottoggle{Xgroupbyline@#4}{\strut\par}{ }%
3049 }%
3050 %

```

`\normalfootstart` `\normalfootstart` is a standard footnote-starting macro, called in the output routine whenever there are footnotes of this series to be printed: it skips a bit and then draws a rule.

Any `\footstart` macro must put onto the page something that takes up space exactly equal to the `\skip\Xfootins` value for the associated series of notes. T_EX 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`.

```

3051 \newcommand*{\normalfootstart}[1]{%
3052 %

```

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. 234. Here is part of this algorithm, when the block of notes are ready to be printed.

```

3053   \ifdimequal{0pt}{\Xprenotes@}{ }%
3054   {%
3055     \iftoggle{Xprenotes@}{%
3056       \togglefalse{Xprenotes@}%

```

```

3057         \skip\csname #1footins\endcsname=%
3058         \glueexpr\csuse{Xprenotes@}+\csuse{Xafterrule@#1}\relax%
3059     }%
3060 }%
3061 }%
3062 \vskip\skip\csname #1footins\endcsname%
3063 %

```

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. 232 for the general description of the problem.

```

3064 \leftskipOpt \rightskipOpt
3065 \ifl@dpairing\else%
3066     \hsize=\old@hsize%
3067 \fi%
3068 \setXnoteswidthliketwocolumns@{#1}%
3069 \setXnotespositionliketwocolumns@{#1}%
3070 %

```

And now, print the footnote's rule to finish the footnote's introduction.

```

3071 \print@Xfootnoterule{#1}%
3072 }%
3073 %

```

`\normalfootgroup` `\normalfootgroup` is a standard footnote-grouping macro: it sends the contents of the footnote-insert box to the output page without alteration.

```

3074 \newcommand*{\normalfootgroup}[1]{%
3075     \csuse{Xbhookgroup@#1}%
3076     \unvbox\csname #1footins\endcsname%
3077     \hsize=\old@hsize%
3078 }%
3079
3080 %

```

`\mpnormalfootgroup` A somewhat different version for minipages. Note that, in this case, we do not make distinctions between the `\Xfootgroup` and `\Xfootstarts` macros.

```

3081 \unless\ifnoledgroup@
3082 \newcommand*{\mpnormalfootgroup}[1]{%
3083     \vskip\skip\@nameuse{mp#1footins}
3084     \ifl@dpairing\ifparledgroup%
3085         \leavevmode\marks\parledgroup@{begin}%
3086         \marks\parledgroup@series{#1}%
3087         \marks\parledgroup@type{Xfootnote}%
3088     \fi\fi\normalcolor%
3089     \ifparledgroup%
3090         \ifl@dpairing%
3091         \else%

```

```

3092 \setXnoteswidthliketwocolumns@{#1}%
3093 \setXnotespositionliketwocolumns@{#1}%
3094 \print@Xfootnoterule{#1}%%
3095 \fi%
3096 \else%
3097 \setXnoteswidthliketwocolumns@{#1}%
3098 \setXnotespositionliketwocolumns@{#1}%
3099 \print@Xfootnoterule{#1}%%
3100 \fi%
3101 \setlength{\parindent}{0pt}
3102 \csuse{Xhookgroup@#1}%
3103 \unvbox\csname mp#1footins\endcsname}}
3104 \fi
3105 %

```

XIII.6.3 Paragraphed footnotes

The paragraphed-footnote option reformats all the footnotes of one series for a page into a single paragraph; this is especially appropriate when the notes are numerous and brief. The code is based on *The TeXbook*, pp. 398–400, with alterations for our environment. This algorithm uses a considerable amount of save-stack space: a T_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 paragraphed; it takes the series letter as argument. We include the setting of `\count\footins` to 1000 for the footnote series just in case user is switching to paragraphed footnotes after having columnar ones, since they change this value (see below).

The argument of `\Xarrangement@footparagraph` is the letter denoting the series of notes to be paragraphed.

```

3106 \newcommand*{\Xarrangement@paragraph}[1]{%
3107 \csgdef{series@display#1}{paragraph}
3108 \expandafter\let\csname #1footstart\endcsname=\parafootstart
3109 \expandafter\let\csname v#1footnote\endcsname=\paravfootnote
3110 \expandafter\let\csname #1footfmt\endcsname=\parafootfmt
3111 \expandafter\let\csname #1footgroup\endcsname=\parafootgroup
3112 \count\csname #1footins\endcsname=1000
3113 \csxdef{default@#1footins}{1000}%Use this to confine the notes to one
side only
3114 \dimen\csname #1footins\endcsname=\csuse{Xmaxhnotes@#1}
3115 \skip\csname #1footins\endcsname=\csuse{Xbeforenotes@#1}%
3116 \advance\skip\csname #1footins\endcsname by\csuse{Xafterrule@#1}%
3117 \para@footsetup{#1}
3118 %

```

And the extra setup for minipages.

```

3119 \ifnoledgroup@else
3120 \expandafter\let\csname mpv#1footnote\endcsname=\mpparavfootnote

```



```

3121 \expandafter\let\csname mp#1footgroup\endcsname=\mpparafootgroup
3122 \count\csname mp#1footins\endcsname=1000
3123 \dimen\csname mp#1footins\endcsname=\csuse{Xmaxhnotes@#1}
3124 \skip\csname mp#1footins\endcsname=\csuse{Xbeforenotes@#1}%
3125 \advance\skip\csname mp#1footins\endcsname by\csuse{Xafterrule@#1}%
3126 \fi
3127 }
3128 %

```

`\footfudgefiddle` For paragraphed footnotes \TeX has to estimate the amount of space required. If it underestimates this then the notes may get too long and run off the bottom of the text block. `\footfudgefiddle` can be increased from its default 64 (say, to 70) to increase the estimate.

```

3129 \providecommand{\footfudgefiddle}{64}
3130 %

```

`\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 \mathTeX not `\hsize`. Peter Wilson have also included `\footfudgefiddle`.

```

3131 \newcommand*{\para@footsetup}[1]{\csuse{Xbhookgroup@#1}\csuse{
Xnotefontsize@#1}
3132 \setXnoteswidthliketwocolumns@{#1}%
3133 \ifcempty{Xwidth@#1}%
3134 {}%
3135 {\columnwidth=\expandafter\dimexpr\csuse{Xwidth@#1}\relax}%
3136 \dimen0=\baselineskip
3137 \multiply\dimen0 by 1024
3138 \divide \dimen0 by \columnwidth \multiply\dimen0 by \footfudgefiddle\
relax
3139 \csxdef{#1footfudgefactor}{%
3140 \expandafter\strip@pt\dimen0 }}
3141
3142 %

```

`\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.

```

3143 \newcommand*\parafootstart}[1]{%
3144   \rightskip=0pt \leftskip=0pt%
3145   \nottoggle{Xparindent@#1}{\parindent=\z@}{}%
3146   \ifdimequal{0pt}{\Xprenotes@}{}%
3147   {%
3148     \iftoggle{Xprenotes@}{%
3149       \togglefalse{Xprenotes@}%
3150       \skip\csname #1footins\endcsname=%
3151       \glueexpr\csuse{Xprenotes@}+\csuse{Xafterrule@#1}\relax%
3152     }%
3153   }%
3154 }%
3155 \vskip\skip\csname #1footins\endcsname%
3156 \setXnoteswidthliketwocolumns@{#1}%
3157 \setXnotespositionliketwocolumns@{#1}%
3158 \print@Xfootnoterule{#1}%
3159 \let\bidir@RTL@everypar\@empty%
3160 \noindent\leavevmode}
3161 %

```

`\parafootnote` `\parafootnote` is a version of the `\vfootnote` command that is used for paragraphed notes. It gets appended to the `\inserts@list` list by an outer-level footnote command like `\Afootnote`. The first argument is the note series letter; the second is the full text of the printed note itself, including line numbers, lemmata, and footnote text.

The initial model for this insertion is, of course, the `\insert\footins` definition in *The TeXbook*, p. 398. There, the footnotes are first collected up in hboxes, and these hboxes are later unpacked and stuck together into a paragraph.

However, Michael Downes has pointed out that because text in hboxes gets typeset in restricted horizontal mode, there are some undesirable side-effects if you later want to break such text across lines. In restricted horizontal mode, where \TeX does not expect to have to break lines, it does not insert certain items like `\discretionary`s. If you later unbox these hboxes and stick them together, as the *TeXbook* macros do to make these footnotes, you lose the ability to hyphenate after an explicit hyphen. This can lead to overfull hboxes when you would not expect to find them, and to the uninitiated it might be very hard to see why the problem had arisen.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

³²Michael Downes, ‘Line Breaking in `\unboxed` Text’, *TUGboat* 11 (1990), pp. 605–612.

³³See *The TeXbook*, p. 455 (editions after January 1990).

a paragraph consisting of a single long line. Later, there is an extra level of unboxing to be done: we have to unpack the `\vbox`, as well as the `hboxes` inside it, but that is not too hard. For details, we refer you to Michael's article, where the issues are clearly explained.³⁴ Michael's unboxing macro is called `\Xunvxh`: `unvbox`, extract the last line, and `unhbox` it.

Doing things this way has an important consequence: as Michael pointed out, you really can't put an explicit line-break into a note built in a `\vbox` the way we are doing.³⁵ In other words, be very careful not to use `\break`, or `\penalty-10000`, or any equivalent inside your para-footnote. If you do, most of the note will probably disappear. You *are* allowed to make strong suggestions; in fact `\penalty-9999` will be quite okay. Just do not make the break mandatory. We have not applied any of Michael's solutions here, since we feel that the problem is exiguous, and `reledmac` is quite baroque enough already. If you think you are having this problem, look up Michael's solutions.

One more thing; we set `\leftskip` and `\rightskip` to zero. This has the effect of neutralizing any such skips which may apply to the main text (cf. XIII.6.2 p. 182 above). We need to do this, since `\footfudgefactor` is calculated on the assumption that the notes are `\hsize` wide.

So, finally, here is the modified foot-paragraph code, which sets the footnote in vertical mode so that language and discretionary nodes are included.

```

3162 \newcommand*{\paravfootnote}[2]{%
3163   \csuse{Xbeforeinserting@#1}%
3164   \insert\csname #1footins\endcsname
3165   \bgroup
3166   \csuse{Xnotefontsize@#1}
3167   \footsplitskips
3168   \setbox0=\vbox{\hsize=\maxdimen%
3169     \let\ bidi@RTL@everypar\@empty%
3170     \insert@Xtxtbeforenotes{#1}%
3171     \noindent\csuse{Xbhooknote@#1}%
3172     \csname #1footfmt\endcsname #2{#1}}%
3173   \setbox0=\hbox{\Xunvxh{0}{#1}}%
3174   \dp0=0pt
3175   \ht0=\csname #1footfudgefactor\endcsname\wd0
3176 %

```

Here we produce the contents of the footnote from box 0, and add a penalty of 0 between boxes in this insert.

```

3177   \if@RTL\noindent \leavevmode\fi\box0%
3178   \penalty0
3179 \egroup}
3180 %
3181 %

```

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

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.

paragraphs. After trying out such a split (see *The TeXbook*, p. 124), T_EX 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.

```

3182 \newcommand*{\mpparavfootnote}[2]{%
3183   \global\setbox\@nameuse{mp#1footins}\vbox{%
3184     \unvbox\@nameuse{mp#1footins}%
3185     \csuse{Xnotefontsize@#1}
3186     \footsplitskips
3187     \setbox0=\vbox{\hsize=\maxdimen%
3188       \let\bidir@RTL@everypar\@empty%
3189       \insert@Xtxtbeforenotes{#1}%
3190       \noindent\color@begingroup%
3191       \csuse{Xbhooknote@#1}%
3192       \csname #1footfmt\endcsname #2{#1}\color@endgroup}%
3193     \setbox0=\hbox{\Xunvxh{0}{#1}}%
3194     \dp0=\z@
3195     \ht0=\csname #1footfudgefactor\endcsname\wd0
3196     \box0
3197     \penalty0
3198   }}
3199
3200 %

```

`\Xunvxh` Here is (modified) Michael's definition of `\unvxh`, used above. Michael's macro also takes care to remove some unwanted penalties and glue that T_EX automatically attaches to the end of paragraphs. When T_EX finishes a paragraph, it throws away any remaining glue, and then tacks on the following items: a `\penalty` of 10000, a `\parfillskip` and a `\rightskip` (*The TeXbook*, pp. 99–100). `\unvxh` cancels these unwanted paragraph-final items using `\unskip` and `\unpenalty`.

```

3201 \newcommand*{\Xunvxh}[2]{%
3202   \setbox0=\vbox{\unvbox#1%
3203     \global\setbox1=\lastbox}%
3204   \unhbox1
3205   \unskip           % remove \rightskip,
3206   \unskip           % remove \parfillskip,
3207   \unpenalty       % remove \penalty of 10000,
3208   \hskip\csuse{Xafternote@#2}\relax}% add the glue to go between the notes
3209
3210 %

```

`\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).

```

3211 \newcommand*{\parafootfmt}[4]{%
3212   \Xstorelineinfo{#1}{#4}%
3213   \Xinsertparafootsep{#4}%
3214   \ledsetnormalparstuff@common%
3215   \printlinefootnote{#1}{#4}%
3216   \print@lemma{#1}{#2}{#4}%
3217   \csuse{Xwrapcontent@#4}{#3}%
3218   \penalty-10 }
3219 %

```

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.

```

3220 \newcommand*{\parafootgroup}[1]{%
3221   \hsize=\expandafter\dimexpr\csuse{Xwidth@#1}\relax%
3222   \unvbox\csname #1footins\endcsname
3223   \ifcsstring{Xragged@#1}{L}{\RaggedLeft}{}%
3224   \ifcsstring{Xragged@#1}{R}{\RaggedRight}{}%
3225   \makehboxofhboxes
3226   \setbox0=\hbox{\unhbox0 \removehboxes}%
3227   \csuse{Xbhookgroup@#1}%
3228   \csuse{Xnotefontsize@#1}%
3229   \unhbox0\par%
3230   \global\hsize=\old@hsize%
3231   }%
3232
3233 %

```

`\mpparafootgroup` The minipage version.

```

3234 \newcommand*{\mpparafootgroup}[1]{%
3235   \setXnoteswidthliketwocolumns@{#1}%
3236   \vskip\skip\@nameuse{mp#1footins}
3237   \ifl@dpairing\ifparledgroup%
3238     \leavevmode\marks\parledgroup@{begin}%
3239     \marks\parledgroup@series{#1}%

```

```

3240 \marks\parledgroup@type{Xfootnote}%
3241 \fi\fi\normalcolor
3242 \ifparledgroup%
3243 \ifl@dpairing%
3244 \else%
3245 \setXnoteswidthliketwocolumns@{#1}%
3246 \setXnotespositionliketwocolumns@{#1}%
3247 \print@Xfootnoterule{#1}%%
3248 \fi%
3249 \else%
3250 \setXnoteswidthliketwocolumns@{#1}%
3251 \setXnotespositionliketwocolumns@{#1}%
3252 \print@Xfootnoterule{#1}%
3253 \fi%
3254 \unvbox\csname mp#1footins\endcsname
3255 \ifcsstring{Xragged@#1}{L}{\RaggedLeft}{}%
3256 \ifcsstring{Xragged@#1}{R}{\RaggedRight}{}%
3257 \makehboxofhboxes
3258 \setbox0=\hbox{\unhbox0 \removehboxes}%
3259 \csuse{Xhookgroup@#1}%
3260 \csuse{Xnotefontsize@#1}%
3261 \nottoggle{Xparindent@#1}{\parindent=\z@}{}%
3262 \unhbox0\par}}
3263
3264 %

```

And finally, the two macros which are required to transform the long horizontal box stored in the insert' box to a printable text.

```

\makehboxofhboxes65 \newcommand*\makehboxofhboxes{\setbox0=\hbox{}}%
\removehboxes66 \loop
3267 \unpenalty
3268 \setbox2=\lastbox
3269 \ifhbox2
3270 \setbox0=\hbox{\box2\unhbox0}%
3271 \repeat}
3272
3273 \newcommand*\removehboxes{\setbox0=\lastbox
3274 \ifhbox0{\removehboxes}\unhbox0 \fi}
3275
3276 %

```

Insertion of the footnotes separator The command `\Xinsertparafootsep{<series>}` must be called at the beginning of `\parafootftm`.

```

\prevpage@num77 \newcommand{\Xinsertparafootsep}[1]{%
\Xinsertparafootsep78 \ifledRcol@%
3279 \ifnumequal{\csuse{#1prevpage@numR}}{\page@numR}%

```

```

3280     {\ifcsdef{prevline#1}% Be sur \prevline#1 exists.
3281       {\ifcsequal{prevline#1}{lineinfo}%
3282         {\ifcseempty{Xsymlinenum@#1}{\csuse{Xparafootsep@#1}}{}}%
3283         {\csuse{Xparafootsep@#1}}%
3284       }%
3285       {\csuse{Xparafootsep@#1}}%
3286     }%
3287   {}%
3288   \global\csname #1prevpage@numR\endcsname=\page@numR%
3289 \else%
3290   \ifnumequal{\csuse{#1prevpage@num}}{\page@num}%
3291     {\ifcsdef{prevline#1}% Be sur \prevline#1 exists.
3292       {\ifcsequal{prevline#1}{lineinfo}%
3293         {\ifcseempty{Xsymlinenum@#1}{\csuse{Xparafootsep@#1}}{}}%
3294         {\csuse{Xparafootsep@#1}}%
3295       }%
3296       {\csuse{Xparafootsep@#1}}%
3297     }%
3298   {}%
3299   \global\csname #1prevpage@num\endcsname=\page@num%
3300 \fi%
3301 }
3302 %

```

XIII.6.4 Columnar footnotes

Common tools

`\rigidbalance` We will now define macros for three-column notes and two-column notes. Both sets of macros will use `\rigidbalance`, which splits a box (#1) into a number (#2) of columns, each with a space (#3) between the top baseline and the top of the `\vbox`. The `\Xrigidbalance` macro is taken from *The TeXbook*, p. 397, with a slight change to the syntax of the arguments so that they do not depend on white space. Note also the extra unboxing in `\splitoff`, which allows the new `\vbox` to have its natural height as it goes into the alignment.

The \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`.

```

3303 \newcount\@k \newdimen\@h
3304 \newcommand*\Xrigidbalance}[3]{%
3305   \hsize=\expandafter\dimexpr\csuse{Xwidth@\@currentseries}\relax%
3306   \rigidbalance{#1}{#2}{#3}%
3307 }%
3308
3309 \newcommand*\rigidbalanceX}[3]{%
3310   \hsize=\expandafter\dimexpr\csuse{widthX@\@currentseries}\relax%
3311   \rigidbalance{#1}{#2}{#3}%

```

```

3312 }%
3313
3314 \newcommand*\rigidbalance}[3]{%
3315   \setbox0=\box#1 \@k=#2 \@h=#3%
3316   \@@line{\splittopskip=\@h \vbadness=\@M \hfilneg
3317   \valign{##\vfil\cr\dosplits}}
3318
3319 \newcommand*\dosplits{\ifnum\@k>0 \noalign{\hfil}\splitoff
3320   \global\advance\@k-1\cr\dosplits\fi}
3321
3322 \newcommand*\splitoff{\dimen0=\ht0
3323   \divide\dimen0 by\@k \advance\dimen0 by\@h
3324   \setbox2 \vsplit0 to \dimen0
3325   \unvbox2 }
3326
3327 %

```

Three columns

```

\Xarrangement@threecol}28 \newcommand*\Xarrangement@threecol}[1]{%
3329   \csgdef{series@display#1}{threecol}
3330   \expandafter\let\csname v#1footnote\endcsname=\threecolvfootnote
3331   \expandafter\let\csname #1footfmt\endcsname=\threecolfootfmt
3332   \expandafter\let\csname #1footgroup\endcsname=\threecolfootgroup
3333   \dimen\csname #1footins\endcsname=\csuse{Xmaxhnotes@#1}%
3334   \skip\csname #1footins\endcsname=\csuse{Xbeforenotes@#1}%
3335   \advance\skip\csname #1footins\endcsname by\csuse{Xafterrule@#1}%
3336   \threecolfootsetup{#1}
3337 %

```

The additional setup for minipages.

```

3338   \ifnoledgroup@else
3339     \expandafter\let\csname mpv#1footnote\endcsname=\mpnormalvfootnote
3340     \expandafter\let\csname mp#1footgroup\endcsname=\mpthreecolfootgroup
3341     \skip\csname mp#1footins\endcsname=\csuse{Xbeforenotes@#1}%
3342     \advance\skip\csname mp#1footins\endcsname by\csuse{Xafterrule@#1}%
3343     \mpthreecolfootsetup{#1}
3344   \fi
3345 }
3346
3347 %

```

The `\footstart` and `\footnoterule` macros for these notes assume the normal values (XIII.6.2 p. 182 above).

`\threecolfootsetup` The `\threecolfootsetup` macro calculates and sets some numbers for three-column footnotes.

We set the `\count` of the foot insert to 333. Each footnote can be thought of as contributing only one third of its height to the page, since the footnote insertion has

been made as a long narrow column, which then gets trisected by the `\rigidbalance` routine (inside `\threecolfootgroup`). These new, shorter columns are saved in a box, and then that box is *put back* into the footnote insert, replacing the original collection of the footnotes. This new box is, therefore, only about a third of the height of the original one.

The `\dimen` value for this note series has to change in the inverse way: it needs to be three times the actual limit on the amount of space these notes are allowed to fill on the page, because when \TeX is accumulating material for the page and checking that limit, it does not apply the `\count` scaling.

```

3348 \newcommand*\threecolfootsetup}[1]{%
3349   \count\csname #1footins\endcsname 333
3350   \csxdef{default@#1footins}{333}%Use this to confine the notes to one
side only
3351   \multiply\dimen\csname #1footins\endcsname \thr@@}
3352 %

```

`\mpthreecolfootsetup` The setup for minipages.

```

3353 \newcommand*\mpthreecolfootsetup}[1]{%
3354   \count\csname mp#1footins\endcsname 333
3355   \multiply\dimen\csname mp#1footins\endcsname \thr@@}
3356 %
3357 %

```

`\threecolvfootnote` `\threecolvfootnote` This is the `\vfootnote` command for three-column notes. However, most of the code is deported on `\threecolvfootnote@inserted`. The call to `\Xnotefontsize@<s>` ensures that the `\splittopskip` and `\splitmaxdepth` take their values from the right `\strutbox`: the one used in a footnotes. Note especially the importance of temporarily reducing the `\hsize` to 0.3 of its normal value. This determines the widths of the individual columns. So if the normal `\hsize` is (say) 10 cm, then each column will be $0.3 \times 10 = 3$ cm wide, leaving a gap of 1 cm spread equally between columns (i.e., .5 cm between each).

The arguments are #1 the note series letter and #2 the full text of the note (including numbers, lemma and text).

```

3358 \notbool{parapparus@}{\newcommand*{\newcommand}\threecolvfootnote}[2]{%
3359   \iftoggle{Xgroupbyline@#1}{%
3360     \prepare@Xgroupbyline{#1}{#2}{\threecolvfootnote@inserted}%
3361   }%
3362   {%
3363     \X@beforeinsertion{#1}%
3364     \insert\csname #1footins\endcsname{%
3365       \threecolvfootnote@inserted{#1}{#2}%
3366     }%
3367   }%
3368 }%
3369 %

```

```

\threecolfootnote@inserted70 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\
threecolfootnote@inserted}[2]{%
3371 \hspace=\expandafter\dimexpr\csuse{Xwidth@#1}\relax%
3372 \noindent\csuse{Xhooknote@#1}%
3373 \csuse{Xnotefontsize@#1}%
3374 \footsplitskips%
3375 \csname #1footfmt\endcsname #2{#1}%
3376 }%
3377 %

```

`\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.

```

3378 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\threecolfootfmt}[4]{%
3379 \Xstorelineinfo{#1}{#4}%
3380 \threecol@begin@insert{#4}%
3381 \hspace{\parindent}%
3382 \printlinefootnote{#1}{#4}%
3383 \print@lemma{#1}{#2}{#4}%
3384 \csuse{Xwrapcontent@#4}{#3}%
3385 \nottoggle{Xgroupbyline@#4}%
3386 {\strut\par\allowbreak}%
3387 }%
3388 }%
3389 %

```

`\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`.

```

3390 \newcommand{\threecol@begin@insert}[1]{%
3391 \normal@pars%
3392 \nottoggle{Xgroupbyline@#1}%
3393 {\hspace \csuse{Xhsizethreecol@#1}}%
3394 }%
3395 \nottoggle{Xparindent@#1}{\parindent=\z@}{}%
3396 \tolerance=5000%
3397 \hangindent=\csuse{Xhangindent@#1}%
3398 \everypar{\hangindent=\csuse{Xhangindent@#1}}%
3399 \@tempdima=\parindent%
3400 \csuse{Xcolalign@#1}%
3401 \parindent=\@tempdima%
3402 \strut%
3403 }%
3404 %

```

`\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.

```

3405 \newcommand*\threecolfootgroup}[1]{%
3406   \csuse{Xbhookgroup@#1}\par%
3407   \splittopskip=\ht\strutbox
3408   \expandafter
3409   \Xrigidbalance\csname #1footins\endcsname \thr@@ \splittopskip}
3410 %

```

`\mpthreecolfootgroup` The setup for minipages.

```

3411 \newcommand*\mpthreecolfootgroup}[1]{%
3412   \vskip\skip\@nameuse{mp#1footins}
3413   \ifl@dpairing\ifparledgroup%
3414     \leavevmode\marks\parledgroup@{begin}%
3415     \marks\parledgroup@series{#1}%
3416     \marks\parledgroup@type{Xfootnote}%
3417   \fi\fi\normalcolor
3418   \ifparledgroup%
3419     \ifl@dpairing%
3420     \else%
3421       \setXnoteswidthliketwocolumns@{#1}%
3422       \setXnotespositionliketwocolumns@{#1}%
3423       \print@Xfootnoterule{#1}%
3424     \fi%
3425   \else%
3426     \setXnoteswidthliketwocolumns@{#1}%
3427     \setXnotespositionliketwocolumns@{#1}%
3428     \print@Xfootnoterule{#1}%
3429   \fi%
3430   \csuse{Xbhookgroup@#1}\par%
3431   \splittopskip=\ht\strutbox
3432   \expandafter
3433   \Xrigidbalance\csname mp#1footins\endcsname \thr@@ \splittopskip}}
3434 %
3435 %

```

Two columns

```

\Xarrangement@twocol \newcommand*\Xarrangement@twocol}[1]{%
3437   \csgdef{series@display#1}{twocol}
3438   \expandafter\let\csname v#1footnote\endcsname=\twocolvfootnote
3439   \expandafter\let\csname #1footfmt\endcsname=\twocolfootfmt

```

```

3440 \expandafter\let\csname #1footgroup\endcsname=\twocolfootgroup
3441 \dimen\csname #1footins\endcsname=\csuse{Xmaxhnotes@#1}%
3442 \skip\csname #1footins\endcsname=\csuse{Xbeforenotes@#1}%
3443 \advance\skip\csname #1footins\endcsname by\csuse{Xafterrule@#1}%
3444 \twocolfootsetup{#1}
3445 %

```

The additional setup for minipages.

```

3446 \ifnoledgroup@else
3447 \expandafter\let\csname mpv#1footnote\endcsname=\mpnormalvfootnote
3448 \expandafter\let\csname mp#1footgroup\endcsname=\mptwocolfootgroup
3449 \skip\csname mp#1footins\endcsname=\csuse{Xbeforenotes@#1}%
3450 \advance\skip\csname mp#1footins\endcsname by\csuse{Xafterrule@#1}%
3451 \mptwocolfootsetup{#1}
3452 \fi
3453 }
3454
3455 %

```

`\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 `\hsiz`.

`\twocolvfootnote@inserted`
`\twocolfootfmt`
`\twocolfootgroup`

```

3456 \newcommand*{\twocolfootsetup}[1]{%
3457 \count\csname #1footins\endcsname 500
3458 \csxdef{default@#1footins}{500}%
3459 \multiply\dimen\csname #1footins\endcsname \tw@}
3460 %

3461 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\twocolvfootnote}[2]{%
3462 \iftoggle{Xgroupbyline@#1}{%
3463 \prepare@Xgroupbyline{#1}{#2}{\twocolvfootnote@inserted}%
3464 }{%
3465 \X@beforeinsertion{#1}%
3466 \insert\csname #1footins\endcsname{%
3467 \twocolvfootnote@inserted{#1}{#2}%
3468 }%
3469 }%
3470 }%

3471 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\twocolvfootnote@inserted}[2]{%
3472 \hsiz=\expandafter\dimexpr\csuse{Xwidth@#1}\relax%
3473 \noindent\csuse{Xbhooknote@#1}%
3474 \csuse{Xnotefontsize@#1}%
3475 \footplitskips%
3476 \csname #1footfmt\endcsname #2{#1}%
3477 }%
3478 %

```

```

3479 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\twocolfootfmt}[4]{% 4th
arg is optional, for backward compatibility
3480 \Xstorelineinfo{#1}{#4}%
3481 \twocol@begin@insert{#4}%
3482 \hspace{\parindent}%
3483 \printlinefootnote{#1}{#4}%
3484 \print@lemma{#1}{#2}{#4}%
3485 \csuse{Xwrapcontent@#4}{#3}%
3486 \nottoggle{Xgroupbyline@#4}%
3487   {\strut\par\allowbreak}%
3488   }%
3489 }%
3490 %

3491 \newcommand{\twocol@begin@insert}[1]{%
3492   \normal@pars%
3493   \hsize \csuse{Xhsizetwocol@#1}%
3494   \nottoggle{Xparindent@#1}{\parindent=\z@}{}%
3495   \tolerance=5000%
3496   \hangindent=\csuse{Xhangindent@#1}%
3497   \everypar{\hangindent=\csuse{Xhangindent@#1}}%
3498   \@tempdima=\parindent%
3499   \csuse{Xcolalign@#1}%
3500   \parindent=\@tempdima%
3501   \strut%
3502 }%
3503
3504 \newcommand*{\twocolfootgroup}[1]{%
3505   \csuse{Xbhookgroup@#1}\par%
3506   \splittopskip=\ht\strutbox
3507   \expandafter
3508   \Xrigidbalance\csname #1footins\endcsname \tw@ \splittopskip}
3509
3510 %

```

`\mptwocolfootsetup` The versions for minipages.

`\mptwocolfootgroup`

```

3511 \newcommand*{\mptwocolfootsetup}[1]{%
3512   \count\csname mp#1footins\endcsname 500
3513   \multiply\dimen\csname mp#1footins\endcsname \tw@}
3514 %

3515 \newcommand*{\mptwocolfootgroup}[1]{%
3516   \vskip\skip\@nameuse{mp#1footins}
3517   \ifl@dpairing\ifparledgroup%
3518     \leavevmode\marks\parledgroup@{begin}%
3519     \marks\parledgroup@series{#1}%
3520     \marks\parledgroup@type{Xfootnote}%
3521     \fi\fi\normalcolor
3522   \ifparledgroup%

```

```

3523 \ifl@pairing%
3524 \else%
3525   \setXnoteswidthliketwocolumns@{#1}%
3526   \setXnotespositionliketwocolumns@{#1}%
3527   \print@Xfootnoterule{#1}%
3528   \fi%
3529 \else%
3530   \setXnoteswidthliketwocolumns@{#1}%
3531   \setXnotespositionliketwocolumns@{#1}%
3532   \print@Xfootnoterule{#1}%
3533   \fi%
3534   \csuse{Xbhookgroup@#1}\par%
3535   \splittopskip=\ht\strutbox
3536   \expandafter
3537   \Xrigidbalance\csname mp#1footins\endcsname \tw@ \splittopskip}}
3538
3539 %

```

XIII.7 Critical notes presentation

Here, we define some common macro which are used in order to print a critical notes, that is a note with 1) line number 2) lemma 3) lemma separator 4) text associated to the lemma.

XIII.7.1 Font tools

`\endashchar` The fonts that are used for printing notes might not have the character mapping we expect: for example, the Computer Modern font that contains old-style numerals does not contain an en-dash or square brackets, and its period and comma are in odd locations. To allow use of the standard footnote macros with such fonts, we use the following macros for certain characters.

The `\endashchar` macro is simply an en-dash from the normal font and is immune to changes in the surrounding font. The same goes for the full stop. These two are used in `\printlines`. The right bracket macro is the same again; it crops up in `\normalfootfmt` and the other footnote macros for controlling the format of the footnotes.

Note that these commands are not directly called by `reledmac`, but are enclosed as default value of specific hooks. Consequently, people should not redefine them, but use instead the `\Xlinrangeseparator`, `\Xendlinrangeseparator`, `\Xsublinesep`, `\Xendsublinesep` and `\Xlemmaseparator` macros.

With `polyglossia`, each critical note has a `\footnote@lang` which shows the language of the lemma, and which can be used to switch the bracket from right to left.

```

3540 \def\endashchar{\textnormal{--}}
3541
3542 \newcommand*\fullstop{\textnormal{.}}
3543 \def\Xsublinesep@side{\fullstop}
3544

```

```

3545 \newcommand*{\rbracket}{\textnormal{%
3546   \csuse{text\csuse{footnote@lang}}{%
3547     \ifluatex%
3548     \ifdefstring{\footnote@luatextextdir}{TRT}{\thinspace[]{\thinspace
}}%
3549     \else%
3550     \thinspace}%
3551     \fi}%
3552   }%
3553 }
3554
3555 %

```

XIII.7.2 Pstart number in footnote

`\printpstart` The `\printpstart` macro prints the pstart number for a note.

```

3556 \newcommand{\printpstart}[0]{%
3557   \ifboolexpr{bool{!@dpairing} or bool{!@dprintingpages} or bool{
!@dprintingcolumns}}{%
3558     \ifledRcol%
3559     \thepstartR%
3560     \else%
3561     \thepstartL%
3562     \fi%
3563   }{%
3564     \thepstart%
3565   }%
3566 }
3567 %

```

XIII.7.3 Lemma printing

`\print@lemma` `\print@lemma` is called inside critical footnotes to print the lemma and the lemma separator (#1: line number and font information, #2: lemma, #3: series).

```

3568 %
3569 \newcommand{\print@lemma}[3]{%
3570   \bgroup%
3571   \nottoggle{Xlemmadisablefontselection@#3}%
3572   {\select@lemmafont#1|}%
3573   }%
3574   \bgroup%
3575   \csuse{Xlemmafont@#3}%Deprecated
3576   \csuse{Xwraplemma@#3}{#2}%
3577   \egroup%
3578   \egroup%
3579   \iftoggle{nosep@}{%
3580     \hskip\csuse{Xinplaceoflemmaseparator@#3}%

```

```

3581 \relax%
3582 }%
3583 {\ifcseempty{Xlemmaseparator@#3}%
3584 {%
3585   \hskip\csuse{Xinplaceoflemmaseparator@#3}%
3586   \relax%
3587 }%
3588 {%
3589   \nobreak%
3590   \hskip\csuse{Xbeforelemmaseparator@#3}%
3591   \csuse{Xlemmaseparator@#3}%
3592   \hskip\csuse{Xafterlemmaseparator@#3}%
3593   \relax%
3594 }%
3595 }%
3596 }%
3597 %

```

XIII.7.4 Line number printing

\Xstorelineinfo The `\Xstorelineinfo` macro is used to store some data about line number of the current critical footnote, data which will be reused later for the `\Xnumberonlyfirstinline` and related setting.

#1 footnote specification for the current footnote ; #2 footnote series.

```

3598 \newcommand{\Xstorelineinfo}[2]{%
3599   \l@dp@rsefootspec#1|}%
3600   \iftoggle{Xnumberonlyfirstintwolines@#2}{%
3601     \xdef\lineinfo@{\l@dparsedstartline - \l@dparsedstartsub - \
3602     \l@dparsedendline - \l@dparsedendsub}%
3603     }%
3604     {%
3605       \xdef\lineinfo@{\l@dparsedstartline - \l@dparsedstartsub}%
3606       }%
3607 }%

```

\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`.

```

3608 \newcommand{\printlinefootnote}[2]{%
3609   \iftoggle{nonum@}{%Try if the line number must printed for this specific
3610   not (by default, yes)
3611   \hspace{\csuse{Xinplaceofnumber@#2}}%
3612   }%
3613   {%

```



```

3614 \iftoggle{Xnonumber@#2}%Try if the line number must printed (by
default, yes)
3615 {%
3616 \hspace{\csuse{Xinplaceofnumber@#2}}%
3617 }%
3618 {%
3619 {\iftoggle{Xnumberonlyfirstinline@#2}% If for this series the
line number must be printed only in the first time.
3620 {%
3621 \ifcsdef{prevline#2}%
3622 {%Be sure the \prevline exists.
3623 \ifcsequal{prevline#2}{\lineinfo@}%Try it
3624 {%
3625 \ifcsequal{Xsymlinenum@#2}%
3626 {%
3627 \hspace{\csuse{Xinplaceofnumber@#2}}%
3628 }%
3629 {\printsymlinefootnotearea{#2}}%
3630 }%
3631 {%
3632 \printlinefootnotearea{#1}{#2}%
3633 }%
3634 }%
3635 {%
3636 \printlinefootnotearea{#1}{#2}%
3637 }%
3638 }%
3639 {%
3640 \printlinefootnotearea{#1}{#2}%
3641 }%
3642 \csxdef{prevline#2}{\lineinfo@}%
3643 }%
3644 }%
3645 }%
3646 }%
3647 }
3648 %

```

`\printsymlinefootnotearea` This macro prints the space before the line symbol, changes the font, when prints the line symbol and the space after it.

```

3649 \newcommand{\printsymlinefootnotearea}[1]{%
3650 \hspace{\csuse{Xbeforesymlinenum@#1}}%
3651 \csuse{Xnotenumfont@#1}%
3652 \ifdimequal{\csuse{Xboxsymlinenum@#1}}{\z@}%
3653 {\csuse{Xsymlinenum@#1}}%
3654 {\hbox to \csuse{Xboxsymlinenum@#1}%
3655 {\csuse{Xsymlinenum@#1}\hfill}}%
3656 }%
3657 \hspace{\csuse{Xaftersymlinenum@#1}}%

```

```
3658 }%
3659 %
```

`\printlinefootnotearea` This macro prints the space before the line number, changes the font, then prints the line number and the space after it. It is called by `\printlinefootnote` depending of the options about repeating line numbers. The first argument is line information, the second is the notes series (A, B, C, etc.)

```
3660 \newcommand{\printlinefootnotearea}[2]{%
3661   \printXbeforenumber{#2}%
3662   \csuse{Xnotenumfont@#2}%
3663   \boxfootnotenumbers{#1}{#2}%
3664   \printXafternumber{#2}%
3665 }%
3666 %
```

`\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.

```
3667 \newcommand{\boxfootnotenumbers}[2]{%
3668   \ifdimequal{\csuse{Xboxlinenum@#2}}{0pt}{%
3669     \printlinefootnotenumbers{#1}{#2}%
3670   }%
3671   {%
3672     \hbox to \csuse{Xboxlinenum@#2}%
3673     {%
3674       \IfSubStr{RC}{\csuse{Xboxlinenumalign@#2}}{\hfill}{}%
3675       \printlinefootnotenumbers{#1}{#2}%
3676       \IfSubStr{LC}{\csuse{Xboxlinenumalign@#2}}{\hfill}{}%
3677     }%
3678   }%
3679 }%
3680 %
```

`\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.

```
3681 \newcommand{\printlinefootnotenumbers}[2]{%
3682   \xdef\@currentseries{#2}%
3683   \ifboolexpr{%
3684     (togl{Xpstart@#2} and bool{numberpstart})%
3685     or togl{Xpstarteverytime@#2}}%
3686   {\printpstart}{}%
3687   \iftoggle{Xstanza@#2}{%
3688     \ifnumberstanza%
3689     \printstanza%
3690     \csuse{Xstanzaseparator@#2}%

```

```

3691     \fi%
3692   }{}%
3693   \iftoggle{Xonlypstart@#2}{}{%
3694     \csuse{Xtxtbeforenumber@#2}%
3695     \printlines#1|\ifledRcol@{\Rlineflag\fi}}%
3696   }%
3697   %

```

\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.)

```

3698 \newcommand{\printXbeforenumber}[1]{%
3699   \hspace{\csuse{Xbeforenumber@#1}}%
3700 }%
3701 %

```

\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

```

3702 \newcommand{\printXafternumber}[1]{%
3703   \iftoggle{Xnonbreakableafternumber@#1}{\nobreak}{}%
3704   \hspace{\csuse{Xafternumber@#1}}%
3705 }%
3706 %

```

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. 104: 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@pn` for page numbers;
- `\ifl@d@ssub` for starting sub-line;
- `\ifl@d@elin` for ending line;
- `\ifl@d@esl` for ending sub-line; and
- `\ifl@d@dash` for the dash between the starting and ending groups.

There is no boolean for the line number because it is always printed.

Maieul Rouquette has added `\ifl@d@Xtwolines` and `\ifl@d@Xmorethantwolines` to print a symbol which stands for “and subsequent” when there are two, three or more lines. Is also defines `\@annot@start@print` and `\@annot@end@print` which defines annotation associated to the start line numbers and end linenumbers in critical footnotes.

We print the page numbers only if: 1) we are doing the lineation by page, and 2) the ending page number is different from the starting page number.a

```

3742 \ifbypage@
3743   \ifnum#4=#1 \else
3744     \l@d@pnumtrue
3745     \l@d@dashtrue
3746   \fi
3747 \fi
3748 %

```

We print the ending line number if: (1) we are printing the ending page number, or (2) the ending line number is different from the starting line number. If one of these condition is true, we also print the annotation linked to the ending line number annotations.

```

3749 \ifboolexpr{%
3750   bool{\l@d@pnum}%
3751   or not test{\ifnumequal{#2}{#5}}%
3752 }{%
3753   \l@d@elintrue%
3754   \l@d@dashtrue%
3755   \unless\ifx\relax\annot@end%
3756     \def\@annot@end@print{%
3757       \l@wrapcs@ifnotempty{Xwraplinenumannotation@\@currentseries}{\
annot@end}%
3758     }%
3759   \fi%
3760   }{%
3761 %

```

We define the starting line number annotation as a merge of the starting annotation and ending annotation if we don't print ending line number. Otherwise, it is only the starting annotation.

```

3762 \ifl@d@elin%
3763   \def\@annot@start@print{%
3764     \l@wrapcs@ifnotempty{Xwraplinenumannotation@\@currentseries}{\
annot@start}%
3765   }%
3766   \else%
3767     \ifx\annot@start\annot@end%
3768       \unless\ifx\@annot@start\relax%
3769         \def\@annot@start@print{%
3770           \l@wrapcs@ifnotempty{Xwraplinenumannotation@\@currentseries}{\
annot@start}%
3771         }%
3772     \fi%
3773   \else%
3774     \ifx\@annot@end@print\relax%
3775       \def\@annot@start@print{%
3776         \l@wrapcs@ifnotempty{Xwraplinenumannotation@\@currentseries}{%

```

```

3777     \ifx\annot@start\empty\else%
3778         \annot@start%
3779         \ifdefined\linrangesep@%
3780             \linrangesep@%
3781         \else%
3782             \csuse{Xlinrangeseparator@\@currentseries}%
3783         \fi%
3784     \fi%
3785     \annot@end%
3786 }%
3787 }%
3788 \else%
3789     \let\@annot@start@print\@annot@end@print%
3790     \let\@annot@end@print\relax%
3791 \fi%
3792 \fi%
3793 \fi%
3794 %

```

We print the starting sub-line if it is nonzero.

```

3795 \l@d@ssubfalse
3796 \ifnum#3=0 \else
3797     \l@d@ssubtrue
3798 \fi
3799 %

```

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.

```

3800 \l@d@eslfalse
3801 \ifnum#6=0 \else
3802     \ifnum#6=#3
3803         \ifl@d@elin \l@d@esltrue \else \l@d@eslfalse \fi
3804     \else
3805         \l@d@esltrue
3806         \l@d@dashtrue
3807     \fi
3808 \fi%
3809 %

```

However, if the `\Xtwolines` is set for the current series, we do not print the last line number.

```

3810 \ifl@d@dash%
3811     \ifboolexpr{togl{fulllines@} or test{\ifcsempy{Xtwolines@\@currentseries}}}%
3812     {%
3813     {%
3814     \setistwofollowinglines{#1}{#2}{#4}{#5}%
3815     \ifboolexpr{%
3816     (%

```

```

3817         togl {Xtwolinesbutnotmore@\@currentseries}%
3818         and not%
3819             (%
3820                 bool {istwofollowinglines@}%
3821             )%
3822     )%
3823     or%
3824     (%
3825         (not test{\ifnumequal{#1}{#4}})%
3826         and togl{Xtwolinesonlyinsamepage@\@currentseries}%
3827     )%
3828     }%
3829     {}%
3830     {%
3831         \l@d@dashfalse%
3832         \l@d@Xtwolinestrue%
3833         \l@d@elinfalse%
3834         \l@d@eslfalse%
3835         \ifcsemt{Xmorethantwolines@\@currentseries}%
3836             {}%
3837             {\ifistwofollowinglines\else%
3838                 \l@d@Xmorethantwolinestrue%
3839             }%
3840         }%
3841     }%
3842 }%
3843 \fi%
3844 %

```

End of \setprintlines.

```

3845 }%
3846 %

```

`\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.

```

3847 \newif\ifistwofollowinglines@%
3848 \newcommand{\setistwofollowinglines}[4]{%
3849   \ifcsdef{lastlinenumberon@#1}%
3850     {\numdef{\tmp}{\csuse{lastlinenumberon@#1}}}%
3851     {\numdef{\tmp}{0}}%
3852   \istwofollowinglines@false%
3853   \ifnumequal{#4-#2}{1}%
3854     {\istwofollowinglines@true}%
3855   {\ifbypage@%
3856     \ifnumequal{#3-#1}{1}%
3857     {%
3858       \ifnumequal{#2}{\tmp}%
3859       {\ifnumequal{#4}{1}{\istwofollowinglines@true}{}}%
3860       {}}%
3861     }%
3862     {}%
3863   \fi%
3864 }%
3865 }%
3866 %

```

`\printlines` So, we have decided which part of line number sets will be printed depending of these value. Now we are ready to print them. If the lineation is by pstart, we print the pstart. Arguments are 1) start page number 2) start line number 3) start subline number 4) end page number 5) end line number 6) end subline number 7) font specification 8) side flag

```

3867 \def\printlines#1|#2|#3|#4|#5|#6|#7|#8|{%
3868   \begingroup%
3869 %

```

If we use Lua \TeX , ensure we use good text's direction.

```

3870   \ifluatex%
3871     \edef\@tmp{\the\textdir}%
3872     \ifdefstring{\@tmp}{TLT}{\textdir TLT}%Test in order to prevent
spurious space (bug #397)
3873   \fi%
3874 %

```

Decide which part of line number components we will print.

```

3875   \setprintlines{#1}{#2}{#3}{#4}{#5}{#6}%
3876 %

```

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.


```

3877 \ifdimequal{\csuse{Xboxstartlinenum@\@currentseries}}{0pt}%
3878 {\bgroup}%
3879 {\leavevmode\hbox to \csuse{Xboxstartlinenum@\@currentseries}\bgroup\
hfill}%
3880 \ifcsstring{Xlinenumannotationposition@\@currentseries}{before}%
3881 {\@annot@start@print}%
3882 {}%
3883 \ifl@d@pnum%
3884 \wrap@edcrossref{\@this@crossref@start}{#1}%
3885 \csuse{Xpagelinesep@\@currentseries}%
3886 \fi%
3887 \wrap@edcrossref{\@this@crossref@start}{%
3888 \linenumrep{#2}%
3889 \iftoggle{Xlineflag@\@currentseries}{#8}{-}%
3890 }%
3891 \ifl@d@ssub%
3892 \csuse{Xsublinesep@\@currentseries}%
3893 \wrap@edcrossref{\@this@crossref@start}{\sublinenumrep{#3}}%
3894 \fi
3895 \ifcsstring{Xlinenumannotationposition@\@currentseries}{after}%
3896 {\@annot@start@print}%
3897 {}%
3898 \egroup%
3899 %

```

Then print the dash + end line number, or the range symbol.

```

3900 \ifdimequal{\csuse{Xboxendlinenum@\@currentseries}}{0pt}%
3901 {\bgroup}%
3902 {\hbox to \csuse{Xboxendlinenum@\@currentseries}\bgroup}%
3903 \ifl@d@xtwolines%
3904 \ifl@d@xmorethantwolines%
3905 \csuse{Xmorethantwolines@\@currentseries}%
3906 \else%
3907 \csuse{Xtwolines@\@currentseries}%
3908 \fi%
3909 \else%
3910 \ifl@d@dash%
3911 \ifdefined\linerangesep%
3912 \linerangesep%
3913 \else%
3914 \csuse{Xlinerangeseparator@\@currentseries}%
3915 \fi%
3916 \fi%
3917 \ifcsstring{Xlinenumannotationposition@\@currentseries}{before}%
3918 {\@annot@end@print}%
3919 {}%
3920 \ifl@d@pnum%
3921 \wrap@edcrossref{\@this@crossref@end}{#4}%
3922 \csuse{Xpagelinesep@\@currentseries}%
3923 \fi%

```

```

3924 \ifl@d@elin%
3925   \wrap@edcrossref{\@this@crossref@end}{%
3926     \linenumrep{#5}%
3927     \iftoggle{Xlineflag@\@currentseries}{#8}{}%
3928   }%
3929 \fi%
3930 \ifl@d@esl%
3931   \ifl@d@elin%
3932     \csuse{Xsublinesep@\@currentseries}%
3933   \fi%
3934   \wrap@edcrossref{\@this@crossref@end}{\sublinenumrep{#6}}%
3935 \fi%
3936 \ifcsstring{Xlinenumannotationposition@\@currentseries}{after}%
3937   {\@annot@end@print}%
3938 {}%
3939 \fi%
3940 \ifdimequal{\csuse{Xboxendlinenum@\@currentseries}}{0pt}%
3941 {}%
3942 {\hfill}%Prevent underfull hbox
3943 \egroup%
3944 \endgroup%
3945 }%
3946 %

```

XIII.7.5 Footnote grouped by line

`\prepare@Xgroupbyline` `\prepare@Xgroupbyline` is a macro called on on the `\metaXXXvfootnote` if `\Xgroupbyline` is set to true, instead of calling directly the `\insert`.

```

3947 \newcommand{\prepare@Xgroupbyline}[3]{%
3948   \iftoggle{Xgroupbylineseparatwolines@#1}{%
3949     \l@dparsespec#2%
3950     \ifcsdef{#1@forinserting@\l@dparseendpage-\l@dparseendline-\l@dparseendsub}%
3951     {%
3952       \csgappto%
3953         {#1@forinserting@\l@dparseendpage-\l@dparseendline-\l@dparseendsub}%
3954       {%
3955         \ifcsemt{Xsymlinenum@#1}%
3956         {\csuse{Xparafootsep@#1}}%
3957         {}%
3958         #3{#1}{#2}%
3959         \hskip\csuse{Xafternote@#1}\relax%
3960       }%
3961     }%
3962   }%
3963   \csdef%
3964     {#1@forinserting@\l@dparseendpage-\l@dparseendline-\l@dparseendsub}%

```

```

3965     {%
3966     #3{#1}{#2}%
3967     \hskip\csuse{Xafternote@#1}\relax%
3968     }%
3969   }%
3970   \listcsxadd{#1@forinserting}{\l@dparsedendpage-\l@dparsedendline-\l@dparsedendsub}%
3971 }{%
3972 \ifcsdef{#1@forinserting@all}{%
3973   \csgappto%
3974   {#1@forinserting@all}}%
3975   {%
3976     \ifcsempy{Xsymlinenum@#1}%
3977     {\csuse{Xparafootsep@#1}}%
3978     {}%
3979     #3{#1}{#2}%
3980     \hskip\csuse{Xafternote@#1}\relax%
3981   }%
3982 }%
3983 {%
3984   \csdef%
3985   {#1@forinserting@all}}%
3986   {%
3987     #3{#1}{#2}%
3988     \hskip\csuse{Xafternote@#1}\relax%
3989   }%
3990 }%
3991 \listcsgadd{#1@forinserting}{all}%
3992 }%
3993 }%
3994 %

```

XIV Familiar footnotes

XIV.1 Adjacent footnotes

The original edmac provided users with five series of critical footnotes (`\Afootnote` `\Bfootnote` `\Cfootnote` `\Dfootnote` `\Efootnote`), and \LaTeX provides a single numbered footnote. The `reledmac` package uses the edmac mechanism to provide six series of numbered footnotes.

First, though, the `footmisc` package has an option whereby two or more consecutive `\footnotes` have their marks separated by commas. This seemed to Peter Wilson such a useful ability that it was provided automatically by `eledmac`.

Maïeul Rouquette has maintained this feature in `reledmac`, despite he thought that is not directly in relationship with the aim of `reledmac`.

`\multiplefootnotemarker` These macros may have been defined by the memoir class, are provided by the `footmisc`
`\multfootsep` package and perhaps by other footnote packages. That is why we use `\providecommand`

and not `\newcommand`.

```
3995 \providecommand*\multiplefootnotemarker}{3sp}
3996 \providecommand*\multfootsep}{\textsuperscript{\normalfont,}}
3997
3998 %
```

`\m@mmf@prepare` A pair of self-cancelling kerns. This may have been defined in the memoir class.

```
3999 \providecommand*\m@mmf@prepare}{%
4000   \kern-\multiplefootnotemarker
4001   \kern\multiplefootnotemarker\relax}
4002 %
```

`\m@mmf@check` This may have been defined in the memoir class. If it recognises the last kern as `\multiplefootnotemarker` it typesets `\multfootsep`.

```
4003 \providecommand*\m@mmf@check}{%
4004   \ifdim\lastkern=\multiplefootnotemarker\relax
4005     \edef\@x@sf{\the\spacefactor}%
4006     \unkern
4007     \multfootsep
4008     \spacefactor\@x@sf\relax
4009   \fi}
4010
4011 %
```

We have to modify `\@footnotetext` and `\@footnotemark`. However, if memoir is used the modifications have already been made.

```
4012 \@ifclassloaded{memoir}{}{%
4013 %
```

`\@footnotetext` Add `\m@mmf@prepare` at the end of `\@footnotetext`.

```
4014 \apptocmd{\@footnotetext}{\m@mmf@prepare}{}{}
4015 %
```

`\@footnotemark` Modify `\@footnotemark` to cater for adjacent footnotes.

```
4016
4017 \patchcmd{\@footnotemark}
4018   {\nobreak}
4019   {\m@mmf@check
4020   \nobreak
4021   }
4022   {}{}
4023 \patchcmd{\@footnotemark}
4024   {\@makefnmark}
```

```

4025 {\@makefnmark
4026 \m@mmf@prepare
4027 }
4028 {}{}
4029 %

```

Finished the modifications for the non-memoir case.

```

4030 }
4031
4032 %

```

XIV.2 Regular footnotes for numbered texts

`\l@doldold@footnotetext` In order to enable the regular `\footnotes` in numbered text we have to play around with its `\@footnotetext`, using different forms for when in numbered or regular text.

```

4033 \pretocmd{\@footnotetext}{%
4034 \ifnumberedpar@
4035 \edtext{}{\l@dbfnote{#1}}}%
4036 \else
4037 }{}{}
4038 \apptocmd{\@footnotetext}{\fi}{}{}%
4039 %

```

`\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
4040
4041 \patchcmd%
4042 {\footnote}%
4043 {\stepcounter\@mpfn}%
4044 {%
4045 \ifboolexpr{bool{l@dpairing} or bool{l@dprintingpages} or bool{
l@dprintingcolumns}}{%
4046 \global\advance\footnote@reading by \@ne%
4047 \get@thisfootnote%
4048 \get@fnmark{\thisc@footnote}%
4049 \ifcsdef{footnotereading\the\footnote@reading=typeset}%
4050 {\setcounter{\@mpfn}{\csuse{footnotereading\the\footnote@reading=
typeset}}}%
4051 {\setcounter{\@mpfn}{\footnote@reading}}%
4052 }{}%
4053 \stepcounter\@mpfn%
4054 }%
4055 %

```

```

4056   {}
4057   {}
4058
4059   \newcommand{\get@thisfootnote}{%
4060     \ifboolexpr{bool{l@dpairing} or bool{l@dprintingpages} or bool{
4061       l@dprintingcolumns}}{%
4062       \protected@xdef\thisc@footnote{\the\footnote@reading}%
4063     }{%
4064       \protected@xdef\thisc@footnote{\the\c@footnote}%
4065     }%
4066
4067   \newcommand{\l@dbfnote}[1]{%
4068     \get@thisfootnote%
4069     \gdef\@tag{#1\relax}%
4070     \ifledRcol%
4071       \xright@appenditem{%
4072         \ifdefined\Hy@footnote@currentHref%
4073           \noexpand\def\noexpand\Hy@footnote@currentHref{\
4074             Hy@footnote@currentHref}%
4075         \fi%
4076         \noexpand\vl@dbfnote{\expandonce\@tag}}{\thisc@footnote}%
4077       }%
4078       \to\inserts@listR
4079       \global\advance\insert@countR \@ne%
4080     \else%
4081       \xright@appenditem{%
4082         \ifdefined\Hy@footnote@currentHref%
4083           \noexpand\def\noexpand\Hy@footnote@currentHref{\
4084             Hy@footnote@currentHref}%
4085         \fi%
4086         \noexpand\vl@dbfnote{\expandonce\@tag}}{\thisc@footnote}%
4087       }%
4088       \to\inserts@list
4089       \global\advance\insert@count \@ne%
4090     \fi
4091     \ignorespaces%
4092   }%
4093
4094   \newcommand{\get@fnmark}[1]{%
4095     \ifboolexpr{bool{l@dpairing} or bool{l@dprintingpages} or bool{
4096       l@dprintingcolumns}}{%
4097       {%
4098         \stepcounter{footnote@typeset}%
4099         \setcounter{footnote}{\c@footnote@typeset}%
4100         \immediate\write\@mainaux{%
4101           \csgdef{footnotereading#1=typeset}{\the\c@footnote@typeset}%
4102         }%
4103         \def\@thefnmark{\thefootnote}%
4104       }%
4105     }%

```

```

4102   {%
4103     \setcounter{footnote}{#1}%
4104     \def\@thefnmark{\thefootnote}%
4105   }%
4106 }%
4107
4108 \newcommand{\vl@dbfnote}[2]{%
4109   \get@fnmark{#2}%
4110   \@footnotetext{#1}%
4111 }%
4112 %

```

XIV.3 Footnote formats

Some of the code for the various formats is remarkably similar to that in section ??.

The following macros generally set things up for the ‘standard’ footnote format.

`\prebodyfootmark` Two convenience macros for use by `\dots@footnotemark\dots` macros.

```

\prebodyfootmark
\postbodyfootmark
4113 \newcommand*{\prebodyfootmark}{%
4114   \leavevmode
4115   \ifhmode
4116     \edef\@x@sf{\the\spacefactor}%
4117     \m@mzf@check
4118     \nobreak
4119     \fi}
4120 \newcommand{\postbodyfootmark}{%
4121   \m@mzf@prepare
4122   \ifhmode\spacefactor\@x@sf\fi\relax}
4123
4124 %

```

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.

```

4125 \newcommandx{\arrangementX}[2][1,usedefault]{%
4126   \def\do##1{%
4127     \csname arrangementX@#2\endcsname{##1}%
4128   }%
4129   \ifstrempy{#1}%
4130     {%
4131       \dolistloop{\@series}%
4132     }%
4133   {
4134     \docsvlist{#1}%

```

```

4135 }%
4136 }%
4137 %

```

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.

```

4138 \newcommand*{\normal@footnotemarkX}[1]{%
4139   \prebodyfootmark
4140   \wrapped@bodyfootmarkX{#1}%
4141   \postbodyfootmark}
4142
4143 %

```

`\normalbodyfootmarkX` The `\normalbodyfootmarkX{<series>}` *really* typesets the in-text marker. The style is the normal superscript.

```

4144 \newcommand*{\normalbodyfootmarkX}[1]{%
4145   \hbox{\textsuperscript{\normalfont\@nameuse{@thefnmark#1}}}}
4146 %

```

`\normalvfootnoteX` `\normalvfootnoteX{<series>}{<text>}` does the `\insert` for the `<series>` and calls the series' `\footfmt . . .` to format the `<text>`.

```

4147 \notbool{parapparatus@}{\newcommand*}{\newcommand*}{\normalvfootnoteX}[2]{%
4148   \beforeinsertionX{#1}%
4149   \insert\@nameuse{footins#1}\bgroup
4150     \reset@font%
4151     \hsize=\expandafter\dimexpr\csuse{widthX@#1}\relax%
4152     \noindent\csuse{bhooknoteX@#1}%
4153     \csuse{notefontsizeX@#1}%
4154     \footsplitskips
4155     \ifl@dpairing\ifl@dpageing\else%
4156       \setnoteswidthliketwocolumnsX@{#1}%
4157     \fi\fi%
4158     \setnotesXpositionliketwocolumns@{#1}%
4159     \spaceskip=\z@skip \xspaceskip=\z@skip
4160     \csuse{\csuse{footnote@dir}}\@nameuse{footfmt#1}{#1}{#2}\egroup}
4161
4162 %

```

`\mpnormalvfootnoteX` The minipage version.

```

4163 \newcommand*{\mpnormalvfootnoteX}[2]{%
4164   \get@thisfootnoteX{#1}%
4165   \get@fnmarkX{#1}{\thisc@footnote}%
4166   \edef\thisc@footnoteX@reading{\the\csname footnote#1@reading\endcsname}%
4167   \global\setbox\@nameuse{mpfootins#1}\vbox{%

```



```

4168 \unvbox\@nameuse{mpfootins#1}
4169 \noindent\csuse{bhooknoteX@#1}%
4170 \csuse{notefontsizeX@#1}%
4171 \hsize\columnwidth
4172 \@parboxrestore
4173 \color@begingroup
4174 \@nameuse{footfmt#1}{#1}{#2}\color@endgroup}}
4175
4176 %

```

`\normalfootfmtX` `\normalfootfmtX{<series>}{<text>}` typesets the footnote text, prepended by the marker.

```

4177 \notbool{parapparatus@}{\newcommand*{\newcommand}{\normalfootfmtX}[2]{%
4178 \ifluatex%
4179 \textdir\footnote@luatextextdir%
4180 \pardir\footnote@luatexpardir%
4181 \par%
4182 \fi%
4183 \protected@edef\@currentlabel{%
4184 \@nameuse{@thefnmark#1}%
4185 }%
4186 \ledsetnormalparstuffX{#1}%
4187 \hangindent=\csuse{hangindentX@#1}%
4188 \everypar{\hangindent=\csuse{hangindentX@#1}}%
4189 \rule\z@\splittopskip%
4190 {\csuse{notenunfontX@#1}\wrapped@footfootmarkX{#1}}%
4191 \csuse{wrapcontentX@#1}{#2}%
4192 \strut\par}}
4193
4194 %

```

`\normalfootfootmarkX` `\normalfootfootmarkX{<series>}` is called by `\normalfootfmtX` to typeset the footnote marker in the footer before the footnote text.

```

4195 \newcommand*{\normalfootfootmarkX}[1]{%
4196 \textsuperscript{\@nameuse{@thefnmark#1}}}
4197
4198 %

```

`\normalfootstartX` `\normalfootstartX{<series>}` is the `<series>` footnote starting macro used in the output routine.

```

4199 \newcommand*{\normalfootstartX}[1]{%
4200 \ifdimequal{Opt}{\prenotesX@}{}%
4201 {%
4202 \iftoggle{prenotesX@}{%
4203 \togglefalse{prenotesX@}%
4204 \skip\csname footins#1\endcsname=%
4205 \glueexpr\csuse{prenotesX@}+\csuse{afterruleX@#1}\relax%
4206 }%

```

```

4207     }%
4208   }%
4209   \vskip\skip\csname footins#1\endcsname%
4210   \leftskip=\z@
4211   \rightskip=\z@
4212   \ifl@dpairing\else%
4213     \hsize=\old@hsize%
4214   \fi%
4215   \setnoteswidthliketwocolumnsX@{#1}%
4216   \setnotesXpositionliketwocolumns@{#1}%
4217   \print@footnoteXrule{#1}%
4218 }%
4219
4220 %

```

`\normalfootnoteruleX` The rule drawn before the footnote series group.

```

4221 \let\normalfootnoteruleX=\footnoterule
4222
4223 %

```

`\normalfootgroupX` `\normalfootgroupX{<series>}` sends the contents of the `<series>` insert box to the output page without alteration.

```

4224 \newcommand*{\normalfootgroupX}[1]{%
4225   \csuse{bhookgroupX@#1}%
4226   \unvbox@nameuse{footins#1}%
4227   \hsize=\old@hsize%
4228 }%
4229
4230 %

```

`\mpnormalfootgroupX` The minipage version.

```

4231 \newcommand*{\mpnormalfootgroupX}[1]{%
4232   \vskip\skip@nameuse{mpfootins#1}
4233   \ifl@dpairing\ifparledgroup%
4234     \leavevmode\marks\parledgroup@{begin}%
4235     \marks\parledgroup@series{#1}%
4236     \marks\parledgroup@type{footnoteX}%
4237   \fi\fi\normalcolor
4238   \ifparledgroup%
4239     \ifl@dpairing%
4240     \else%
4241       \setnoteswidthliketwocolumnsX@{#1}%
4242       \setnotesXpositionliketwocolumns@{#1}%
4243       \print@footnoteXrule{#1}%
4244     \fi%
4245   \else%
4246     \setnoteswidthliketwocolumnsX@{#1}%

```

```

4247 \setnotesXpositionliketwocolumns@{#1}%
4248 \print@footnoteXrule{#1}%
4249 \fi%
4250 \csuse{bhookgroupX@#1}%
4251 \unvbox\@nameuse{mpfootins#1}}
4252
4253 %

```

\normalbfnoteX₅₄

```

4255 \newcommand{\normalbfnoteX}[2]{%
4256 \get@thisfootnoteX{#1}%
4257 \ifledRcol%
4258 \ifluatex
4259 \footnotelang@lua[R]%
4260 \fi
4261 \@ifundefined{xpg@main@language}%if polyglossia
4262 {}%
4263 {\footnotelang@poly[R]}%
4264 \xright@appenditem{%
4265 \noexpand\led@set@index@fornote{#1}%
4266 \noexpand\prepare@edindex@fornote{\l@d@nums}%
4267 \unexpanded{\def\this@footnoteX@reading}{\the\csname footnote#1
@reading\endcsname}%
4268 \noexpand\vbfnoteX{#1}{#2}{\this@footnote}%
4269 \noexpand\led@reinit@index@fornote%
4270 \unexpanded{\advance\@edindex@fornote@m@ne}%
4271 }%
4272 \to\inserts@listR
4273 \global\advance\insert@countR \@ne%
4274 \else%
4275 \ifluatex
4276 \footnotelang@lua%
4277 \fi
4278 \@ifundefined{xpg@main@language}%if polyglossia
4279 {}%
4280 {\footnotelang@poly}%
4281 \xright@appenditem{%
4282 \noexpand\led@set@index@fornote{#1}%
4283 \noexpand\prepare@edindex@fornote{\l@d@nums}%
4284 \unexpanded{\def\this@footnoteX@reading}{\the\csname footnote#1
@reading\endcsname}%
4285 \noexpand\vbfnoteX{#1}{#2}{\this@footnote}%
4286 \noexpand\led@reinit@index@fornote%
4287 \unexpanded{\advance\@edindex@fornote@m@ne}%
4288 }%
4289 \to\inserts@list
4290 \global\advance\insert@count \@ne%
4291 \fi
4292 \ignorespaces}

```

```
4293
4294 %
```

`\get@thisfootnoteX` The macro `\get@thisfootnote` command just saves the footnote number in the `\thisfootnote` macro, depending on the use of pairing environments.

```
4295 \newcommand{\get@thisfootnoteX}[1]{%
4296   \ifboolexpr{bool{l@dpairing} or bool{l@dprintingpages} or bool{
l@dprintingcolumns}}{%
4297     \protected@xdef\thisc@footnote{\the\csname footnote#1@reading\
endcsname}%
4298   }{%
4299     \protected@xdef\thisc@footnote{\the\csname c@footnote#1\endcsname}%
4300   }%
4301 }%
4302 %
```

`\vbfnoteX` This command calls the correct footnote-inserting commands.

```
4303 \newcommand{\vbfnoteX}[3]{%
4304   \get@fnmarkX{#1}{#3}%
4305   \@nameuse{regvfootnote#1}{#1}{#2}%
4306 }%
4307
4308 %
```

`\get@fnmarkX` This command gets the correct footnote number when typesetting parallel texts.

```
4309 \newcommand{\get@fnmarkX}[2]{%
4310   \ifboolexpr{bool{l@dpairing} or bool{l@dprintingpages} or bool{
l@dprintingcolumns}}{%
4311     {%
4312       \stepcounter{footnote#1@typeset}%
4313       \setcounter{footnote#1}{\value{footnote#1@typeset}}%
4314       \@namedef{@thefnmark#1}{\csuse{thefootnote#1}}%
4315       \immediate\write\@mainaux{%
4316         \csgdef{footnote#1reading#2=typeset}{\the\csname c@footnote#1
@typeset\endcsname}%
4317       }%
4318     }%
4319     {%
4320       \setcounter{footnote#1}{#2}%
4321       \@namedef{@thefnmark#1}{\csuse{thefootnote#1}}%
4322     }%
4323   }
4324 %
4325 %
```

```

\vnunfootnoteX26 \newcommand{\vnunfootnoteX}[2]{%
4327 \ifnumberedpar@
4328 \edtext{}{\normalbfnoteX{#1}{#2}}%
4329 \else
4330 \def\this@footnoteX@reading{\the\csname footnote#1@reading\endcsname}%
4331 \get@thisfootnoteX{#1}%
4332 \get@fnmarkX{#1}{\expandonce\this@footnote}%
4333 \@nameuse{regvfootnote#1}{#1}{#2}%
4334 \fi}
4335
4336 %

```

`arrangementX@normal` `\arrangementX@normal{<series>}` initialises the settings for the `<series>` footnotes. This should always be called for each series.

```

4337 \newcommand*{\arrangementX@normal}[1]{%
4338 \csgdef{series@displayX#1}{normal}
4339 \expandafter\let\csname footstart#1\endcsname=\normalfootstartX
4340 \@namedef{@footnotemark#1}{\normal@footnotemarkX{#1}}
4341 \@namedef{bodyfootmark#1}{\normalbodyfootmarkX{#1}}
4342 \expandafter\let\csname regvfootnote#1\endcsname=\normalvfootnoteX
4343 \expandafter\let\csname vfootnote#1\endcsname=\vnunfootnoteX
4344 \expandafter\let\csname footfmt#1\endcsname=\normalfootfmtX
4345 \@namedef{footfootmark#1}{\normalfootfootmarkX{#1}}
4346 \expandafter\let\csname footgroup#1\endcsname=\normalfootgroupX
4347 \expandafter\let\csname footnoterule#1\endcsname=\normalfootnoteruleX
4348 \count\csname footins#1\endcsname=1000
4349 \csxdef{default@footins#1}{1000}%Use to have note only for one side
4350 \dimen\csname footins#1\endcsname=\csuse{maxhnotesX@#1}
4351 \skip\csname footins#1\endcsname=\csuse{beforenotesX@#1}%
4352 \advance\skip\csname footins#1\endcsname by\csuse{afterruleX@#1}%
4353 %

```

Additions for minipages.

```

4354 \ifnoledgroup@ \else%
4355 \expandafter\let\csname mpvfootnote#1\endcsname=\mpnormalvfootnoteX
4356 \expandafter\let\csname mpfootgroup#1\endcsname=\mpnormalfootgroupX
4357 \count\csname mpfootins#1\endcsname=1000
4358 \dimen\csname mpfootins#1\endcsname=\csuse{maxhnotesX@#1}
4359 \skip\csname mpfootins#1\endcsname=\csuse{beforenotesX@#1}%
4360 \advance\skip\csname mpfootins#1\endcsname by\csuse{afterruleX@#1}%
4361 \fi
4362 }
4363
4364 %

```

XIV.4.3 Two columns footnotes

The following macros set footnotes in two columns. It is assumed that the length of each footnote is less than the column width.

```

\arrangementX@twocol  \newcommand*\arrangementX@twocol}[1]{%
4366   \csgdef{series@displayX#1}{twocol}
4367   \expandafter\let\csname regvfootnote#1\endcsname=\twocolvfootnoteX
4368   \expandafter\let\csname footfmt#1\endcsname=\twocolfootfmtX
4369   \expandafter\let\csname footgroup#1\endcsname=\twocolfootgroupX
4370   \dimen\csname footins#1\endcsname=\csuse{maxhnotesX@#1}%
4371   \skip\csname footins#1\endcsname=\csuse{beforenotesX@#1}%
4372   \advance\skip\csname footins#1\endcsname by \csuse{afterruleX@#1}\relax%
4373   \twocolfootsetupX{#1}
4374   \ifnoledgroup@else%
4375     \expandafter\let\csname mpvfootnote#1\endcsname=\mpnormalvfootnoteX
4376     \expandafter\let\csname mpfootgroup#1\endcsname=\mptwocolfootgroupX
4377     \skip\csname mpfootins#1\endcsname=\csuse{beforenotesX@#1}%
4378     \advance\skip\csname mpfootins#1\endcsname by \csuse{afterruleX@#1}
4379     \mptwocolfootsetupX{#1}
4380   \fi%
4381 }
4382
4383 %

```

```

\twocolfootsetupX  \twocolfootsetupX{<series>}
\mptwocolfootsetupX
4384 \newcommand*\twocolfootsetupX}[1]{%
4385   \count\csname footins#1\endcsname 500
4386   \csxdef{default@footins#1}{500}%Use this to confine the notes to one
side only
4387   \multiply\dimen\csname footins#1\endcsname by \tw@}
4388   \newcommand*\mptwocolfootsetupX}[1]{%
4389     \count\csname mpfootins#1\endcsname 500
4390     \multiply\dimen\csname mpfootins#1\endcsname by \tw@}
4391
4392 %

```

```

\twocolvfootnoteX  \twocolvfootnoteX{<series>}
4393 \notbool{parapparatus@}\newcommand*\newcommand*\twocolvfootnoteX}[2]{%
4394   \beforeinsertion@X{#1}%
4395   \insert\csname footins#1\endcsname\bgroup%
4396     \hsize=\expandafter\dimexpr\csuse{widthX@#1}\relax%
4397     \noindent\csuse{bhooknoteX@#1}%
4398     \csuse{notefontsizeX@#1}%
4399     \footsplitskips%
4400     \spaceskip=\z@skip \xspaceskip=\z@skip%
4401     \@nameuse{footfmt#1}{#1}{#2}\egroup}
4402
4403 %

```

```

\twocolfootfmtX  \twocolfootfmtX{<series>}

```

```

4404 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\twocolfootfmtX}[2]{%
4405   \protected@edef\@currentlabel{%
4406     \@nameuse{@thefnmark#1}%
4407   }%
4408   \normal@pars%
4409   \hangindent=\csuse{hangindentX@#1}%
4410   \everypar{\hangindent=\csuse{hangindentX@#1}}%
4411   \hspace \csuse{hsizetwocolX@#1}%
4412   \nottoggle{parindentX@#1}{\parindent=\z@}{}%
4413   \tolerance=5000\relax%
4414   \par%
4415   \@tempdima=\parindent%
4416   \csuse{colalignX@#1}%
4417   \parindent=\@tempdima%
4418   {\hspace{\parindent}}%
4419   \csuse{notenumentX@#1}\wrapped@footfootmarkX{#1}\strut%
4420   \csuse{wrapcontentX@#1}{#2}%
4421   \strut\par}%
4422 \allowbreak%
4423 }%
4424
4425 %

```

```

\twocolfootgroupX \twocolfootgroupX{series}
\mptwocolfootgroupX
4426 \newcommand*{\twocolfootgroupX}[1]{\csuse{hookgroupX@#1}\csuse{
notefontsizeX@#1}
4427   \splittopskip=\ht\strutbox
4428   \expandafter
4429   \rigidbalanceX\csname footins#1\endcsname \tw@ \splittopskip}}
4430
4431 \newcommand*{\mptwocolfootgroupX}[1]{%
4432   \vskip\skip\@nameuse{mpfootins#1}
4433   \ifl@dpairing\ifparledgroup%
4434     \leavevmode\marks\parledgroup@{begin}%
4435     \marks\parledgroup@series{#1}%
4436     \marks\parledgroup@type{footnoteX}%
4437   \fi\fi\normalcolor
4438   \ifparledgroup%
4439     \ifl@dpairing%
4440     \else%
4441       \setnoteswidthliketwocolumnsX@{#1}%
4442       \setnotesXpositionliketwocolumns@{#1}%
4443       \print@footnoteXrule{#1}%
4444     \fi%
4445   \else%
4446     \setnoteswidthliketwocolumnsX@{#1}%
4447     \setnotesXpositionliketwocolumns@{#1}%
4448     \print@footnoteXrule{#1}%
4449   \fi%

```

```

4450 \cuse{bhookgroupX@#1}%
4451 \splittopskip=\ht\strutbox
4452 \expandafter
4453 \rigidbalanceX\csname mpfootins#1\endcsname \tw@ \splittopskip}}
4454
4455 %

```

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@threecol56 \newcommand*\arrangementX@threecol}[1]{%
4457 \csgdef{series@displayX#1}{threecol}
4458 \expandafter\let\csname regvfootnote#1\endcsname=\threecolvfootnoteX
4459 \expandafter\let\csname footfmt#1\endcsname=\threecolfootfmtX
4460 \expandafter\let\csname footgroup#1\endcsname=\threecolfootgroupX
4461 \dimen\csname footins#1\endcsname=\cuse{maxhnotesX@#1}%
4462 \skip\csname footins#1\endcsname=\cuse{beforenotesX@#1}%
4463 \advance\skip\csname footins#1\endcsname by \cuse{afterruleX@#1}\relax%
4464 \threecolfootsetupX{#1}
4465 \ifnoledgroup@\else%
4466 \expandafter\let\csname mpvfootnote#1\endcsname=\mpnormalvfootnoteX
4467 \expandafter\let\csname mpfootgroup#1\endcsname=\mpthreecolfootgroupX
4468 \skip\csname mpfootins#1\endcsname=\cuse{beforenotesX@#1}%
4469 \advance\skip\csname mpfootins#1\endcsname by \cuse{afterruleX@#1}
4470 \mpthreecolfootsetupX{#1}
4471 \fi%
4472 }
4473
4474 %

```

```

\threecolfootsetupX \threecolfootsetupX{<series>}
\mpthreecolfootsetupX
4475 \newcommand*\threecolfootsetupX}[1]{%
4476 \count\csname footins#1\endcsname 333
4477 \csxdef{default@footins#1}{333}%Use this to confine the notes to one
side only
4478 \multiply\dimen\csname footins#1\endcsname by \thr@@}
4479 \newcommand*\mpthreecolfootsetupX}[1]{%
4480 \count\csname mpfootins#1\endcsname 333
4481 \multiply\dimen\csname mpfootins#1\endcsname by \thr@@}
4482
4483 %

```

```

\threecolvfootnoteX \threecolvfootnoteX{<series>}{<text>}
4484 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\threecolvfootnoteX}[2]{
%

```



```

4485 \beforeinsertionX{#1}%
4486 \insert\csname footins#1\endcsname\bgroup%
4487   \hsize=\expandafter\dimexpr\csuse{widthX@#1}\relax%
4488   \noindent\csuse{bhooknoteX@#1}%
4489   \csuse{notefontsizeX@#1}%
4490   \footsplitskips%
4491   \@nameuse{footfmt#1}{#1}{#2}\egroup}
4492
4493 %

```

`\threecolfootfmtX` `\threecolfootfmtX{<series>}`

```

4494 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\threecolfootfmtX}[2]{%
4495   \protected@edef\@currentlabel{%
4496     \@nameuse{@thefnmark#1}%
4497   }%
4498   \hangindent=\csuse{hangindentX@#1}%
4499   \everypar{\hangindent=\csuse{hangindentX@#1}}%
4500   \normal@pars%
4501   \hsize \csuse{hsizethreecolX@#1}%
4502   \nottoggle{parindentX@#1}{\parindent=\z@}{}%
4503   \tolerance=5000\relax%
4504   \@tempdima=\parindent%
4505   \csuse{colalignX@#1}%
4506   \parindent=\@tempdima%
4507   {\hspace{\parindent}}%
4508   \csuse{notenumentX@#1}\wrapped@footfootmarkX{#1}\strut%
4509   \csuse{wrapcontentX@#1}{#2}%
4510   \strut\par}\allowbreak}
4511
4512 %

```

`\threecolfootgroupX` `\threecolfootgroupX{<series>}`
`\mpthreecolfootgroupX`

```

4513 \newcommand*{\threecolfootgroupX}[1]{\csuse{bhookgroupX@#1}\csuse{
notefontsizeX@#1}
4514   \splittopskip=\ht\strutbox
4515   \expandafter
4516   \rigidbalanceX\csname footins#1\endcsname \thr@@ \splittopskip}}
4517
4518 \newcommand*{\mpthreecolfootgroupX}[1]{%
4519   \vskip\skip\@nameuse{mpfootins#1}
4520   \ifl@dpairing\ifparledgroup
4521     \leavevmode\marks\parledgroup@{begin}%
4522     \marks\parledgroup@series{#1}%
4523     \marks\parledgroup@type{footnoteX}%
4524     \fi\fi\normalcolor
4525   \ifparledgroup%
4526     \ifl@dpairing%
4527     \else%

```

```

4528 \setnoteswidthliketwocolumnsX@{#1}%
4529 \setnotesXpositionliketwocolumns@{#1}%
4530 \print@footnoteXrule{#1}%
4531 \fi%
4532 \else%
4533 \setnoteswidthliketwocolumnsX@{#1}%
4534 \setnotesXpositionliketwocolumns@{#1}%
4535 \print@footnoteXrule{#1}%
4536 \fi%
4537 \csuse{bhookgroupX@#1}%
4538 \splittopskip=\ht\strutbox
4539 \expandafter
4540 \rigidbalanceX\csname mpfootins#1\endcsname \thr@@ \splittopskip}}
4541
4542 %

```

XIV.4.5 Paraphed footnotes

The following macros set footnotes as one paragraph.

`\arrangementX@threecol` `\footparagraphX{<series>}`

```

4543 \newcommand*{\arrangementX@paragraph}[1]{%
4544 \csgdef{series@displayX#1}{paragraph}%
4545 \expandafter\let\csname footstart#1\endcsname=\parafootstartX
4546 \expandafter\let\csname regvfootnote#1\endcsname=\para@vfootnoteX
4547 \expandafter\let\csname footfmt#1\endcsname=\parafootfmtX
4548 \expandafter\let\csname footgroup#1\endcsname=\para@footgroupX
4549 \expandafter\let\csname footnoterule#1\endcsname=\normalfootnoteruleX
4550 \count\csname footins#1\endcsname=1000
4551 \csxdef{default@footins#1}{1000}%Use this to confine the notes to one
side only
4552 \dimen\csname footins#1\endcsname=\csuse{maxhnotesX@#1}
4553 \skip\csname footins#1\endcsname=\csuse{beforenotesX@#1}%
4554 \advance\skip\csname footins#1\endcsname by\csuse{afterruleX@#1}%
4555 \para@footsetupX{#1}
4556 \ifnoledgroup@else
4557 \expandafter\let\csname mpvfootnote#1\endcsname=\mppara@vfootnoteX
4558 \expandafter\let\csname mpfootgroup#1\endcsname=\mppara@footgroupX
4559 \count\csname mpfootins#1\endcsname=1000
4560 \dimen\csname mpfootins#1\endcsname=\csuse{maxhnotesX@#1}
4561 \skip\csname mpfootins#1\endcsname=\csuse{beforenotesX@#1}%
4562 \advance\skip\csname mpfootins#1\endcsname by\csuse{afterruleX@#1}%
4563 \fi
4564 }
4565
4566 %

```

`\para@footsetupX` `\para@footsetupX{<series>}`

```

4567 \newcommand*{\para@footsetupX}[1]{\csuse{bhookgroupX@#1}\csuse{
notefontsizeX@#1}
4568 \setnoteswidthliketwocolumnsX@{#1}%
4569 \ifcsemtyp{widthX@#1}%
4570 {}%
4571 {\columnwidth=\expandafter\dimexpr\csuse{widthX@#1}\relax}%
4572 \dimen0=\baselineskip
4573 \multiply\dimen0 by 1024
4574 \divide\dimen0 by \columnwidth \multiply\dimen0 by \footfudgefiddle\relax
%
4575 \expandafter
4576 \xdef\csname footfudgefactor#1\endcsname{%
4577 \expandafter\strip@pt\dimen0 }}}
4578
4579 %

```

`\parafootstartX` `\parafootstartX{<series>}`

```

4580 \newcommand*{\parafootstartX}[1]{%
4581 \ifdimequal{0pt}{\prenotesX@}{}%
4582 {}%
4583 \iftoggle{prenotesX@}{%
4584 \togglefalse{prenotesX@}%
4585 \skip\csname footins#1\endcsname=%
4586 \glueexpr\csuse{prenotesX@}+\csuse{afterruleX@#1}\relax%
4587 }%
4588 {}%
4589 }%
4590 \leftskip=\z@
4591 \rightskip=\z@
4592 \nottoggle{parindentX@#1}{\parindent=\z@}{}%
4593 \vskip\skip\@nameuse{footins#1}%
4594 \setnoteswidthliketwocolumnsX@{#1}%
4595 \setnotesXpositionliketwocolumns@{#1}%
4596 \print@footnoteXrule{#1}%
4597 }
4598
4599 %

```

`\para@vfootnoteX` `\para@vfootnoteX{<series>}{<text>}`
`\mppara@vfootnoteX`

```

4600 \newcommand*{\para@vfootnoteX}[2]{%
4601 \csuse{beforeinsertingX@#1}%
4602 \insert\csname footins#1\endcsname%
4603 \bgroup
4604 \csuse{notefontsizeX@#1}
4605 \footsplitskips
4606 \setbox0=\vbox{\hsize=\maxdimen%
4607 \let\bidirectional@everypar\@empty%
4608 \insert@txtbeforenotesX@#1}%

```

```

4609     \noindent\csuse{bhooknoteX@#1}%
4610     \@nameuse{footfmt#1}{#1}{#2}}%
4611     \setbox0=\hbox{\unvxhX{0}{#1}}%
4612     \dp0=\z@
4613     \ht0=\csname footfudgefactor#1\endcsname\wd0
4614     \box0
4615     \penalty0
4616     \egroup}
4617 \newcommand*\mppara@vfootnoteX}[2]{%
4618     \get@thisfootnoteX{#1}%
4619     \get@fnmarkX{#1}{\thisc@footnote}%
4620     \edef\this@footnoteX@reading{\the\csname footnote#1@reading\endcsname}%
4621     \global\setbox\@nameuse{mpfootins#1}\vbox{%
4622         \unvbox\@nameuse{mpfootins#1}
4623         \csuse{notefontsizeX@#1}
4624         \footsplitskips
4625         \setbox0=\vbox{\hsize=\maxdimen%
4626             \let\bidir@RTL@everypar\@empty%
4627             \noindent\color@begingroup%
4628             \csuse{bhooknoteX@#1}%
4629             \@nameuse{footfmt#1}{#1}{#2}\color@endgroup}%
4630         \setbox0=\hbox{\unvxhX{0}{#1}}%
4631         \dp0=\z@
4632         \ht0=\csname footfudgefactor#1\endcsname\wd0
4633         \box0
4634         \penalty0}}
4635
4636 %

```

```

\unvxhX#1 \newcommand*\unvxhX}[2]{% 2th is optional for retro-compatibility
4638     \setbox0=\vbox{\unvbox#1%
4639         \global\setbox1=\lastbox}%
4640     \unhbox1
4641     \unskip           % remove \rightskip,
4642     \unskip           % remove \parfillskip,
4643     \unpenalty        % remove \penalty of 10000,
4644     \hskip\csuse{afternoteX@#2}%
4645     \relax}% but add the glue to go between the notes
4646
4647 %

```

\parafootfmtX \parafootfmtX{<series>}

```

4648 \newcommand*\parafootfmtX}[2]{%
4649     \protected@edef\@currentlabel{%
4650         \@nameuse{@thefnmark#1}%
4651     }%
4652     \insertparafootsepX{#1}%
4653     \ledsetnormalparstuff@common%

```

```

4654 {\csuse{notenumfontX@#1}%
4655 \csuse{notenumfontX@#1}%
4656 \wrapped@footfootmarkX{#1}%
4657 \strut%
4658 \csuse{wrapcontentX@#1}{#2}%
4659 \penalty-10}}
4660
4661 %

```

`\para@footgroupX` `\para@footgroupX{<series>}`
`\mppara@footgroupX`

```

4662 \newcommand*{\para@footgroupX}[1]{%
4663 \hspace=\expandafter\dimexpr\csuse{widthX@#1}\relax%
4664 \unvbox\csname footins#1\endcsname
4665 \ifcsstring{raggedX@#1}{L}{\RaggedLeft}{}%
4666 \ifcsstring{raggedX@#1}{R}{\RaggedRight}{}%
4667 \makehboxofhboxes
4668 \setbox0=\hbox{\unhbox0 \removehboxes}%
4669 \csuse{bhookgroupX@#1}
4670 \csuse{notefontsizeX@#1}
4671 \unhbox0\par}
4672
4673 \newcommand*{\mppara@footgroupX}[1]{%
4674 \setnoteswidthliketwocolumnsX@{#1}%
4675 \vskip\skip\@nameuse{mpfootins#1}
4676 \ifl@dpairing\ifparledgroup
4677 \leavevmode%
4678 \leavevmode\marks\parledgroup@{begin}%
4679 \marks\parledgroup@series{#1}%
4680 \marks\parledgroup@type{footnoteX}%
4681 \fi\fi\normalcolor
4682 \ifparledgroup%
4683 \ifl@dpairing%
4684 \else%
4685 \setnoteswidthliketwocolumnsX@{#1}%
4686 \setnotesXpositionliketwocolumns@{#1}%
4687 \print@footnoteXrule{#1}%
4688 \fi%
4689 \else%
4690 \setnoteswidthliketwocolumnsX@{#1}%
4691 \setnotesXpositionliketwocolumns@{#1}%
4692 \print@footnoteXrule{#1}%
4693 \fi%
4694 \unvbox\csname mpfootins#1\endcsname
4695 \ifcsstring{raggedX@#1}{L}{\RaggedLeft}{}%
4696 \ifcsstring{raggedX@#1}{R}{\RaggedRight}{}%
4697 \makehboxofhboxes
4698 \setbox0=\hbox{\unhbox0 \removehboxes}%
4699 \csuse{bhookgroupX@#1}
4700 \csuse{notefontsizeX@#1}%

```

```

4701 \nottoggle{parindentX@#1}{\parindent=\z@}{}%
4702 \unhbox0\par}}
4703
4704 %

```

Insertion of the footnotes separator The command `\insertparafootsepX{<series>}` must be called at the beginning of `\parafootftmX`.

```

\insertparafootsepX \newcommand{\insertparafootsepX}[1]{%
4706   \ifledRcol@%
4707     \ifnumequal{\csuse{prevpage#1@numR}}{\page@numR}%
4708       {\csuse{Xparafootsep@#1}}%
4709       {}%
4710     \global\csname prevpage#1@numR\endcsname=\page@numR%
4711   \else%
4712     \ifnumequal{\csuse{prevpage#1@num}}{\page@num}%
4713       {\csuse{Xparafootsep@#1}}%
4714       {}%
4715     \global\csname prevpage#1@num\endcsname=\page@num%
4716   \fi%
4717 }
4718 %

```

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.

```

4719 \newcommand{\wrapped@footfootmarkX}[1]{%
4720   \ifdefined\hypertarget%
4721     \hyperlink%
4722       {@bodyfootmark#1@this@footnoteX@reading}%
4723       {@nameuse{footfootmark#1}}%
4724     \Hy@raisedlink{%
4725       \hypertarget%
4726       {@footnotemark#1@this@footnoteX@reading}%
4727       }%
4728     }%
4729   \else%
4730     \@nameuse{footfootmark#1}%
4731   \fi%
4732 }%
4733 %

```

`\wrapped@bodyfootmarkX` `\wrapped@bodyfootmarkX` prints the footnote mark of the text body, wrapped in `hyperref` package's commands, if needed.

```

4734 \newcommand{\wrapped@bodyfootmarkX}[1]{%
4735   \ifdefined\hypertarget%
4736     \hyperlink%
4737       {@footnotemark#1\expandafter\the\csname footnote#1@reading\
endcsname}%
4738     {\@nameuse{bodyfootmark#1}}%
4739   \Hy@raisedlink{%
4740     \hypertarget%
4741       {@bodyfootmark#1\expandafter\the\csname footnote#1@reading\
endcsname}%
4742     {}%
4743   }%
4744   \else%
4745     \@nameuse{bodyfootmark#1}%
4746   \fi%
4747 }%
4748 %

```

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_{La}TeX.

That why we have defined \ledsetnormalparstuff@common in order to make this setting for both familiar and critical notes. This command is called by command to make specific setting to critical or familiar footnote.

```

\ledsetnormalparstuff@common 49 \newcommand*{\ledsetnormalparstuff@common}{%
\Xledsetnormalparstuff 50 \ifluatex%
\ledsetnormalparstuffX 51 \ifdefstring{\footnote@luatextextdir}{TLT}{}%
4752   {\textdir\footnote@luatextextdir}%
4753   \pardir\footnote@luatexpardir%
4754   \fi%
4755   \csuse{\csuse{footnote@dir}}%
4756   \normal@pars%
4757   \parfillskip \z@ \@plus 1fil}%
4758
4759 \newcommand*{\Xledsetnormalparstuff}[1]{%
4760   \ledsetnormalparstuff@common%
4761   \nottoggle{Xparindent@#1}{\parindent=\z@}{\hspace{\parindent}}%
4762 }%
4763
4764 \newcommand*{\ledsetnormalparstuffX}[1]{%
4765   \ledsetnormalparstuff@common%

```

```

4766 \nottoggle{parindentX@#1}{\parindent=\z@}{\hspace{\parindent}}}%
4767 }%
4768 %

```

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@`

```

4769
4770 \newdimen\old@hsize%
4771 \AtBeginDocument{\old@hsize=\hsize}%
4772
4773 \newcommand{\setXnoteswidthliketwocolumns@}[1]{%
4774   \global\let\hsize@fornote=\hsize%
4775   \global\old@hsize=\hsize%
4776   \let\old@columnwidth=\columnwidth%
4777   \iftoggle{Xnoteswidthliketwocolumns@#1}%
4778     {%
4779     \setwidthliketwocolumns%
4780     \global\let\hsize@fornote=\hsize%
4781     }%
4782     {}%
4783   \let\hsize=\hsize@fornote%
4784   \let\columnwidth=\old@columnwidth%
4785 }%
4786
4787 \newcommand{\setnoteswidthliketwocolumnsX@}[1]{%
4788   \global\let\hsize@fornote=\hsize%
4789   \global\old@hsize=\hsize%
4790   \let\old@columnwidth=\columnwidth%
4791   \iftoggle{noteswidthliketwocolumnsX@#1}%
4792     {%
4793     \setwidthliketwocolumns%
4794     \global\let\hsize@fornote=\hsize%
4795     }%
4796     {}%
4797   \let\hsize=\hsize@fornote%
4798   \let\columnwidth=\old@columnwidth%
4799 }%
4800
4801 %

```


`\setnotespositionliketwocolumns@` and `\setnotespositionliketwocolumns@X` These two commands set the position of the critical / familiar footnotes, depending on the hooks `Xnoteswidthliketwocolumns` and `noteswidthliketwocolumnsX`. They call commands which are defined only in `reledpar`, because this feature has no sense without `reledpar`.

```

4802 \newcommand{\setXnotespositionliketwocolumns@}[1]{%
4803   \iftoggle{Xnoteswidthliketwocolumns@#1}{%
4804     \csuse{setnotespositionliketwocolumns@\columns@position}%
4805   }{}%
4806 }%
4807
4808 \newcommand{\setnotesXpositionliketwocolumns@}[1]{%
4809   \iftoggle{noteswidthliketwocolumnsX@#1}{%
4810     \csuse{setnotespositionliketwocolumns@\columns@position}%
4811   }{}%
4812 }%
4813
4814 %

```

XVII Footnotes' order

`\fnpos` and `\mpfnpos` The `\fnpos` and `\mpfnpos` simply place their arguments in `\@fnpos` and `\@mpfnpos`, which will be used later in the output routine.

```

4815 \def\@fnpos{familiar-critical}
4816 \def\@mpfnpos{critical-familiar}
4817 \newcommand{\fnpos}[1]{\xdef\@fnpos{#1}}
4818 \newcommand{\mpfnpos}[1]{\xdef\@mpfnpos{#1}}
4819 %

```

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@20 \newcommand{\print@Xfootnoterule}[1]{%
\print@footnoterule@21 \vskip-\csuse{Xafterrule@#1}%Because count in \dimen\csuse{#1footins}
4822 \nointerlineskip%
4823 \moveleft-\leftskip\vbox{\csuse{#1footnoterule}}%
4824 \nointerlineskip%
4825 \vskip\csuse{Xafterrule@#1}%
4826 }%
4827
4828 \newcommand{\print@footnoteruleX}[1]{%
4829 \vskip-\csuse{afterruleX@#1}%Because count in \dimen\csuse{footins#1}
4830 \nointerlineskip%
4831 \moveleft-\leftskip\vbox{\csuse{footnoterule#1}}%

```

```

4832 \nointerlineskip%
4833 \vskip\csuse{afterruleX@#1}%
4834 }%
4835
4836 %

```

XIX Specific skip for first series of footnotes

XIX.0.1 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.0.2 User level command

`\Xprenotes@` If user redefines `\Xprenotes@`, via `\Xprenotes` to a value greater than 0 pt, this skip `\Xprenotes` will be added before first series notes instead of the notes skip.

```

4837 \newtoggle{Xprenotes@}%
4838 \toggletrue{Xprenotes@}%
4839 \newcommand{\Xprenotes@}{Opt}%

```

```

4840 \newcommand*{\Xprenotes}[1]{\renewcommand{\Xprenotes@}{#1}}%
4841 \newcommand{\preXnotes}[1]{\led@warning@preXnotes@deprecated\Xprenotes{#1}}
%For compatibility
4842 %

```

The same, but for familiar footnotes.

```

\Xprenotes@43 \newtoggle{prenotesX@}
\Xprenotes@44 \toggletrue{prenotesX@}
4845 \newcommand{\prenotesX@}{Opt}
4846 \newcommand*{\prenotesX}[1]{\renewcommand{\prenotesX@}{#1}}
4847 %

```

XIX.03 Internal commands

```

firstXseries@48 \gdef\firstXseries@{}
prepare@Xprenotes49 \newcommand{\prepare@Xprenotes}[1]{%
4850 \ifdimequal{Opt}{\Xprenotes@}%
4851 {}%
4852 {}%
4853 \IfStrEq{\firstXseries@}{}{%
4854 \global\skip\csuse{#1footins}=\Xprenotes@%
4855 \global\advance\skip\csname #1footins\endcsname by\csuse{Xafterrule@
#1}%
4856 \gdef\firstXseries@{#1}%
4857 }%
4858 {}%
4859 \ifseriesbefore{#1}{\firstXseries@}%
4860 {}%
4861 \global\skip\csuse{#1footins}=\csuse{Xbeforenotes@\firstXseries@}%
4862 \global\advance\skip\csname #1footins\endcsname by\csuse{Xafterrule@
#1}%
4863 \gdef\firstXseries@{#1}%
4864 }%
4865 {}%
4866 }%
4867 }%
4868 }
4869 %

```

The same thing is required for familiar notes and \prenotesX.

```

firstseriesX@70 \gdef\firstseriesX@{}
prepare@prenotesX71 \newcommand{\prepare@prenotesX}[1]{%
4872 \ifdimequal{Opt}{\prenotesX@}%
4873 {}%
4874 {}%

```

```

4875 \IfStrEq{\firstseriesX@}{}{-%
4876 \global\skip\csuse{footins#1}=\prenotesX@%
4877 \global\advance\skip\csname footins#1\endcsname by\csuse{afterruleX@
#1}%
4878 \gdef\firstseriesX@{#1}%
4879 }%
4880 {%
4881 \ifseriesbefore{#1}{\firstseriesX@}%
4882 f%
4883 \global\skip\csuse{footins#1}=\csuse{beforenotesX@firstseriesX@}%
4884 \global\advance\skip\csname footins#1\endcsname by\csuse{afterruleX@
#1}%
4885 \gdef\firstXseries@{#1}%
4886 }%
4887 {%
4888 }%
4889 }%
4890 }
4891 %

```

XX Endnotes

First, check the noend option.

```

4892 \ifbool{noend@}{}{%Used instead of \ifnoend@ to prevent expansion problem
4893 %

```

XX.1 Internal commands

`\l@dend@open` and `\l@dend@close` are the macros that are used to open and close the endnote file. Note that all our writing to this file is `\immediate`: all page and line numbers for the endnotes are generated by the same mechanism we use for the footnotes, so that there is no need to defer any writing to catch information from the output routine. The argument of these two command is the series letter.

```

4894 \newcommand{\l@dend@open}[1]{-%
4895 \global\booltrue{l@dend@#1}%
4896 \expandafter\immediate%
4897 \expandafter\openout%
4898 \csname l@d@#1end\endcsname%
4899 =\l@auxdir\jobname.#1end\relax%
4900 }%
4901 \newcommand{\l@dend@close}[1]{-%
4902 \global\boolfalse{l@dend@#1}%
4903 \expandafter\immediate%
4904 \expandafter\closeout\csname l@d@#1end\endcsname%
4905 }%
4906
4907 %

```

`\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.

```

4908 \newcommand{\l@dend@stuff}{%
4909   \def\do##1{%
4910     \ifbool{l@dend@##1}{%
4911       {\l@dend@open{##1}}%
4912       \expandafter\immediate\expandafter\write\csname l@d@##1end\endcsname{\
string\l@d@section{\the\section@num}\@percentchar}%
4913     }%
4914     \dolistloop{\@series}%
4915   }%
4916
4917 %

```

`\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.

```

4918 \global\newcommand{\parapparatus@}{\long}\def\endprint#1#2#3#4#5#6#7{ {%
4919   \csuse{Xendhooknote@#4}%
4920   \csuse{Xendnotefontsize@#4}%
4921   \hangindent=\csuse{Xendhangindent@#4}%
4922   \ifXendinsertsep@%
4923     \hskip\csuse{Xendafternote@#4}\relax%
4924     \csuse{Xendsep@#4}%
4925   \else%
4926     \iftoggle{Xendparagraph@#4}%
4927       {\global\Xendinsertsep@true}%
4928     }%
4929   \fi%
4930   \xdef\@currentseries{#4}%
4931   \def\do##1{%

```

```

4932 \setkeys[mac]{truefootnoteoption}{##1}%
4933 }%
4934 \notblank{#5}{\docsvlist{#5}}{}%
4935 \IfStrEq{#6}{R}{\ledRcol@true}{}%
4936 \def\@this@crossref@start{#7:start}%
4937 \def\@this@crossref@end{#7:end}%
4938 \printlineendnote{#1}{#4}%
4939 \IfStrEq{#6}{R}{\ledRcol@false}{}%
4940 \undef\@this@crossref@start%
4941 \undef\@this@crossref@end%
4942 \nottoggle{Xendlemmadisablefontselection@#4}%
4943 {\select@lemmafонт#1|}%
4944 {}%
4945 \bgroup%
4946 \csuse{Xendlemmafонт@#4}%
4947 \csuse{Xendwraplemma@#4}{#2}%
4948 \egroup%
4949 \ifboolexpr{%
4950 togl {nosep@}%
4951 or test{\ifcsemt{Xendlemmaseparator@#4}}%
4952 }%
4953 {\hskip\csuse{Xendinplaceoflemmaseparator@#4}\relax}%
4954 {\nobreak%
4955 \hskip\csuse{Xendbeforelemmaseparator@#4}%
4956 \csuse{Xendlemmaseparator@#4}%
4957 \hskip\csuse{Xendafterlemmaseparator@#4}%
4958 \relax%
4959 }%
4960 \csuse{Xendwrapcontent@#4}{#3}%
4961 \nottoggle{Xendparagraph@#4}{\par}{}%
4962 \def\do##1{%
4963 \setkeys[mac]{falsefootnoteoption}{##1}%
4964 }%
4965 \notblank{#5}{\docsvlist{#5}}{}%
4966 }}%
4967
4968 \let\l@d@section=\@gobble
4969
4970 %

```

\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.

```

4971 \newcommand{\printlineendnote}[2]{%
4972 \l@dp@rsefootspec#1|%
4973 \iftoggle{Xendnumberonlyfirstintwolines@#2}{%
4974 \xdef\lineinfo{\l@dparsedstartpage - \l@dparsedstartline - \
\l@dparsedstartsub - \l@dparsedendpage - \l@dparsedendline - \
\l@dparsedendsub}%

```

```

4975 }%
4976 {%
4977 \xdef\lineinfo{\l@dparsedstartpage - \l@dparsedstartline - \
l@dparsedstartsub}%
4978 }%
4979 \ifboolexpr{%
4980 togl {nonum}%
4981 or togl {Xendnonumber@#2}%
4982 }%
4983 {\hspace{\csuse{Xendinplaceofnumber@#2}}}%
4984 {%
4985 \iftoggle{Xendnumberonlyfirstinline@#2}%
4986 {\ifcsdef{prevendline#2}%
4987 {\ifcsequal{prevendline#2}{\lineinfo@}%
4988 }%
4989 \csuse{Xendbhookinplaceofnumber@#2}%
4990 \ifcsequal{Xendsymmlinenumber@#2}%
4991 {\hspace{\csuse{Xendinplaceofnumber@#2}}}%
4992 {\printsymmlineendnotearea{#2}}}%
4993 \csuse{Xendahookinplaceofnumber@#2}%
4994 }%
4995 {\printlineendnotearea{#1}{#2}}}%
4996 {\printlineendnotearea{#1}{#2}}}%
4997 }%
4998 {\printlineendnotearea{#1}{#2}}%We keep every time line
4999 \csxdef{prevendline#2}{\lineinfo@}%
5000 }%
5001 }%
5002 %

```

```

\printsymmlineendnotearea#3 \newcommand{\printsymmlineendnotearea}[1]{%
5004 \hspace{\csuse{Xendbeforesymmlinenumber@#1}}%
5005 \csuse{Xendnotenumfont@#1}%
5006 \ifdimequal{\csuse{Xendboxsymmlinenumber@#1}}{\z@}%
5007 {\csuse{Xendsymmlinenumber@#1}}%
5008 {\hbox to \csuse{Xendboxsymmlinenumber@#1}%
5009 {\csuse{Xendsymmlinenumber@#1}\hfill}}%
5010 }%
5011 \hspace{\csuse{Xendaftersymmlinenumber@#1}}%
5012 }%
5013 %

```

`\printlineendnotearea` This macro prints the space before the line number, changes the font, then prints the line number and the space after it. It is called by `\endprint` depending of the options about repeating line numbers. The first argument is line information, the second is the notes series (A, B, C, etc.)

```

5014 \newcommand{\printlineendnotearea}[2]{%
5015 \csuse{Xendbhooklinenumber@#2}%

```

```

5016 \hspace{\csuse{Xendbeforenumber@#2}}%
5017 \bgroup%
5018 \csuse{Xendnotenumfont@#2}%
5019 \ifdimequal{\csuse{Xendboxlinenum@#2}}{Opt}%
5020 {\printendlines#1||\ifledRcol@\@Rlineflag\fi}%
5021 {\leavevmode%
5022 \hbox to \csuse{Xendboxlinenum@#2}%
5023 {%
5024 \IfSubStr{RC}{\csuse{Xendboxlinenumalign@#2}}{\hfill}{}%
5025 \printendlines#1||\ifledRcol@\@Rlineflag\fi%
5026 \IfSubStr{LC}{\csuse{Xendboxlinenumalign@#2}}{\hfill}{}%
5027 }}%
5028 \egroup%
5029 \hspace{\csuse{Xendafternumber@#2}}%
5030 \csuse{Xendahooklinenum@#2}%
5031 }%
5032 %

```

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.

```

\toendnotes33 \newcommandx{\toendnotes}[2][1,usedefault]{%
\toendnotes*34 \ifboolexpr{bool{numbering} or bool{numberingR}}{%
5035 \def\do##1{%
5036 \expandafter\immediate\expandafter\write\csname l@d@##1end\endcsname%
5037 {\unexpanded{#2}\@percentchar}%
5038 }%
5039 \ifstrempy{#1}%
5040 {\dolistloop{\@series}}%
5041 {\docsvlist{#1}}%
5042 }{\led@err@toendnotes@outsidenumbering}%
5043 }%
5044 \WithSuffix\newcommandx\toendnotes*[2][1,usedefault]{%
5045 \ifboolexpr{bool{numbering} or bool{numberingR}}{%
5046 \def\do##1{%
5047 \expandafter\immediate\expandafter\write\csname l@d@##1end\endcsname%
5048 {#2\@percentchar}%
5049 }%
5050 \ifstrempy{#1}%
5051 {\dolistloop{\@series}}%
5052 {\docsvlist{#1}}%
5053 }{\led@err@toendnotes@outsidenumbering}%
5054 }%

```



```
5055 %
```

XX.2.2 Printing endnotes

`\doendnotes` `\ifXendinsertsep@` `\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.

```
5056 \newif\ifXendinsertsep@%
5057 \newcommand*\doendnotes}[1]{%
5058   \l@dend@close{#1}%
5059   \begingroup
5060     \csxdef{prevpagenum@#1}{}%
5061     \csxdef{prevpagerange@#1}{}%
5062     \makeatletter
5063     \expandafter\let\csname #1end\endcsname=\endprint
5064     \input\l@auxdir\jobname.#1end%
5065     \global\Xendinsertsep@false%
5066   \endgroup}
5067 %
```

`\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.

```
5068 \newcommand*\doendnotesbysection}[1]{%
5069   \l@dend@close{#1}%
5070   \csxdef{prevpagenum@#1}{}%
5071   \csxdef{prevpagerange@#1}{}%
5072   \global\expandafter\advance\csname #1end@bysection\endcsname by 1%
5073   \begingroup%
5074     \makeatletter%
5075     \def\l@d@section##1{%
5076       \ifnumequal{##1}{\csname #1end@bysection\endcsname}%
5077         {\cslet{#1end}{\endprint}}%
5078         {\cslet{#1end}{\@gobbleseven}}%
5079     }%
5080     \input\l@auxdir\jobname.#1end%
5081     \global\Xendinsertsep@false%
5082   \endgroup%
5083 }%
5084 %
```

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.

```
5085 }%
5086 %
```

`\setprintendlines` The `\printendlines` macro is similar to `\printlines` but is for printing endnotes rather than footnotes.

The principal difference between foot- and endnotes is that footnotes are printed on the page where they are specified but endnotes are printed at a different point in the document. We need an indication of the source of an endnote; `\setprintendlines` provides this by always printing the page number. The coding is slightly simpler than `\setprintlines`.

from the starting page number.

```
5087 \newcommand*\setprintendlines}[6]{%
5088   \let\@annot@start@print\relax%
5089   \let\@annot@end@print\relax%
5090   \l@d@pnumfalse%
5091   \l@d@dashfalse%
5092   \l@d@elinfalse%
5093   %
```

First of all, we print the second page number only if the ending page number is different

```
5094   \ifnum#4=#1 \else
5095     \l@d@pnumtrue
5096     \l@d@dashtrue
5097   \fi
5098   %
```

We print the ending line number if: (1) we are printing the ending page number, or (2) the ending line is different from the starting line number.

```
5099   \ifboolexpr{%
5100     bool{\l@d@pnum}%
5101     or not test{\ifnumequal{#2}{#5}}%
5102   }{%
5103     \l@d@elintrue%
5104     \l@d@dashtrue%
5105     \l@d@elintrue
5106     \l@d@dashtrue
5107     \unless\ifx\relax\@annot@end%
5108     \def\@annot@end@print{%
5109       \l@wrapcs@ifnotempty{Xendwraplinenumannotation@\@currentseries}{\
annot@end}%
5110     }%
5111     \fi%
5112   }%
5113   {%
5114   %
```

We define the starting line number annotation as a merge of the starting annotation and ending annotation if we don't print ending line number. Otherwise, it is only the starting annotation.

```
5115   \ifl@d@elin%
5116   \def\@annot@start@print{%
```

```

5117     \l@wrapcs@ifnotempty{Xendwraplinenumannotation@\@currentseries}{\
annot@start}%
5118     }%
5119     \else%
5120     \ifx\annot@start\annot@end%
5121     \unless\ifx\@annot@start\relax%
5122     \def\@annot@start@print{%
5123     \l@wrapcs@ifnotempty{Xendwraplinenumannotation@\@currentseries}{\
annot@start}%
5124     }%
5125     \fi%
5126     \else%
5127     \ifx\@annot@end@print\relax%
5128     \def\@annot@start@print{%
5129     \l@wrapcs@ifnotempty{Xendwraplinenumannotation@\@currentseries}{%
5130     \ifx\annot@start\empty\else%
5131     \annot@start%
5132     \ifdefined\linerrangesep%
5133     \linerrangesep%
5134     \else%
5135     \csuse{Xendlinerrangeseparator@\@currentseries}%
5136     \fi%
5137     \fi%
5138     \annot@end%
5139     }%
5140     }%
5141     \else%
5142     \let\@annot@start@print\@annot@end@print%
5143     \let\@annot@end@print\relax%
5144     \fi%
5145     \fi%
5146     \fi%
5147     %

```

We print the starting sub-line if it is nonzero.

```

5148     \l@d@ssubfalse
5149     \ifnum#3=0 \else
5150     \l@d@ssubtrue
5151     \fi
5152     %

```

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.

```

5153     \l@d@eslfalse
5154     \ifnum#6=0 \else
5155     \ifnum#6=#3
5156     \ifl@d@elin \l@d@esltrue \else \l@d@eslfalse \fi
5157     \else
5158     \l@d@esltrue
5159     \l@d@dashtrue

```

```

5160     \fi
5161     \fi%
5162 %

5163     \ifl@d@dash%
5164     \ifboolexpr{togl{fulllines@} or test{\ifcsemtyp{Xendtwolines@}\
@currentseries}}}%
5165     {}%
5166     {%
5167     \setistwofollowinglines{#1}{#2}{#4}{#5}%
5168     \ifboolexpr{%
5169     (%
5170         togl {Xendtwolinesbutnotmore@\@currentseries}%
5171         and not%
5172         (%
5173             bool {istwofollowinglines@}%
5174             )%
5175     )%
5176     or%
5177     (%
5178         (not test{\ifnumequal{#1}{#4}})%
5179         and togl{Xendtwolinesonlyinsamepage@\@currentseries}%
5180     )%
5181     }%
5182     {}%
5183     {%
5184     \l@d@dashfalse%
5185     \l@d@Xtwolinestrue%
5186     \l@d@elinfalse%
5187     \l@d@eslfalse%
5188     \ifcsemtyp{Xendmorethantwolines@\@currentseries}%
5189     {}%
5190     {\ifistwofollowinglines@\else%
5191     \l@d@Xmorethantwolinestrue%
5192     \fi%
5193     }%
5194     }%
5195     }%
5196     \fi%
5197 %

```

End of \setprintendlines.

```

5198 }%
5199 %

```

\printendlines Now we are ready to print it all.

```

5200 \def\printendlines#1|#2|#3|#4|#5|#6|#7|#8|{%
5201     \begingroup
5202     \setprintendlines{#1}{#2}{#3}{#4}{#5}{#6}%

```

5203 %

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.

```
5204 \ifdimequal{\csuse{Xendboxstartlinenum@\@currentseries}}{0pt}%
5205 {\bgroup}%
5206 {\leavevmode\hbox to \csuse{Xendboxstartlinenum@\@currentseries}\bgroup
\hfill}}%
5207 %
```

Then, print the starting page number-

```
5208 \ifboolexpr{%
5209   (%
5210     test{\ifcsstring{prevpagenum@\@currentseries}{#1}}%
5211     and not%
5212     (togl{Xendpagenumberonlyfirstifsingle@\@currentseries} and bool{
1@d@pnum})}%
5213   )%
5214   or%
5215   (%
5216     test {\ifcsstring{prevpagerange@\@currentseries}{#1-#4}}%
5217   )%
5218 }%
5219 {%
5220 \ifcsemtyp{Xendsympagenum@\@currentseries}%
5221   {\hspace{\csuse{Xendinplaceofpagenumber@\@currentseries}}}%
5222   {\csuse{Xendsympagenum@\@currentseries}}%
5223 }%
5224 {%
5225 \wrap@edcrossref{\@this@crossref@start}{\printnpnum{#1}}%
5226 }%
5227 %
```

Then, determine what must be printed before the start line.

```
5228 \ifl@d@dash%
5229 \ifl@d@pnum%
5230   \csuse{Xendlineprefixsingle@\@currentseries}%
5231 \else%
5232   \ifcsemtyp{Xendlineprefixmore@\@currentseries}%
5233     {\csuse{Xendlineprefixsingle@\@currentseries}}%
5234     {\csuse{Xendlineprefixmore@\@currentseries}}%
5235 \fi%
5236 \else%
5237   \csuse{Xendlineprefixsingle@\@currentseries}%
5238 \fi%
5239 %
```

The print the starting line, followed, if needed, by the side flag and the starting sub line number, and the line number annotation.

```

5240 \ifcsstring{Xendlinenumannotationposition@\@currentseries}{before}%
5241   {\@annot@start@print}%
5242   }%
5243 \wrap@edcrossref{\@this@crossref@start}{%
5244   \ifledRcol@%
5245     \linenumrepR{#2}%
5246   \else%
5247     \linenumrep{#2}%
5248   \fi%
5249 }%
5250 \iftoggle{Xendlineflag@\@currentseries}{\ifledRcol@\@Rlineflag\fi}{%
5251 \ifl@d@ssub%
5252   \csuse{Xendsublinesep@\@currentseries}%
5253   \wrap@edcrossref{\@this@crossref@start}{%
5254     \ifledRcol@%
5255       \sublinenumrepR{#3}%
5256     \else%
5257       \sublinenumrep{#3}%
5258     \fi%
5259   }%
5260 \fi%
5261 \ifcsstring{Xendlinenumannotationposition@\@currentseries}{after}%
5262   {\@annot@start@print}%
5263   }%
5264 %

```

Close the box.

```

5265 \egroup%
5266 %

```

Open the box for the ending line number.

```

5267 \ifdimequal{\csuse{Xendboxendlinenum@\@currentseries}}{0pt}%
5268   {\bgroup}%
5269   {\hbox to \csuse{Xendboxendlinenum@\@currentseries}\bgroup}%
5270 %

```

Print the dash + the ending line number, or the line number range symbol.

```

5271 \ifl@d@Xtwolines%
5272   \ifl@d@Xmorethantwolines%
5273     \csuse{Xendmorethantwolines@\@currentseries}%
5274   \else%
5275     \csuse{Xendtwolines@\@currentseries}%
5276   \fi%
5277 \else%
5278   \ifl@d@dash%
5279     \ifdefined\linerangesep@%
5280       \linerangesep@%

```

```

5281     \else%
5282         \csuse{Xendlinerrangeseparator@\@currentseries}%
5283     \fi%
5284 \fi%
5285 %

```

Print the ending page number.

```

5286 \ifl@d@pnum%
5287 \ifcsstring{prevpagerange@\@currentseries}{#1-#4}%
5288 {%
5289 \ifcsempy{Xendsympagenum@\@currentseries}%
5290 {\hspace{\csuse{Xendingplaceofpagenumber@\@currentseries}}}%
5291 {\csuse{Xendsympagenum@\@currentseries}}%
5292 }%
5293 {%
5294 \wrap@edcrossref{\@this@crossref@end}\printpnum{#4}%
5295 }%
5296 \fi%
5297 %

```

Print the ending line number, with, if needed, the line prefix, and followed by the side flag, the subtitle number, and line number annotation.

```

5298 \ifcsstring{Xendlinenumannotationposition@\@currentseries}{before}%
5299 {\@annot@end@print}%
5300 }%
5301 \ifl@d@elin%
5302 \ifl@d@pnum\csuse{Xendlineprefixsingle@\@currentseries}\fi%
5303 \wrap@edcrossref{\@this@crossref@end}{%
5304 \ifledRcol@%
5305 \linenumrepR{#5}%
5306 \else%
5307 \linenumrep{#5}%
5308 \fi%
5309 }%
5310 \iftoggle{Xendlineflag@\@currentseries}{\ifledRcol@\@Rlineflag\fi}{%
5311 \fi%
5312 \ifl@d@esl%
5313 \ifl@d@elin%
5314 \csuse{Xendsublinesep@\@currentseries}%
5315 \fi%
5316 \wrap@edcrossref{\@this@crossref@end}{%
5317 \ifledRcol@%
5318 \sublinenumrepR{#6}%
5319 \else%
5320 \sublinenumrep{#6}%
5321 \fi%
5322 }%
5323 \fi%
5324 \ifcsstring{Xendlinenumannotationposition@\@currentseries}{after}%
5325 {\@annot@end@print}%

```

```

5326     {}%
5327     \fi%
5328 %

```

Close the ending line box.

```

5329     \ifdimequal{\csuse{Xendboxendlinenum@\@currentseries}}{0pt}%
5330     {}%
5331     {\hfill}%Prevent underfull hbox
5332     \egroup%
5333 %

```

And, finally, save, if needed, the current page number for the Xendpagenumberonlyfirst hooks.

```

5334     \iftoggle{Xendpagenumberonlyfirst@\@currentseries}%
5335     {\iftoggle{Xendpagenumberonlyfirstintwo@\@currentseries}%
5336     {\csxdef{prevpagerange@\@currentseries}{#1-#4}}%
5337     {\csxdef{prevpagenum@\@currentseries}{#4}}%
5338     }%
5339     {}%
5340 %

```

Now, the end of \printendlines macro.

```

5341     \endgroup%
5342 }%
5343 %
5344 %

```

\printnpnum A macro to print a page number in an endnote. Should not be override anymore

```

5345 \newcommand*\printnpnum[1]{\csuse{Xendbeforepagenumber@\@currentseries
5346 }#1\csuse{Xendafterpagenumber@\@currentseries}}
5347 %

```

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@.

```

5348 \newcommand{\newseries}[1]{%
5349     \def\do##1{\newseries@{##1}}%
5350     \docsvlist{#1}
5351 }
5352 %

```

\@series The \series@ macro is an etoolbox list, which contains the name of all series.


```
5353 \newcommand{\@series}{-}
5354 %
```

The command `\newseries@series` creates a new series of the footnote.

```
\newseries@35 \newcommand{\newseries@}[1]{
5356 %
```

XXI.1 Test if series is still existing

```
5357 \xifinlist{#1}{\@series}{\led@warn@SeriesStillExist{#1}}%
5358 {%
5359 %
```

XXI.2 Init specific to reledpar

When calling `\newseries@` after having loaded `reledpar`, we need to load specific setting.

```
5360 \ifdefined\newseries@par%
5361 \newseries@par{#1}%
5362 \fi%
5363 %
```

XXI.3 For critical footnotes

Critical footnotes are those which start with letters. We look for the `\nocritical` option of `reledmac`.

```
5364 \unless\ifnocritical@
5365 %
```

XXI.3.1 Options

```
5366 \newtoggle{Xlineflag@#1}
5367 \newtoggle{Xparindent@#1}
5368 \newtoggle{Xlemmadisablefontselection@#1}
5369 \csgdef{Xwrapcontent@#1}{-}%
5370 \csgdef{Xbeforeinserting@#1}{-}%
5371 \csgdef{Xhangindent@#1}{Opt}%
5372 \csgdef{Xragged@#1}{-}%
5373 \csgdef{Xhsizetwocol@#1}{0.45 \hsize}%
5374 \csgdef{Xhsizethreecol@#1}{.3 \hsize}%
5375 \csgdef{Xcolalign@#1}{\raggedright}%
5376 \csgdef{Xnotenumfont@#1}{\normalfont}%
5377 \csgdef{Xnotefontsize@#1}{\footnotesize}%
5378 \csgdef{Xhooknote@#1}{-}%
```

```

5379 \csgdef{Xbhookgroup@#1}{}%
5380
5381 \csgdef{Xboxlinenum@#1}{Opt}%
5382 \csgdef{Xboxlinenumalign@#1}{L}%
5383
5384 \csgdef{Xboxstartlinenum@#1}{Opt}%
5385 \csgdef{Xboxendlinenum@#1}{Opt}%
5386
5387 \csgdef{Xboxsymlinenum@#1}{Opt}%
5388 \newtoggle{Xnumberonlyfirstinline@#1}%
5389 \newtoggle{Xgroupbyline@#1}%
5390 \newtoggle{Xgroupbylineseparatetwolines@#1}%
5391 \newtoggle{Xnumberonlyfirstintwolines@#1}%
5392 \csgdef{Xtwolines@#1}{}%
5393 \csgdef{Xmorethantwolines@#1}{}%
5394 \csgdef{Xsublinesep@#1}{\fullstop}%
5395 \csgdef{Xpagelinesep@#1}{\csname Xsublinesep@#1\endcsname}%for
backward compatibility, call Xsublinesep@#1
5396 \newtoggle{Xtwolinesbutnotmore@#1}%
5397 \newtoggle{Xtwolinesonlyinsamepage@#1}%
5398 \newtoggle{Xonlypstart@#1}%
5399 \newtoggle{Xpstarteverytime@#1}%
5400 \newtoggle{Xpstart@#1}%
5401 \newtoggle{Xstanza@#1}%
5402 \csgdef{Xstanzaseparator@#1}{}%
5403 \csgdef{Xsymlinenum@#1}{}%
5404 \newtoggle{Xnonumber@#1}%
5405 \csgdef{Xbeforenumber@#1}{Opt}%
5406 \csgdef{Xtxtbeforenumber@#1}{}%
5407 \csgdef{Xafternumber@#1}{0.5em}%
5408 \newtoggle{Xnonbreakableafternumber@#1}%
5409 \csgdef{Xbeforesymlinenum@#1}{\csuse{Xbeforenumber@#1}}%
5410 \csgdef{Xaftersymlinenum@#1}{\csuse{Xafternumber@#1}}%
5411 \csgdef{Xinplaceofnumber@#1}{1em}%
5412 \global\cslet{Xlemmaseparator@#1}{\rbracket}%
5413 \csgdef{Xbeforelemmaseparator@#1}{0em}%
5414 \csgdef{Xafterlemmaseparator@#1}{0.5em}%
5415 \csgdef{Xinplaceoflemmaseparator@#1}{1em}%
5416 \csgdef{Xbeforenotes@#1}{1.2em \@plus .6em \@minus .6em}
5417 \csgdef{Xafterrule@#1}{Opt}
5418
5419 \csgdef{Xtxtbeforenotes@#1}{%
5420 \newtoggle{Xtxtbeforenotes@#1@typeset}}%Not directly used by user,
but internal
5421 \newtoggle{Xtxtbeforenotesonlyonce@#1}%
5422
5423 \csgdef{Xmaxhnotes@#1}{0.8\vsizer}
5424 \newtoggle{Xnoteswidthliketwocolumns@#1}%
5425 \csgdef{Xparafootsep@#1}{}%
5426 \csgdef{Xafternote@#1}{1em plus.4em minus.4em}

```

```

5427 \csgdef{Xlinerangeseparator@#1}{\endashchar}%
5428
5429 \csgdef{Xlemmafont@#1}{}%
5430 \csgdef{Xwraplemma@#1}{}
5431 \csgdef{Xwidth@#1}{\hspace}%
5432 \csgdef{Xwraplinenumannotation@#1}{\textsuperscript}%
5433 %

```

XXI.3.2 Create inserts, needed to add notes in foot

As regards inserts, see chapter 15 of *The TeXbook* by D. Knuth.

```

5434 \expandafter\newinsert\csname #1footins\endcsname%
5435 \unless\ifnoledgroup%
5436 \expandafter\newinsert\csname mp#1footins\endcsname%
5437 \fi%
5438 %

```

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.

```

5439 \global\notbool{parapparatus@}{\expandafter\newcommand\expandafter
*}{\expandafter\newcommand}\csname #1footnote\endcsname [2] [] {}%
5440 \if@edtext@secondarg%
5441 \ifledRcol%
5442 \ifcsstring{Xonlyside@#1}{L}{\
led@error@note@called@onrightside{#1footnote}}{}%
5443 \else%
5444 \ifcsstring{Xonlyside@#1}{R}{\
led@error@note@called@onleftside{#1footnote}}{}%
5445 \fi%
5446 \begingroup%
5447 \newcommand{\content}{##2}%
5448 \ifnumberedpar%
5449 \ifledRcol%
5450 \ifluatex%
5451 \footnotelang@lua[R]%
5452 \fi%
5453 \@ifundefined{xpg@main@language}%if polyglossia
5454 {}%
5455 {\footnotelang@poly[R]}%
5456 \footnoteoptions@{R}{##1}{true}%
5457 \xright@appenditem{%
5458 \ifbool{indtl@innote}%
5459 {\unexpanded{\let\index\nindex}}%
5460 {}%
5461 \ifbool{indtl@notenumber}%
5462 {\unexpanded{\let\index\nindex}}%There is no note
...number so

```

```

5463         {}%
5464         \noexpand\Xnote@true%
5465         \noexpand\prepare@Xprenotes{#1}%
5466         \noexpand\prepare@edindex@fornote{\l@d@nums}%
5467         \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.
5468         \noexpand\setcounter{stanzaR}{\the\c@stanzaR}%Save
stanzaR counter for footnote
5469         \unexpanded{\def\@this@crossref@start}{\theedtext:
start}%
5470         \unexpanded{\def\@this@crossref@end}{\theedtext:end}%
5471         \expandonce{\@beforeinsertofthisedtext}% Internal for
now, no reason to make it public
5472         \noexpand\parse@annot\l@current@annot|%
5473         \noexpand\csuse{v#1footnote}{#1}%
5474         {\l@d@nums}{\expandonce\@tag}{\expandonce\content}}
%
5475         \noexpand\Xnote@false%
5476         \unexpanded{\advance\@edindex@fornote@m@ne}%
5477         \unexpanded{\undef\@this@crossref@start}%
5478         \unexpanded{\undef\@this@crossref@end}%
5479         \ifbool{indtl@innote}%
5480         {\unexpanded{\let\index\orig@index}}%
5481         {}%
5482         \ifbool{indtl@notenumber}%
5483         {\unexpanded{\let\index\orig@index}}%
5484         {}%
5485         }\to\inserts@listR
5486         \footnoteoptions@{R}{#1}{false}%
5487         \global\advance\insert@countR \@ne%
5488         \else%
5489         \ifluatex%
5490         \footnotelang@lua%
5491         \fi%
5492         \@ifundefined{xpg@main@language}%if polyglossia
5493         {}%
5494         {\footnotelang@poly}%
5495         \footnoteoptions@{L}{#1}{true}%
5496         \xright@appenditem{%
5497         \ifbool{indtl@innote}%
5498         {\unexpanded{\let\index\nindex}}%
5499         {}%
5500         \ifbool{indtl@notenumber}%
5501         {\unexpanded{\let\index\nindex}}%There is no note
...number so
5502         {}%
5503         \noexpand\Xnote@true%
5504         \noexpand\prepare@Xprenotes{#1}%

```

```

5505         \noexpand\prepare@edindex@fornote{\l@d@nums}%
5506         \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.
5507         \ifl@pairing%
5508         \noexpand\setcounter{stanzaL}{\the\c@stanzaL}%Save
stanzaR counter for footnote
5509         \fi%
5510         \unexpanded{\def\@this@crossref@start}{\theedtext:
start}}%
5511         \unexpanded{\def\@this@crossref@end}{\theedtext:end}%
5512         \expandonce{\@beforeinsertofthisedtext}%Internal for
now, no reason to make it public
5513         \noexpand\parse@annot\l@current@annot|%
5514         \noexpand\csuse{v#1footnote}%
5515         {#1}%
5516         {\l@d@nums}{\expandonce\@tag}{\expandonce\content
}}%
5517         \unexpanded{\undef\@this@crossref@start}%
5518         \unexpanded{\undef\@this@crossref@end}%
5519         \noexpand\Xnote@false%
5520         \unexpanded{\advance\@edindex@fornote@m@ne}%
5521         \ifbool{indtl@innote}%
5522         {\unexpanded{\let\index\orig@index}}%
5523         {}%
5524         \ifbool{indtl@notenumber}%
5525         {\unexpanded{\let\index\orig@index}}%
5526         {}%
5527         }\to\inserts@list
5528         \global\advance\insert@count \@ne%
5529         \footnoteoptions@{L}{##1}{false}%
5530         \fi
5531         \else
5532         \csuse{v#1footnote}{#1}{0|0|0|0|0|0|0|0}{##1}}%
5533         \fi%
5534         \endgroup%
5535         \else%
5536         \led@err@FootnoteNotInSecondArgEdtext{#1}%
5537         \fi%
5538         \ignorespaces%
5539         }
5540 %

```

Create counter used to determine on which page the previous note was called.

```

5541         \expandafter\newcount\csname #1prevpage@num\endcsname%
5542         \expandafter\newcount\csname #1prevpage@numR\endcsname%
5543 %

```

We need to be able to modify reledmac's footnote macros and restore their

```
5544 \global\csletcs{#1@@footnote}{#1footnote}
5545 %
```

XXI.3.4 Set standard display

```
5546 \Xarrangement@normal{#1}%
5547 %
```

End of for critical footnotes.

```
5548 \fi
5549 %
```

XXI.4 For familiar footnotes

Familiar footnotes are those which end with letters. We look for the `nofamiliar` option of `reledmac`.

```
5550 \unless\ifnofamiliar@
5551 %
```

XXI.4.1 Options

```
5552 \newtoggle{parindentX@#1}
5553 \csgdef{wrapcontentX@#1}{}%
5554 \csgdef{hangindentX@#1}{Opt}%
5555 \csgdef{beforeinsertingX@#1}{}%
5556 \csgdef{raggedX@#1}{}%
5557 \csgdef{hsizetwocolX@#1}{0.45 \hsize}%
5558 \csgdef{hsizethreecolX@#1}{.3 \hsize}%
5559 \csgdef{colalignX@#1}{\raggedright}%
5560 \csgdef{notenumfontX@#1}{\normalfont}%
5561 \csgdef{notefontsizeX@#1}{\footnotesize}%
5562 \csgdef{bhooknoteX@#1}{}%
5563 \csgdef{bhookgroupX@#1}{}%
5564 \csgdef{afterruleX@#1}{Opt}
5565 \csgdef{beforenotesX@#1}{1.2em \@plus .6em \@minus .6em}
5566 \csgdef{maxhnotesX@#1}{0.8\vsizel}%
5567 \newtoggle{noteswidthliketwocolumnsX@#1}%
5568 \csgdef{parafootsepX@#1}{}%
5569 \csgdef{afternoteX@#1}{1em plus.4em minus.4em}
5570 \csgdef{widthX@#1}{\hsizel}%
5571 \csgdef{txtbeforenotesX@#1}{}%
5572 \newtoggle{txtbeforenotesX@#1@typeset}%Not directly used by user,
but internal
5573 \newtoggle{txtbeforenotesonlyonceX@#1}%
5574 \csgdef{Xlinenumannotationposition@#1}{after}%
5575 % End of for familiar footnotes.
5576 % \subsubsection{Create inserts, needed to add notes in foot}
5577 % As regards inserts, see chapter 15 of the TeXBook by D. Knuth.
```

```

5578 % \begin{macrocode}
5579 \expandafter\newinsert\csname footins#1\endcsname%
5580 \unless\ifnoledgroup%
5581 \expandafter\newinsert\csname mpfootins#1\endcsname%
5582 \fi%
5583 %

```

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.

```

5584 \global\expandafter\newcommand\csname footnote#1\endcsname[1]{%
5585 \begingroup%
5586 \prepare@prenotesX{#1}%
5587 \newcommand{\content}{##1}%
5588 %
5589 %

```

If we use the `\csquotes` package, we reset the quotation level.

```

5590 \ifdefined\csq@qlevel%
5591 \csq@qlevel=0\relax%
5592 \fi%
5593 %

```

If we are preparing parallel typesetting, we cannot just increase the footnote counter. Read `reledpar`'s handbook about that (?? p. ??).

```

5594 \global\expandafter\advance\csname footnote#1@reading\
endcsname by \@ne%
5595 \ifboolexpr{bool{!@dpairing} or bool{!@dprintingpages} or
bool{!@dprintingcolumns}}{%
5596 \ifcsdef{footnote#1@reading\the\csname footnote#1@reading\
endcsname=typeset}%
5597 {\setcounter{footnote#1}{\cuse{footnote#1@reading\the\
csname footnote#1@reading\endcsname=typeset}}}%
5598 {\setcounter{footnote#1}{\the\csname footnote#1@reading
\endcsname}}%
5599 }{%
5600 \stepcounter{footnote#1}%
5601 }%
5602 %

```

We also have to check consistency with `\onlysideX` setting.

```

5603 \ifledRcol%
5604 \ifcsstring{onlysideX@#1}{L}{\
led@error@note@called@onrightside{footnote#1}}{%
5605 \else%
5606 \ifcsstring{onlysideX@#1}{R}{\
led@error@note@called@onleftside{footnote#1}}{%
5607 \fi%
5608 %

```

And now, the feature not depending of whether we are preparing parallel typesetting

```

5609         \protected@csxdef{@thefnmark#1}{\csuse{thefootnote#1}}%
5610         \nottoggle{nomk@}%Nomk is set to true when using \
footnoteXnomk with \parpackage
5611         {\csuse{@footnotemark#1}}%
5612         {}%
5613         \ifluatex%
5614         \xdef\footnote@luatextextdir{\the\textdir}%
5615         \xdef\footnote@luatexpardir{\the\pardir}%
5616         \fi%
5617         \if@ledgroup%
5618         \led@set@index@fornote{#1}%
5619         \fi%
5620         \csuse{vfootnote#1}{#1}{\expandonce\content}\m@mmf@prepare%
5621         \ifbool{indtl@innote}%
5622         {\let\index\orig@@index}%
5623         {}%
5624         \ifbool{indtl@notenumber}%
5625         {\let\index\orig@@index}%
5626         {}%
5627         \endgroup%
5628     }
5629 %

```

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 (?? p. ??).

```

5630     \newcounter{footnote#1}
5631     \global\expandafter\renewcommand\csname thefootnote#1\endcsname{\
arabic{footnote#1}}
5632     \expandafter\newcount\csname footnote#1@reading\endcsname%
5633 %

```

Create counter used to determine on which page the previous note was called.

```

5634     \expandafter\newcount\csname prevpage#1@num\endcsname%
5635     \expandafter\newcount\csname prevpage#1@numR\endcsname%
5636 %

```

Add `\let\footnoteX@gobble` to `\no@expands`.

```

5637     \expandafter\gappto\expandafter\no@expands\expandafter{\expandafter\
let\csname footnote#1\endcsname@gobble}%
5638 %

```

And now, define `\footnoteXmark` and `\footnoteXtext`, equivalent to `classical\footnotemark` and `\footnotetext`.

```

5639     \expandafter\newcommand\csname footnote#1mark\endcsname{%
5640         \begingroup%

```



```

5641     \prepare@prenotesX{#1}%
5642     \stepcounter{footnote#1}%
5643     \protected@csxdef{@thefnmark#1}{\csuse{thefootnote#1}}%
5644     \csuse{@footnotemark#1}%
5645     \m@mmf@prepare%
5646     \endgroup%
5647   }%
5648   \expandafter\newcommand\csname footnote#1text\endcsname[1]{%
5649     \begingroup%
5650     \csuse{vfootnote#1}{#1}{\expandonce{##1}}%
5651     \endgroup%
5652   }%
5653 %

```

Do not forget to initialize the series.

```

5654   \arrangementX@normal{#1}%
5655   \fi
5656 %

```

XXI.5 The endnotes

Endnotes are commands like `\Xendnote`, where `X` is a series letter. First, we check for the `noend` options.

```

5657   \unless\ifnoend@
5658 %

```

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`

```

5659     \expandafter\newwrite\csname l@d@#1end\endcsname%
5660     \expandafter\newif\csname ifl@dend@#1\endcsname%
5661 %

```

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.

```

5662     \global\expandafter\newcommandx\csname #1endnote\endcsname[2][1,
5663     usedefault]{%
5664       \bgroup%
5665       \newlinechar='40%

```

```

5666 \global\@noneed@Footnotetrue%
5667 \newcommand{\content}{##2}%
5668 \stepcounter{labidx}%
5669 \expandafter\immediate\expandafter\write\csname l@d@#1end\
endcsname{%
5670 \unexpanded{\def\sw@list@inedtext}%
5671 {\expandafter\unexpanded\expandafter{\sw@inthisedtext}}%
5672 \@percentchar\space%Explicit space, to add a linebreak in the
output file
5673 \noexpand\parse@annot\l@current@annot|\@percentchar\space%
5674 \expandafter\string\csname #1end\endcsname%
5675 {\ifnumberedpar@l@d@nums\fi}%
5676 {\ifnumberedpar@\expandonce\@tag\fi}%
5677 {\expandonce\content}%
5678 {#1}%
5679 {\unexpanded{##1}}%
5680 {\ifledRcol R\else L\fi}%
5681 {\theedtext}%
5682 \@percentchar%
5683 }%
5684 \egroup%
5685 \ignorespaces%
5686 }%
5687 %

```

XXI.5.3 Tools

The `\Xtoendnotes` command inserts any arbitrary content into the endnote file. It is an alias of the more generalist `\addtoendnotes`

```

5688 \global\expandafter\newcommand\csname #1toendnotes\endcsname [1]{%
5689 \toendnotes[#1]{##1}%
5690 }%
5691
5692 \expandafter\WithSuffix\expandafter\newcommand\csname #1toendnotes\
5693 endcsname*[1]{%
5694 \toendnotes*[#1]{##1}%
5695 }%
5696
5697 %

```

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.

```

5698 \global\cslet{#1end}{\@gobbleseven}
5699 %
5700 %

```

We need to store the number of times `\doendnotesbysection` is called for one series.

```

5701 \global\expandafter\newcount\csname #1end@bysection\endcsname%
5702 %

```

XXI.5.5 The options

```

5703 \csgdef{Xendwraplemma@#1}{%
5704 \csgdef{Xendwrapcontent@#1}{}%
5705 \csgdef{Xendtwolines@#1}{}%
5706 \csgdef{Xendmorethantwolines@#1}{}%
5707 \newtoggle{Xendtwolinesbutnotmore@#1}{%
5708 \newtoggle{Xendtwolinesonlyinsamepage@#1}{%
5709 \newtoggle{Xendlemmadisablefontselection@#1}%
5710 \csgdef{Xendnotenumfont@#1}{\normalfont}%
5711 \csgdef{Xendnotefontsize@#1}{\footnotesize}%
5712 \csgdef{Xendbhooknote@#1}{}%
5713
5714 \csgdef{Xendsublinesep@#1}{\fullstop}%
5715
5716 \csgdef{Xendbeforenumber@#1}{Opt}
5717 \csgdef{Xendafternumber@#1}{0.5em}
5718
5719 \csgdef{Xendboxlinenum@#1}{Opt}%
5720 \csgdef{Xendboxlinenumalign@#1}{L}%
5721
5722 \csgdef{Xendboxstartlinenum@#1}{Opt}%
5723 \csgdef{Xendboxendlinenum@#1}{Opt}%
5724
5725 \csgdef{Xendlemmaseparator@#1}{}%
5726 \csgdef{Xendbeforelemmaseparator@#1}{0em}%
5727 \csgdef{Xendafterlemmaseparator@#1}{0.5em}%
5728 \csgdef{Xendinplaceoflemmaseparator@#1}{0.5em}%
5729
5730 \newtoggle{Xendparagraph@#1}%
5731 \csgdef{Xendafternote@#1}{1em plus.4em minus.4em}%
5732 \csgdef{Xendsep@#1}{}%
5733
5734 \csgdef{Xendinplaceofnumber@#1}{Opt}%
5735 \newtoggle{Xendnonumber@#1}%
5736
5737 \csgdef{Xendhangindent@#1}{Opt}%
5738 \newtoggle{Xendnumberonlyfirstinline@#1}%
5739 \newtoggle{Xendnumberonlyfirstintwolines@#1}%
5740
5741 \csgdef{Xendbeforestylinenum@#1}{\csuse{Xendbeforenumber@#1}}%

```

```

5742 \csgdef{Xendaftersymmlinenumber@#1}{\csuse{Xendafternumber@#1}}%
5743 \csgdef{Xendsymmlinenumber@#1}{}%
5744 \csgdef{Xendboxsymmlinenumber@#1}{Opt}%
5745
5746 \csgdef{Xendbhooklinenumber@#1}{}%
5747 \csgdef{Xendehooklinenumber@#1}{}%
5748 \csgdef{Xendbhookinplaceofnumber@#1}{}%
5749 \csgdef{Xendehookinplaceofnumber@#1}{}%
5750
5751 \csgdef{Xendlinrangeseparator@#1}{\endashchar}%
5752
5753 \csgdef{Xendbeforepagenumber@#1}{p.}%
5754 \csgdef{Xendafterpagenumber@#1}{) }%
5755 \csgdef{Xendlineprefixsingle@#1}{}%
5756 \csgdef{Xendlineprefixmore@#1}{}%
5757
5758 \newtoggle{Xendlineflag@#1}
5759
5760 \csgdef{Xendlemmafont@#1}{}%
5761
5762 \csgdef{Xendlinenumannotationposition@#1}{after}%
5763 \csgdef{Xendwraplinenumannotation@#1}{\textsuperscript}%
5764
5765 \newtoggle{Xendpagenumberonlyfirst@#1}%
5766 \newtoggle{Xendpagenumberonlyfirstifsingle@#1}%
5767 \newtoggle{Xendpagenumberonlyfirstintwo@#1}%
5768 \csgdef{Xendsympagenumber@#1}{}%
5769 \csgdef{Xendinplaceofpagenumber@#1}{Opt}%
5770
5771 %

```

End of endnotes declaration

```

5772 \fi%
5773 %

```

Dump series in \@series

```

5774 \listxadd{\@series}{#1}
5775 }
5776 }% End of \newseries
5777 %

```

XXI.6 Init standards series (A,B,C,D,E)

```

5778 \expandafter\newseries\expandafter{\default@series}
5779 %

```

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.

```
5780 \newcommand{\seriesatbegin}[1]{%
5781   \StrDel{\@series}{#1}[\@series]%
5782   \edef\@new{}%
5783   \listead{\@new}{#1}%
5784   \listead{\@new}{\@series}%
5785   \xdef\@series{\@new}%
5786 }
5787 %
```

`\seriesatend` And `\seriesatend` moves the series to the end of the list.

```
5788 \newcommand{\seriesatend}[1]{%
5789   \StrDel{\@series}{#1}[\@series]%
5790   \edef\@new{}%
5791   \listead{\@new}{\@series}%
5792   \listead{\@new}{#1}%
5793   \xdef\@series{\@new}%
5794 }
5795 %
```

XXII.2 Test series order

`\ifseriesbefore` `\ifseriesbefore{⟨seriesA⟩}{⟨seriesB⟩}{⟨true⟩}{⟨false⟩}` expands `⟨true⟩` if `⟨seriesA⟩` is printed before `⟨seriesB⟩`, expands `⟨false⟩` otherwise.

```
5796 \newcommand{\ifseriesbefore}[4]{%
5797   \StrPosition{\@series}{#1}[\@first]%
5798   \StrPosition{\@series}{#2}[\@second]%
5799   \ifnumgreater{\@second}{\@first}{#3}{#4}%
5800 }
5801 %
```

XXII.2.1 Get the first series

In some specific case, we need to know the first series of the list of series.

```
\@getfirstseries02 \newcommand{\@getfirstseries}{%
5803   \ifdefempty{\@series}%
5804     {\xdef\@firstseries{}}%
5805     {\StrChar{\@series}{1}[\@firstseries]}%
5806 }%
5807 %
```

XXII.3 Series setting

XXII.3.1 General way of working

The setting's command (like `\numberonlyfirstinline`), also called “hooks” can be divided in two categories: those which require a string values and those which require a boolean value. The first category includes those which require a length value, because we store the length's expression send by user and we evaluate it only in the commands which requires to know the setting. The second category require boolean value only when it is set to FALSE. Otherwise, we understand the insinuated value is TRUE.

For each “hook” command, we store the value in commands (first category) or a `etoolbox`'s toggle (second category) which names are in the form `\<hook>@<series>`. For example, when calling `\twolines{<sq.>}`, we store `sq.` in commands `\twolines@A`, `\twolines@B`, `\twolines@C`...for each series defined for use with `reledmac`, or, if the [`<series>`] optional argument was send, for each series of this argument.

These values are tested in some specific places, scattered in all the code, depending of their effects. The default values are defined by the `\newseries@` command.

In order to prevent code duplication, we have created some generic commands. Some of them change the value of any hook send as argument. Some other, getting a hook name, generate the user level commands.

XXII.3.2 Tools to set options

`\settoggle@series` `\settoggle@series{<series>}{<toggle>}{<value>}` is a generic command to switch toggles for some series. The arguments are:

- #1 (mandatory): the series for which the hooks should be set. If empty, all the series will be affected.
- #2 (mandatory): the name of the hook.
- #3 (mandatory): the new value of toggle (true or false).
- #4 (optional): if equal to `reload`, reload the footnote setting (call again `\Xarrangement` or `\arrangementX` or ... depending of the footnote display).
- #5 (optional): if not empty, and if #1 is empty, change the hook setting for pseudo-series, as `appref`.

```

5808 \newcommandx{\settoggle@series}[5][4,5,usedefault]{%
5809   \def\do##1{%
5810     \global\settoggle{#2@##1}{#3}%
5811     \ifstrequal{#4}{critical}{
5812       \csuse{Xarrangement@}\csuse{series@display##1}{##1}%
5813     }{}
5814     \ifstrequal{#4}{familiar}{
5815       \csuse{arrangementX@}\csuse{series@displayX##1}{##1}%
5816     }{}
5817   }%
5818   \ifstreempty{#1}{%
```

```

5819     \dolistloop{\@series}%
5820     \ifstrempy{#5}{}{%
5821         \docsvlist{#5}%
5822     }
5823 }%
5824 {%
5825     \docsvlist{#1}%
5826 }%
5827 }
5828 %

```

`\setcommand@series` `\setcommand@series{<series>}{<command>}{<value>}` is a generic command to store hook's value into commands specific to some series. The arguments are:

- #1 (mandatory): the series for which the hooks should be set. If empty, all the series will be affected.
- #2 (mandatory): the name of the hook.
- #3 (mandatory): the new value of the hook/command.
- #4 (optional): if equal to `reload`, reload the footnote setting (call `\footnormal` or `\footparagraph` or ... depending of the footnote display).
- #5 (optional): if not empty, and if #1 is empty, change the hook setting for pseudo-series, as `appref`.

```

5829 \newcommandx{\setcommand@series}[5][4,5,usedefault]{%
5830     \def\do##1{
5831         \csgdef{#2@##1}{#3}
5832         \ifstrequal{#4}{critical}{%
5833             \csuse{Xarrangement@\csuse{series@display##1}}{##1}%
5834         }{}
5835         \ifstrequal{#4}{familiar}{%
5836             \csuse{arrangementX@\csuse{series@displayX##1}}{##1}%
5837         }{}%
5838     }%
5839     \ifstrempy{#1}{%
5840         \dolistloop{\@series}%
5841         \ifstrempy{#5}{}{%
5842             \docsvlist{#5}
5843         }
5844     }%
5845     {%
5846         \docsvlist{#1}%
5847     }%
5848 }%
5849 %

```

XXII.3.3 Tools to generate options commands

`\newhookcommand@series` `\newhookcommand@series\command` names is a generic command to add new commands for hooks, like `\Xhsizetwocol`. The first argument is the name of the hook, the second a comma-separated list of pseudo-series where the hook can be used, like `appref` in the case of `\Xtwolines`. The second argument is also used to create commands named `\<hookname><pseudoserries>`, like `\Xtwolinesappref`.

```

5850 \newcommandx{\newhookcommand@series}[2][2,usedefault]{%
5851   \global\expandafter\newcommand\expandafter*\csname #1\endcsname [2] [] {%
5852     \setcommand@series{##1}{#1}{##2} [] [#2]%
5853   }%
5854   \ifstrempy{#2}{-}{%
5855     \def\do##1{%
5856       \global\expandafter\newcommand\expandafter*\csname #1##1\endcsname
5857       [1]{%
5858         \csuse{#1}{##1}{####1}%
5859       }%
5860     \docsvlist{#2}%
5861   }%
5862 }
5863 %

```

`\newhooktoggle@series` `\newhooktoggle@series\command` names is a generic command to add new commands for a new toggle hook, like `\Xnumberonlyfirstinline`. The second argument is also used to create commands named `\<hookname><pseudoserries>`, like `\Xtwolinesbutnotmoreappref`.

```

5864 \newcommandx{\newhooktoggle@series}[2][2,usedefault]{%
5865   \global\expandafter\newcommandx\expandafter*\csname #1\endcsname [2] [1,2={
5866     true},usedefault]{%
5867     \settoggle@series{##1}{#1}{##2} [] [#2]%
5868   }%
5869   \ifstrempy{#2}{-}{%
5870     \def\do##1{%
5871       \global\expandafter\newcommand\expandafter*\csname #1##1\endcsname{%
5872         \csuse{#1}{##1}%
5873       }%
5874     \docsvlist{#2}%
5875   }%
5876 }
5877 %

```

`\newhooktoggle@series@reload` `\newhookcommand@toggle@reload` does the same thing as `\newhooktoggle@series` but the commands created by this macro also reload the series arrangement, depending of type os notes

```

5878 \newcommand{\newhooktoggle@series@reload}[2]{%

```



```

5879 \global\expandafter\newcommand\expandafter*\csname #1\endcsname [2] [1,2={
true},usedefault]{%
5880 \settoggle@series{##1}{#1}{##2}[#2]%
5881 }%
5882 }%
5883 %

```

`\newhookcommand@series@reload` does the same thing as `\newhookcommand@series` but the commands created by this macro also reload the series' arrangement.

```

5884 \newcommand{\newhookcommand@series@reload} [2] {%
5885 \global\expandafter\newcommand\expandafter*\csname #1\endcsname [2] [] {%
5886 \setcommand@series{##1}{#1}{##2}[#2]%
5887 }%
5888 }
5889 %

```

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.

```

5890 \unless\ifnocritical@
5891 \newhookcommand@series{Xwrapcontent}%
5892 \newhookcommand@series{Xbeforeinserting}%
5893 \newhookcommand@series{Xlemmafont}%
5894 \newhookcommand@series{Xwraplemma}%
5895 \newhooktoggle@series{Xparindent}
5896 \newhookcommand@series{Xhangindent}
5897 \newhookcommand@series{Xragged}
5898 \newhookcommand@series{Xhsizetwocol}
5899 \newhookcommand@series{Xhsizethreecol}
5900 \newhookcommand@series{Xcolalign}%
5901 \newhookcommand@series{Xnotenumfont}
5902 \newhookcommand@series{Xbhooknote}
5903 \newhookcommand@series@reload{Xbhookgroup}{critical}
5904 \newhookcommand@series{Xboxsymlinenum}%
5905 \newhookcommand@series{Xsymlinenum}
5906 \newhookcommand@series{Xbeforenumber}
5907 \newhookcommand@series{Xtxtbeforenumber}
5908 \newhookcommand@series{Xafternumber}
5909 \newhookcommand@series{Xbeforesymlinenum}
5910 \newhookcommand@series{Xaftersymlinenum}
5911 \newhookcommand@series{Xinplaceofnumber}
5912 \newhookcommand@series{Xlemmaseparator}
5913 \newhookcommand@series{Xbeforelemmaseparator}
5914 \newhookcommand@series{Xafterlemmaseparator}
5915 \newhookcommand@series{Xinplaceoflemmaseparator}
5916 \newhookcommand@series{Xtxtbeforenotes}

```

```

5917 \newhooktoggle@series{Xtxtbeforenotesonlyonce}%
5918 \newhookcommand@series@reload{Xafterrule}{critical}
5919 \newhooktoggle@series{Xnumberonlyfirstinline}
5920 \newhooktoggle@series{Xnumberonlyfirstintwolines}
5921 \newhooktoggle@series{Xgroupbyline}%
5922 \newhooktoggle@series{Xgroupbylineseparatetwolines}%
5923 \newhooktoggle@series{Xnonumber}
5924 \newhooktoggle@series{Xpstart}
5925 \newhooktoggle@series{Xpstarteverytime}%
5926
5927 \newhooktoggle@series{Xstanza}%
5928 \newhookcommand@series{Xstanzaseparator}%
5929
5930 \newhooktoggle@series{Xonlypstart}
5931 \newhooktoggle@series{Xnonbreakableafternumber}
5932 \newhooktoggle@series{Xlemmadisablefontselection}
5933 \newhookcommand@series@reload{Xmaxhnotes}{critical}
5934 \newhookcommand@series@reload{Xbeforenotes}{critical}
5935 \newhooktoggle@series@reload{Xnoteswidthliketwocolumns}{critical}%
5936 \newhookcommand@series@reload{Xnotefontsize}{critical}
5937
5938 \newhookcommand@series{Xboxlinenum}%
5939 \newhookcommand@series{Xboxlinenumalign}%
5940
5941 \newhookcommand@series{Xboxstartlinenum}%
5942 \newhookcommand@series{Xboxendlinenum}%
5943
5944 \newhookcommand@series{Xafternote}%
5945 \newhookcommand@series{Xparafotsep}
5946 \newhookcommand@series@reload{Xwidth}{critical}%
5947
5948 \ifundef{\Xhsize}%
5949   {%
5950     \newcommandx{\Xhsize}[2][1,usedefault]{%
5951       \led@warning@Xhsize@deprecated%
5952       \Xwidth[#1]{#2}%
5953     }%
5954   }%
5955   {}%
5956 \fi
5957 %

```

Now, hooks for critical notes which also apply for crossreferencing and line number in page sides.

```

5958 \newhooktoggle@series{Xlineflag}[appref,SEref]
5959 \newhookcommand@series{Xtwolines}[appref,SEref]
5960 \newhookcommand@series{Xmorethantwolines}[appref,SEref]
5961 \newhookcommand@series{Xsublinesep}[appref,SEref,side]%
5962 \newhookcommand@series{Xpagelinesep}[appref,SEref,side]%
5963 \newhooktoggle@series{Xtwolinesbutnotmore}[appref,SEref]

```

```

5964 \newhooktoggle@series{Xtwolinesonlyinsamepage}[appref,SEref]
5965 \newhookcommand@series{Xlinorangeseperator}[appref,SEref]
5966 \newhookcommand@series{Xlinenumannotationposition}[side,appref,SEref]%
5967 \newhookcommand@series{Xwraplinenumannotation}[side,ref,appref,SEref]%
5968 %

```

XXII.3.5 Options for familiar notes

Before generating the optional commands for familiar notes, we check the `\nofamiliar` option.

```

5969 \unless\ifnofamiliar@
5970 \newhookcommand@series{wrapcontentX}%
5971 \newhookcommand@series{beforeinsertingX}%
5972 \newhooktoggle@series{parindentX}
5973 \newhookcommand@series{hangindentX}
5974 \newhookcommand@series{raggedX}
5975 \newhookcommand@series{hsizetwocolX}
5976 \newhookcommand@series{hsizethreecolX}
5977 \newhookcommand@series{colalignX}%
5978 \newhookcommand@series{notenumfontX}
5979 \newhookcommand@series{bhooknoteX}
5980 \newhookcommand@series@reload{bhookgroupX}{familiar}
5981 \newhookcommand@series@reload{beforenotesX}{familiar}
5982 \newhookcommand@series@reload{maxhnotesX}{familiar}
5983 \newhooktoggle@series@reload{noteswidthliketwocolumnsX}{familiar}%
5984 \newhookcommand@series@reload{afterruleX}{familiar}
5985 \newhookcommand@series@reload{notefontsizeX}{familiar}
5986 \newhookcommand@series{afternoteX}
5987 \newhookcommand@series{parafootsepX}
5988 \newhookcommand@series{txtbeforenotesX}%
5989 \newhooktoggle@series{txtbeforenotesonlyonceX}%
5990 \newhookcommand@series@reload{widthX}{familiar}%
5991 \ifundef{\hsizeX}%
5992 {%
5993 \newcommandx{\hsizeX}[2][1,usedefault]{%
5994 \led@warning@hsizeX@deprecated%
5995 \widthX[#1]{#2}%
5996 }%
5997 }%
5998 {}%
5999 \fi
6000 %

```

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.

```

6001 \unless\ifnoend@
6002   \newhookcommand@series{Xendwraplemma}
6003   \newhookcommand@series{Xendwrapcontent}
6004   \newhookcommand@series{Xendnotenumfont}
6005   \newhookcommand@series{Xendlemmafont}%
6006   \newhookcommand@series{Xendbhooknote}
6007
6008   \newhookcommand@series{Xendboxlinenum}%
6009   \newhookcommand@series{Xendboxlinenumalign}%
6010
6011   \newhookcommand@series{Xendboxstartlinenum}%
6012   \newhookcommand@series{Xendboxendlinenum}%
6013
6014   \newhookcommand@series{Xendnotefontsize}
6015   \newhooktoggle@series{Xendlemmadisablefontselection}
6016   \newhookcommand@series{Xendlemmaseparator}
6017   \newhookcommand@series{Xendbeforelemmaseparator}
6018   \newhookcommand@series{Xendafterlemmaseparator}
6019   \newhookcommand@series{Xendinplaceoflemmaseparator}
6020
6021   \newhookcommand@series{Xendbeforenumber}%
6022   \newhookcommand@series{Xendafternumber}%
6023
6024   \newhooktoggle@series{Xendparagraph}
6025   \newhookcommand@series{Xendafternote}
6026   \newhookcommand@series{Xendsep}
6027
6028   \newhookcommand@series{Xendinplaceofnumber}%
6029   \newhooktoggle@series{Xendnonumber}%
6030
6031   \newhooktoggle@series{Xendnumberonlyfirstinline}%
6032   \newhooktoggle@series{Xendnumberonlyfirstintwolines}%
6033
6034   \newhookcommand@series{Xendsymmlinenum}%
6035   \newhookcommand@series{Xendbeforesymmlinenum}%
6036   \newhookcommand@series{Xendaftersymmlinenum}%
6037   \newhookcommand@series{Xendboxsymmlinenum}%
6038
6039   \newhookcommand@series{Xendbhooklinenumber}%
6040   \newhookcommand@series{Xendahooklinenumber}%
6041   \newhookcommand@series{Xendbhookinplaceofnumber}%
6042   \newhookcommand@series{Xendahookinplaceofnumber}%
6043
6044   \newhookcommand@series{Xendhangindent}%
6045
6046   \newhooktoggle@series{Xendpagenumberonlyfirst}%
6047   \newhooktoggle@series{Xendpagenumberonlyfirstifsingle}%
6048   \newhooktoggle@series{Xendpagenumberonlyfirstintwo}%
6049   \newhookcommand@series{Xendsympagenum}%
6050   \newhookcommand@series{Xendinplaceofpagenumber}%

```

```

6051 \fi
6052 \newhooktoggle@series{Xendlineflag}[apprefwithpage,SErefwithpage]
6053 \newhookcommand@series{Xendtwwolines}[apprefwithpage,SErefwithpage]
6054 \newhookcommand@series{Xendmorethantwwolines}[apprefwithpage,SErefwithpage]
6055 \newhooktoggle@series{Xendtwwolinesbutnotmore}[apprefwithpage,SErefwithpage]
6056 \newhooktoggle@series{Xendtwwolinesonlyinsamepage}[apprefwithpage,
6057 SErefwithpage]
6058 \newhookcommand@series{Xendlinerangeseparator}[apprefwithpage,SErefwithpage
6059 ]
6060 \newhookcommand@series{Xendbeforepagenumber}[apprefwithpage,SErefwithpage,
6061 SErefonlypage]
6062 \newhookcommand@series{Xendafterpagenumber}[apprefwithpage,SErefwithpage]
6063 \newhookcommand@series{Xendlineprefixsingle}[apprefwithpage,SErefwithpage]
6064 \newhookcommand@series{Xendlineprefixmore}[apprefwithpage,SErefwithpage]
6065 \newhookcommand@series{Xendsublinesep}[apprefwithpage,SErefwithpage]
6066 \newhookcommand@series{Xendlinenumannotationposition}[apprefwithpage,
6067 SErefwithpage]%
6068 \newhookcommand@series{Xendwraplinenumannotation}[apprefwithpage,
6069 SErefwithpage]%
6070 %

```

XXII.4 Hooks for a particular footnote

`\newhooktoggle@specific` `\newhooktoggle@specific` is a generic command to create boolean hook specific to a note.

```

6068 \newcommand{\newhooktoggle@specific}[1]{%
6069 \newtoggle{#1}%
6070 \define@key[mac]{truefootnoteoption}{#1}[]{\global\settoggle{#1}{true}}%
When enabling footnote option
6071 \define@key[mac]{falsefootnoteoption}{#1}[]{\global\settoggle{#1}{false
}}
6072 }
6073 %

```

`\newhookarg@specific` `\newhookarg@specific` is a generic command to create argumen hook specific to a note.

```

6074 \newcommand{\newhookarg@specific}[1]{%
6075 \define@key[mac]{truefootnoteoption}{#1}{\csgdef{#1}{##1}}%When enabling
footnote option
6076 \define@key[mac]{falsefootnoteoption}{#1}{\global\csundef{#1}}%When
disabling footnote option
6077 }
6078 %

```

And now, we define some hooks specific to a note.

```

6079 \newhooktoggle@specific{fulllines}%
6080 \newhooktoggle@specific{nonum}
6081 \newhooktoggle@specific{nosep}
6082 \newhookarg@specific{linangesep}
6083 %

```

`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.

```
6084 %
```

`\nomk@` `\nomk@` toggle is used by `reledpar` to remove the footnote mark in the text when using `\footnoteXmk`. Read `reledpar` handbook.

```

6085 \newtoggle{nomk@}%
6086 %

```

XXII.5 Alias

`\Xnolemmaseparator` `\Xnolemmaseparator[⟨series⟩]` is just an alias for `\Xlemmaseparator[⟨series⟩]{}`.

```

6087 \newcommand*{\Xnolemmaseparator}[1][1]{\Xlemmaseparator[#1]}
6088 %

```

XXIII Output routine

Now we begin the output routine and associated things.

XXIII.1 Extra footnotes output

With luck we might only have to change `\@makecol` and `\@reinserts` of the \TeX 's kernel. Since `reledmac`, we use `etoolbox`'s patching commands instead of overriding. It should provides better compatibility with other package which modify these commands

`\doxtrafeet` `\doxtrafeet` is the code extending `\@makecol` to cater for the extra `reledmac` feet. We have two categories of extra footnotes. By default, we order the footnote inserts so that the regular footnotes of \TeX are first, then familiar familiar footnotes and finally the critical footnotes.

```

6089 \newcommand*{\l@ddoxtrafeet}{%
6090   \IfStrEq{familiar-critical}{\@fnpos}
6091   {\do@feetX\do@Xfeet}%
6092   {%
6093   \IfStrEq{critical-familiar}{\@fnpos}%
6094   {\do@Xfeet\do@feetX}%
6095   {%

```

```

6096     \setbox\@outputbox \vbox{%
6097         \unvbox\@outputbox%
6098         \do@feet@custom@order{-}\@fnpos}%
6099     }%
6100 }%
6101 }%
6102 }%
6103 %
6104 %

```

`\do@feet@custom@order` `\do@feet@custom@order` is called when `\@fnpos` is neither ‘familiar-critical’, nor ‘critical-familiar’, that is, when the order is more complex. In this case, people must define the order for all footnote series. If they don’t, \TeX could perform an infinite run.

```

6105 \newcommand{\do@feet@custom@order}[2]{%
6106     \def\do##1{%
6107         \edef\@notesseries{\@firstoftwo##1}%
6108         \edef\@notetype{\@secondoftwo##1}%
6109         \ifdefstring{\@notetype}{critical}%
6110             {\csuse{#1append@Xnotes}\@notesseries}}%
6111         {\ifdefstring{\@notetype}{familiar}%
6112             {\csuse{#1append@notesX}\@notesseries}}%
6113         {}%
6114     }%
6115 }%
6116 \expandafter\docsvlist\expandafter{#2}%
6117 }%
6118 %

```

`\do@Xfeet` `\do@Xfeet` is the code extending `\@makecol` to cater to the extra critical feet.

```

6119 \newcommand*{\do@Xfeet}{%
6120     \setbox\@outputbox \vbox{%
6121         \unvbox\@outputbox
6122         \@opXfeet}}
6123 %

```

`\@opXfeet` The extra critical feet to be added to the output. . A macro which appends critical

`\append@Xnotes` notes to the output’s routine, also adding vertical space before notes

```

6124 \print@Xnotes \newcommand{\append@Xnotes}[1]{%
6125     \ifvoid\csuse{#1footins}\else%
6126         \global\skip\csuse{#1footins}=\csuse{Xbeforenotes@#1}%
6127         \global\advance\skip\csuse{#1footins} by\csuse{Xafterrule@#1}%
6128         \print@Xnotes{#1}%
6129     \fi%
6130 }%
6131 %

```

The normal way to add one series, `\print@Xnotes`, is replaced by `reledpar` when using `\Pages`.

```

6132 \newcommand\print@Xnotes[1]{%
6133   \xdef\@currentseries{#1}%
6134   \csuse{#1footstart}{#1}%
6135   \csuse{#1footgroup}{#1}%
6136 }%
6137 %

```

We print all series of notes by looping on them. We check before printing them that they are not voided.

```

6138 \newcommand*\@opXfeet{%
6139   \unless\ifnocritical@%
6140     \gdef\firstXseries@{%
6141       \def\do##1{%
6142         \append@Xnotes{##1}%
6143       }%
6144     \dolistloop{\@series}%
6145   \fi%
6146 }%
6147 %

```

`\l@ddodoreintrafeet` `\l@ddodoreintrafeet` is the code for catering for the extra footnotes within `\@reinserts`. We use the same category and ordering as in `\l@ddoxtrafeet`.

```

6148 \newcommand*\l@ddodoreintrafeet{%
6149   \IfStrEq{familiar-critical}{\@fnpos}
6150   {\@doreinfeetX\X@doreinfeet}%
6151   {%
6152   \IfStrEq{critical-familiar}{\@fnpos}%
6153   {\X@doreinfeet\@doreinfeetX}%
6154   {\@doreinfeetX\X@doreinfeet}%
6155   }%
6156 }
6157 %
6158 %

```

`\X@doreinfeet` `\X@doreinfeet` is the code for catering for the extra critical footnotes within `\@reinserts`.

```

6159 \newcommand*\X@doreinfeet{%
6160   \unless\ifnocritical@%
6161   \def\do##1{%
6162     \ifvoid\csuse{##1footins}\else%
6163     \insert\csuse{##1footins}{\unvbox\csuse{##1footins}}%
6164     \fi}%
6165   \dolistloop{\@series}
6166   \fi%
6167 }
6168 %
6169 %

```


`\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.

```

6170 \newcommand{\append@notesX}[1]{%
6171   \ifvoid\csuse{footins#1}\else%
6172     \global\skip\csuse{footins#1}=\csuse{beforenotesX@#1}%
6173     \global\advance\skip\csuse{footins#1} by\csuse{afterruleX@#1}%
6174     \print@notesX{#1}%
6175   \fi%
6176 }%
6177 %

```

The normal way to print one series of notes. `\print@Xnotes` is replaced by `reledpar` when using `\Pages`.

```

6178 \newcommand\print@notesX[1]{%
6179   \xdef\@currentseries{#1}%
6180   \csuse{footstart#1}{#1}%
6181   \csuse{footgroup#1}{#1}%
6182 }%
6183 %

```

We print all the series of notes by looping on them. We check before printing them that they are not voided.

```

6184 \newcommand*\do@feetX{%
6185   \unless\ifnofamiliar@%
6186     \gdef\firstseriesX@{}%
6187     \setbox\@outputbox \vbox{%
6188       \unvbox\@outputbox%
6189       \def\do##1{%
6190         \append@notesX{##1}%
6191       }%
6192       \dolistloop{\@series}%
6193     \fi%
6194   }%
6195
6196 \newcommand{\@doreinfeetX}{%
6197   \unless\ifnofamiliar@%
6198     \def\do##1{%
6199       \ifvoid\csuse{footins##1}\else
6200         \insert%
6201           \csuse{footins##1}
6202           {\unvbox\csuse{footins##1}}%
6203         \fi%
6204       }%
6205     \dolistloop{\@series}%
6206   \fi%
6207 }%
6208 %
6209 %

```

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.)

```

6210 \@ifclassloaded{memoir}{%
6211 %
        memoir is loaded so we use memoir's built in hooks.
6212 \g@addto@macro{\m@mdoextrafeet}{\l@ddoxtrafeet}%
6213 \g@addto@macro{\m@mdodoreinextrafeet}{\l@ddodoreinextrafeet}%
6214 }{%
6215 %

```

memoir has not been loaded, so patch `\@makecol` and `\@reinserts`. If the `fancyhdr` package < version 3.8 has been loaded, we patch the `\latex@makecol` command, because this package redefines the standard `\@makecol` in the preamble, to call `\latex@makecol` which have been `\let` to `\@makecol`. If this package is not loaded, we directly patch `\@makecol`. If the `fancyhdr` package \geq version 3.8, we also directly patch `\@makecol`, because `fancyhdr` does its own patch `\AtBeginDocument`.

```

6216 \ifboolexpr{%
6217   test{\@ifpackageloaded{fancyhdr}}%
6218   and test {\ifdef{\latex@makecol}}%
6219 }{%
6220   \patchcmd%
6221     {\latex@makecol}%
6222     {\xdef\@freelist{\@freelist\@midlist}}%
6223     {\xdef\@freelist{\@freelist\@midlist}\l@ddoxtrafeet}%
6224     {}%
6225     {\led@error@fail@patch@@makecol}%
6226   }{%
6227   \patchcmd%
6228     {\@makecol}%
6229     {\xdef\@freelist{\@freelist\@midlist}}%
6230     {\xdef\@freelist{\@freelist\@midlist}\l@ddoxtrafeet}%
6231     {}%
6232     {\led@error@fail@patch@@makecol}%
6233   }%
6234
6235 \patchcmd%
6236   {\@reinserts}%
6237   {\ifvbox}%
6238   {\l@ddodoreinextrafeet\ifvbox}%
6239   {}%
6240   {\led@error@fail@patch@@reinserts}%
6241 }
6242
6243 %

```

It turns out that `\@doclearpage` also needs modifying.

`\if@led@nofoot` We have to check if there are any leftover feet.

```
6244 \newif\if@led@nofoot
6245
6246 %

6247 \@ifclassloaded{memoir}{%
6248 %
```

If the `memoir` class is loaded we hook into its modified `\@doclearpage`.

```
\@mem@extranofeet 6249 \@g@addto@macro{\@mem@extranofeet}{%%
6250 \def\do#1{%
6251 \unless\ifnocritical@%
6252 \ifvoid\csuse{#1footins}\else\@mem@nofootfalse\fi%
6253 \fi%
6254 \unless\ifnofamiliar@%
6255 \ifvoid\csuse{footins#1}\else\@mem@nofootfalse\fi%
6256 \fi%
6257 }
6258 \dolistloop{\@series}%
6259 }%
6260 }{%
6261 %
```

As `memoir` is not loaded we have patch `\@doclearpage`.

```
\@led@testifnofoot 6262 \newcommand*\@led@testifnofoot{%
\@doclearpage 6263 \@led@nofoottrue%
6264 \ifvoid\footins\else%
6265 \@led@nofootfalse%
6266 \fi%
6267 \def\do##1{%
6268 \unless\ifnocritical@%
6269 \ifvoid\csuse{##1footins}\else%
6270 \@led@nofootfalse%
6271 \fi%
6272 \fi%
6273 \unless\ifnofamiliar@%
6274 \ifvoid\csuse{footins##1}\else%
6275 \@led@nofootfalse%
6276 \fi%
6277 \fi%
6278 }%
6279 \dolistloop{\@series}%
6280 }%
6281
```

```

6282 \pretocmd%
6283   {\@docclearpage}%
6284   {\@led@testifnofoot}%
6285   {}}%
6286   {\led@error@fail@patch@@docclearpage}%
6287
6288 \patchcmd%
6289   {\@docclearpage}%
6290   {\ifvoid\footins}%
6291   {\if@led@nofoot}%
6292   {}}%
6293   {\led@error@fail@patch@@docclearpage}%
6294
6295 }
6296
6297 %

```

XXIV 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.

```

6298 \list@create{\labelref@list}
6299 %

```

`\zz@@@` Two convenience macros to zero three / four labeling counters in one go.

```

\zz@@@
6300 \newcommand*\zz@@@{000|000|000}% Set three counters to zero in one go
6301 \newcommand*\zz@@@@{000|000|000|000}% Set four counters to zero in one go
6302
6303 %

```

`\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 \LaTeX 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.

```

6304 \newcommand*{\edlabel}[1]{%
6305   \leavevmode%
6306   \@bsphack%
6307   \ifboolexpr{bool{ledRcol} or bool{ledRcol@}}{%
6308     \ifXnote@%
6309     \protected@write\@auxout{%
6310       {\string\l@dmake@labelsR\space\thepage|\l@dparsedstartline|\l@dparsedstartsub|\annot@start|\the\c@pstartR|{#1}}%
6311       \ifdef{\hypertarget}%
6312         {\Hy@raisedlink{\hypertarget{#1}-{}}}%
6313         {}%
6314     \else%
6315       \write\linenum@outR{\string\@lab}%
6316       \ifx\labelref@listR\empty%
6317         \xdef\label@refs{\zz@@@}%
6318       \else%
6319         \gl@p\labelref@listR\to\label@refs%
6320       \fi%
6321       \ifvmode%
6322         \advancelabel@refs%
6323       \fi%
6324   }%

```

Use code from the kernel `\label` command to write the correct page number. Also define an `hypertarget` if `hyperref` package is loaded.

```

6325   \protected@write\@auxout{%
6326     {\string\l@dmake@labelsR\space\thepage|\label@refs|\the\c@pstartR|{#1}}%
6327     \ifdef{\hypertarget}%
6328       {\Hy@raisedlink{\hypertarget{#1}-{}}}%
6329       {}%
6330   \fi%
6331 }{%
6332   \ifXnote@%
6333     \ifl@dpairing%pstart or pstartL?
6334     \protected@write\@auxout{%
6335       {\string\l@dmake@labelsR\space\thepage|\l@dparsedstartline|\l@dparsedstartsub|\annot@start|\the\c@pstartL|{#1}}%
6336     \ifdef{\hypertarget}%

```

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.

```

6337     {\Hy@raisedlink{\hypertarget{#1}-{}}}%
6338     }%
6339     \else%
6340     \protected@write\@auxout{%
6341     {\string\l@dmake@labels\space\thepage|\l@dparsedstartline|\
l@dparsedstartsub|\annot@start|\the\c@pstart|{#1}}%
6342     \ifdef{\hypertarget}%
6343     {\Hy@raisedlink{\hypertarget{#1}-{}}}%
6344     }%
6345     \fi%
6346     \else%
6347     \write\linenum@out{\string\@lab}%
6348     \ifx\labelref@list\empty%
6349     \xdef\label@refs{\zz@@@}%
6350     \else%
6351     \gl@p\labelref@list\to\label@refs%
6352     \fi%
6353     \ifvmode%
6354     \advancelabel@refs%
6355     \fi%
6356     \ifl@pairing%Pstart or PstartL?
6357     \protected@write\@auxout{%
6358     {\string\l@dmake@labels\space\thepage|\label@refs|\the\c@pstartL
|{#1}}%
6359     \ifdef{\hypertarget}%
6360     {\Hy@raisedlink{\hypertarget{#1}-{}}}%
6361     }%
6362     \else%
6363     \protected@write\@auxout{%
6364     {\string\l@dmake@labels\space\thepage|\label@refs|\the\c@pstart
|{#1}}%
6365     \ifdef{\hypertarget}%
6366     {\Hy@raisedlink{\hypertarget{#1}-{}}}%
6367     }%
6368     \fi%
6369     \fi%
6370     }%
6371     \@esphack}%
6372
6373 %

```

`\advancelabel@refs` In cases where `\edlabel` is the first element in a paragraph, we have a problem with line counts, because line counts change only at the first horizontal box of the paragraph. Hence, we need to test `\edlabel` if it occurs at the start of a paragraph. To do so, we use `\ifvmode`. If the test is true, we must advance by one unit the amount of text we write into the `.aux` file. We do so using `\advancelabel@refs` command.

```

6374 \newcounter{line}%
6375 \newcounter{subline}%
6376 \newcounter{absline}%

```

```

6377 \newcommand{\advancelabel@refs}{%
6378   \setcounter{line}{\expandafter\labelrefsparseline\label@refs}%
6379   \stepcounter{line}%
6380   \setcounter{absline}{\expandafter\labelrefsparseabsline\label@refs}%
6381   \stepcounter{absline}%
6382   \ifsublines@%
6383     \setcounter{subline}{\expandafter\labelrefsparsesubline\label@refs}
6384   %
6385   \stepcounter{subline}{1}%
6386   \def\label@refs{\theline|\thesubline|\theabsline}%
6387 \else%
6388   \def\label@refs{\theline|0|\theabsline}%
6389 \fi%
6390 }
6391 \def\labelrefsparseline#1|#2|#3{#1}%
6392 \def\labelrefsparsesubline#1|#2|#3{#2}%
6393 \def\labelrefsparseabsline#1|#2|#3{#3}%
6394 %

```

`\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.

```

6394 \newcommand*{\l@dmake@labels}{%
6395 \def\l@dmake@labels#1|#2|#3|#4|#5|#6|#7{%
6396   \expandafter\ifx\csname the@label\csuse{XR@prefix}#7\endcsname%
6397   \relax%
6398 \else%
6399   \led@warn@DuplicateLabel{\csuse{XR@prefix}#7}%
6400 \fi
6401 \global\providetoggle{label@#7@ledRcol}%False is the default value of
6402 this toggle, which tell us if a label is linked to a right or a left side
6403 \csgdef{the@label\csuse{XR@prefix}#7}{#1|#2|#3|#4|#5|#6|\relax}%
6404 \ignorespaces}
6405 %

```

\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.

```

6406 \AtBeginDocument{%
6407   \def\l@dmake@labels#1|#2|#3|#4|#5|#6|#7{%
6408   }
6409 }
6410 %

```

`\@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`.

```

6411 \newcommand*{\@lab}{%
6412   \ifledRcol
6413     \xright@appenditem{\linenumr@p{\line@numR}}|{%
6414       \ifsublines@ \sublinenumr@p{\subline@numR}\else 0\fi|{%
6415         \the\absline@numR|{%
6416           \current@annot%
6417         }%
6418       \to\labelref@listR
6419     \else
6420       \xright@appenditem{%
6421         \linenumr@p{\line@num}|{%
6422           \ifsublines@ \sublinenumr@p{\subline@num}\else 0\fi|{%
6423             \the\absline@num|{%
6424               \current@annot%
6425             }%
6426           }%
6427         \to\labelref@list
6428       \fi}
6429 %

```

`\applabel` `\applabel`, if called in `\edtext` will insert automatically both a start and end label for the current `edtext` lines.

```

6430 \newcommand*{\applabel}[1]{%
6431   \if@edtext@secondarg@%
6432 %

```

Label should not be already defined.

```

6433   \ifcsundef{the@label#1}{%
6434     \csdef{the@label#1}{applabel}%
6435   }%
6436   {%
6437     \led@warn@DuplicateLabel{#1 (applabel)}%
6438   }%
6439 %

```

Parse the `\edtext` line numbers and annotations.

```

6440   \expandafter\l@dp@rsefootspec\l@d@nums|{%
6441   \expandafter\parse@annot\l@current@annot|{%
6442 %

```


Use the \LaTeX standard hack for label.

```
6443 \@bsphack%
6444 %
```

And now, write the data in the auxiliary file.

```
6445 \ifledRcol%
6446 \protected@write\@auxout{}%
6447 {\string\l@dmake@labelsR\space\l@dparsedstartpage|\
l@dparsedstartline|\l@dparsedstartsub||\annot@start|\the\c@pstartR|{#1:
start}}%
6448 \ifdef{\hypertarget}%
6449 {\Hy@raisedlink{\hypertarget{#1:start}}}%
6450 {}%
6451 \protected@write\@auxout{}%
6452 {\string\l@dmake@labelsR\space\l@dparsedendpage|\l@dparsedendline
||\l@dparsedendsub||\annot@end|\the\c@pstartR|{#1:end}}%
6453 \else%
6454 \ifl@dpairing%pstart or pstartL?
6455 \protected@write\@auxout{}%
6456 {\string\l@dmake@labels\space\l@dparsedstartpage|\
l@dparsedstartline|\l@dparsedstartsub||\annot@start|\the\c@pstartL|{#1:
start}}%
6457 \ifdef{\hypertarget}%
6458 {\Hy@raisedlink{\hypertarget{#1:start}}}%
6459 {}%
6460 \protected@write\@auxout{}%
6461 {\string\l@dmake@labels\space\l@dparsedendpage|\
l@dparsedendline|\l@dparsedendsub||\annot@end|\the\c@pstartL|{#1:end}}%
6462 \else%
6463 \protected@write\@auxout{}%
6464 {\string\l@dmake@labels\space\l@dparsedstartpage|\
l@dparsedstartline|\l@dparsedstartsub||\annot@start|\the\c@pstart|{#1:start
}}%
6465 \ifdef{\hypertarget}%
6466 {\Hy@raisedlink{\hypertarget{#1:start}}}%
6467 {}%
6468 \protected@write\@auxout{}%
6469 {\string\l@dmake@labels\space\l@dparsedendpage|\
l@dparsedendline|\l@dparsedendsub||\annot@end|\the\c@pstart|{#1:end}}%
6470 \fi%
6471 \fi%
6472 %
```

Use the \LaTeX standard hack for label.

```
6473 \@esphack%
6474 %
```

Warning if `\applabel` is called outside of `\edtext`.

```
6475 \else%
```

```

6476 \led@warn@AppLabelOutSecondArgEdtext{#1}%
6477 \fi%
6478 %

End of \applabel

6479 }%
6480 %

```

\edlabelS \edlabelS and \edlabelE are just used to mark the beginning and the end of a passage.

```

\edlabelE
\edlabelSE
6481 \newcommand{\edlabelS}[1]{%
6482 \edlabel{#1:start}%
6483 }
6484 \newcommand{\edlabelE}[1]{%
6485 \edlabel{#1:end}%
6486 }
6487 \newcommand{\edlabelSE}[1]{%
6488 \edlabelS{#1}%
6489 \edlabelE{#1}%
6490 }
6491 %

```

\wrap@edcrossref \wrap@edcrossref is called around all reledmac crossref commands, except those which start with x. It adds the hyperlink.

```

6492 \newrobustcmd{\wrap@edcrossref}[2]{%
6493 \ifdef{\hyperlink}%
6494 {\hyperlink{#1}{#2}}%
6495 {#2}%
6496 }
6497 %

```

\edpageref If the specified label exists, \edpageref gives its page number.

\xpageref For this reference command, as for the other two, a special version with prefix x is provided for use in places where the command is to be scanned as a number, as in \linenum. These special versions have two limitations: they do not print error messages if the reference is unknown, and they can't appear as the first label or reference command in the file; you must ensure that a \edlabel or a normal reference command appears first, or these x-commands will always return zeros.

ℒ_{TEX} already defines a \pageref, so changing the name to \edpageref.

```

6498 \newcommand*{\edpageref}[1]{\l@dref@undefined{#1}\wrap@edcrossref{#1}{\l@dgetref@num{1}{#1}}}
6499 \newcommand*{\xpageref}[1]{\l@dgetref@num{1}{#1}}
6500
6501 %

```

\edlineref If the specified label exists, \lineref gives its line number.

\xlineref

```

6502 \newcommand*\edlineref}[1]{%
6503   \l@dref@undefined{#1}%
6504   \wrap@edcrossref{#1}{%
6505     \providetoggle{label@#1@ledRcol}%Required for the first run, when the
label has not yet been parsed on the .aux file
6506     \iftoggle{label@#1@ledRcol}%
6507       {\linenumrepR{\l@dgetref@num{2}{#1}}}%
6508       {\linenumrep{\l@dgetref@num{2}{#1}}}%
6509     \xflagref{#1}%
6510   }%
6511 }%
6512 \newcommand*\xlineref}[1]{\l@dgetref@num{2}{#1}}%
6513 %
6514 %

```

\sublineref If the specified label exists, `\sublineref` gives its sub-line number.

```

\sublineref
\sublineref
6515 \newcommand*\sublineref}[1]{%
6516   \l@dref@undefined{#1}%
6517   \wrap@edcrossref{#1}{%
6518     \providetoggle{label@#1@ledRcol}%Required for the first run, when the
label has not yet been parsed on the .aux file
6519     \iftoggle{label@#1@ledRcol}%
6520       {\sublinenumrepR{\l@dgetref@num{3}{#1}}}%
6521       {\sublinenumrep{\l@dgetref@num{3}{#1}}}%
6522     }%
6523   }%
6524 \newcommand*\xsublineref}[1]{\l@dgetref@num{3}{#1}}
6525 %
6526 %

```

\xabslineref If the specified label exists, `\xabslineref` gives its absolute line number. That is used usually only by some `reledmac` internal macros.

```

6527 \newcommand*\xabslineref}[1]{\l@dgetref@num{4}{#1}}%
6528 %

```

\annotationref If the specified label exist, `\edannotationref` gives its line number annotation associated to it.

```

\annotationref
\annotationref
6529 \newcommand*\annotationref}[1]{%
6530   \l@dref@undefined{#1}%
6531   \wrap@edcrossref{#1}{%
6532     \l@wrap@ifnotempty{\Xwraplinenumannotation@ref}{%
6533       \l@dgetref@num{5}{#1}%
6534     }%
6535   }%
6536 }%
6537 \newcommand*\xannotationref}[1]{%
6538   \l@dgetref@num{5}{#1}%

```

```
6539 }%
6540 %
```

`\pstartref` If the specified label exists, `\pstartref` gives its pstart number.

`\xpstartref`

```
6541 \newcommand*\pstartref[1]{\l@dref@undefined{#1}\wrap@edcrossref{#1}{\l@
l@dgetref@num{6}{#1}}}%
6542 \newcommand*\xpstartref[1]{\l@dgetref@num{6}{#1}}%
6543 %
6544 %
```

`\xflagref` `\xflagref` finds the side flag of any ref defined with `\edlabel`.

```
6545 \newcommand*\xflagref[1]{\l@dgetref@num{7}{#1}}%
6546 %
```

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.

```
6547 \newcommand*\l@dref@undefined[1]{%
6548 \expandafter\ifx\csname the@label#1\endcsname\relax
6549 \led@warn@RefUndefined{#1}%
6550 \fi}
6551 %
6552 %
```

`\l@dgetref@num` Next, `\l@dgetref@num` fetches the number we want. It has two arguments: the first is simply a digit, specifying whether to fetch a page (1), line (2), sub-line (3), (4) pstart number or (5) side flag. (This switching is done by calling `\l@dlabel@parse`.) The second argument is the label-macro, which because of the `\@lab` macro above is defined to be a string of the type 123|456|789.

```
6553 \newcommand*\l@dgetref@num[2]{%
6554 \expandafter
6555 \ifx\csname the@label#2\endcsname \relax
6556 000%
6557 \else
6558 \expandafter\expandafter\expandafter
6559 \l@dlabel@parse\csname the@label#2\endcsname|#1%
6560 \fi}
6561 %
6562 %
```

`\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 arguments. 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 earlier sept numbers to pick out. (It was earlier given as the first argument of `\l@dgetref@num`.)

```

6563 \newcommand*{\l@dlabel@parse}{%
6564 \def\l@dlabel@parse#1|#2|#3|#4|#5|#6|#7|#8{%
6565   \ifcase #8%
6566     \or #1%
6567     \or #2%
6568     \or #3%
6569     \or #4%
6570     \or #5%
6571     \or #6%
6572     \or #7%
6573   \fi}
6574 %

```

`\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.

```

6575 \newcommand*{\xxref}[2]{%
6576   {%
6577     \expandafter\ifx\cename the@label#1\endcename\relax%
6578     \let\@tempa\zz@@@%
6579     \def\@tempc{}%
6580     \else%
6581       \def\@tempa{%
6582         \l@dgetref@num{1}{#1}|%
6583         \l@dgetref@num{2}{#1}|%
6584         \l@dgetref@num{3}{#1}%
6585       }%
6586       \edef\@tempc{%
6587         \l@dgetref@num{5}{#1}%
6588       }%
6589     \fi%
6590     \expandafter\ifx\cename the@label#2\endcename\relax%
6591     \let\@tempb\zz@@@%
6592     \def\@tempd{}%
6593     \else%

```

```

6594     \def\@tempb{%
6595         \l@getref@num{1}{#2}|%
6596         \l@getref@num{2}{#2}|%
6597         \l@getref@num{3}{#2}%
6598     }%
6599     \edef\@tempd{%
6600         \l@getref@num{5}{#2}%
6601     }%
6602     \fi%
6603     \global\appto\@beforeinsertofthisedtext{%
6604         \def\@this@crossref@start{#1}%
6605     }%
6606     \global\appto\@beforeinsertofthisedtext{%
6607         \def\@this@crossref@end{#2}%
6608     }%
6609     \linenum{\@tempa|\@tempb}%
6610     \edef\@tempe{\@tempc|\@tempd}%
6611     \expandafter\lineannot\expandafter{\@tempe}%
6612 }%
6613 }%
6614
6615
6616 %

```

`\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).

`\SErefwithpage` Here we define hooks similar to some those related to critical footnotes or endnotes. `\SErefwithpage` So, first declare the default value of the hooks for the pseudo-series. Also declare the internal toggle which are switch by `reledmac`.

```

6617 \def\Xtwolines@appref{}%
6618 \def\Xtwolines@SEref{}%
6619
6620 \def\Xmorethantwolines@appref{}%
6621 \def\Xmorethantwolines@SEref{}%
6622
6623 \def\Xlinerangeseparator@appref{\endashchar}%
6624 \def\Xlinerangeseparator@SEref{\endashchar}%
6625
6626 \def\Xsublinesep@appref{\fullstop}%
6627 \def\Xsublinesep@SEref{\fullstop}%
6628
6629 \def\Xpagelinesep@appref{\fullstop}%
6630 \def\Xpagelinesep@SEref{\fullstop}%
6631
6632
6633 \newtoggle{Xtwolinesbutnotmore@appref}%

```

```

6634 \newtoggle{Xtwolinesbutnotmore@SEref}%
6635
6636 \newtoggle{Xtwolinesonlyinsamepage@appref}%
6637
6638 \newtoggle{Xtwolinesonlyinsamepage@SEref}%
6639
6640 \newtoggle{Xlineflag@appref}%
6641 \toggletrue{Xlineflag@appref}%Here exception
6642 \newtoggle{Xlineflag@SEref}%
6643 \toggletrue{Xlineflag@SEref}%Here exception
6644
6645 \def\Xlinenumannotationposition@appref{after}%
6646 \def\Xlinenumannotationposition@SEref{after}%
6647
6648 \def\Xwraplinenumannotation@appref{\textsuperscript}%
6649 \def\Xwraplinenumannotation@SEref{\textsuperscript}%
6650
6651 \def\Xendtwolines@apprefwithpage{}%
6652 \def\Xendtwolines@SErefwithpage{}%
6653
6654 \def\Xendmoreethantwolines@apprefwithpage{}%
6655 \def\Xendmoreethantwolines@SErefwithpage{}%
6656
6657 \def\Xendlinerangeseparator@apprefwithpage{\endashchar}
6658 \def\Xendlinerangeseparator@SErefwithpage{\endashchar}
6659 \def\Xendlinerangeseparator@SErefonlypage{\endashchar}
6660
6661 \def\Xendbeforepagenumber@apprefwithpage{p.}%
6662 \def\Xendbeforepagenumber@SErefwithpage{p.}%
6663 \def\Xendbeforepagenumber@SEonlypage{p.}%
6664
6665 \def\Xendafterpagenumber@apprefwithpage{ }%
6666 \def\Xendafterpagenumber@SErefwithpage{ }%
6667
6668
6669 \def\Xendlineprefixsingle@apprefwithpage{}%
6670 \def\Xendlineprefixsingle@SErefwithpage{}%
6671
6672 \def\Xendlineprefixmore@apprefwithpage{}%
6673 \def\Xendlineprefixmore@SErefwithpage{}%
6674
6675 \newtoggle{Xendtwolinesbutnotmore@apprefwithpage}%
6676 \newtoggle{Xendtwolinesbutnotmore@SErefwithpage}%
6677
6678 \def\Xendsublinesep@apprefwithpage{\fullstop}%
6679 \def\Xendsublinesep@SErefwithpage{\fullstop}%
6680
6681 \newtoggle{Xendtwolinesonlyinsamepage@apprefwithpage}%
6682 \newtoggle{Xendtwolinesonlyinsamepage@SErefwithpage}%
6683

```

```

6684 \newtoggle{Xendlineflag@apprefwithpage}
6685 \toggletrue{Xendlineflag@apprefwithpage}%Here, exception
6686 \newtoggle{Xendlineflag@SErefwithpage}
6687 \toggletrue{Xendlineflag@SErefwithpage}%Here, exception
6688
6689 \def\Xendlinenumannotationposition@apprefwithpage{after}%
6690 \def\Xendlinenumannotationposition@SErefwithpage{after}%
6691
6692 \def\Xendwraplinenumannotation@apprefwithpage{\textsuperscript}%
6693 \def\Xendwraplinenumannotation@SErefwithpage{\textsuperscript}%
6694 %

```

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.

```

6695
6696 \gdef\Xboxstartlinenum@appref{Opt}
6697 \gdef\Xboxstartlinenum@SEref{Opt}
6698
6699 \gdef\Xboxendlinenum@appref{Opt}
6700 \gdef\Xboxendlinenum@SEref{Opt}
6701
6702 \gdef\Xendboxstartlinenum@apprefwithpage{Opt}
6703 \gdef\Xendboxstartlinenum@SErefwithpage{Opt}
6704
6705 \gdef\Xendboxendlinenum@apprefwithpage{Opt}
6706 \gdef\Xendboxendlinenum@SErefwithpage{Opt}
6707
6708 \newtoggle{Xendpagenumberonlyfirst@apprefwithpage}
6709 \newtoggle{Xendpagenumberonlyfirst@SErefwithpage}
6710
6711 \newtoggle{Xendpagenumberonlyfirstifsingle@apprefwithpage}
6712 \newtoggle{Xendpagenumberonlyfirstifsingle@SErefwithpage}
6713
6714 \newtoggle{Xendpagenumberonlyfirstintwo@apprefwithpage}
6715 \newtoggle{Xendpagenumberonlyfirstintwo@SErefwithpage}
6716
6717 \gdef\Xendsympagenum@apprefwithpage{}
6718 \gdef\Xendsympagenum@SErefwithpage{}
6719
6720 \gdef\Xendinplaceofpagenumber@apprefwithpage{}
6721 \gdef\Xendinplaceofpagenumber@SErefwithpage{}
6722
6723 %

```

Now, declare the default values of \@apprefprefixsingle and \@apprefprefixmore, \@SErefprefix, \@SErefprefixmore and the commands which defines them.

```

6724 \newcommand\@apprefprefixsingle{}%
6725 \newcommand\@SErefprefixsingle{}%

```



```

6726 \newcommand\@apprefprefixmore{}%
6727 \newcommand\@SErefprefixmore{}%
6728
6729
6730 \newcommand{\setapprefprefixsingle}[1]{%
6731   \gdef\@apprefprefixsingle{#1}%
6732 }
6733 \newcommand{\setSErefprefixsingle}[1]{%
6734   \gdef\@SErefprefixsingle{#1}%
6735 }
6736
6737 \newcommand{\setapprefprefixmore}[1]{%
6738   \gdef\@apprefprefixmore{#1}%
6739 }
6740 \newcommand{\setSErefprefixmore}[1]{%
6741   \gdef\@SErefprefixmore{#1}%
6742 }
6743
6744 %

```

And not `\setSErefonlypageprefixsingle` and `\setSErefonlypageprefixmore`.

```

6745 \newcommand{\setSErefonlypageprefixsingle}[1]{%
6746   \gdef\SErefonlypage@prefixsingle{#1}%
6747 }%
6748 \newcommand{\setSErefonlypageprefixmore}[1]{%
6749   \gdef\SErefonlypage@prefixmore{#1}%
6750 }%
6751 %

```

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.

```

6752
6753 \newcommandx{\appref}[2][1,usedefault]{\reformatted@{#1}{#2}{appref}}
6754 \newcommandx{\SEref}[2][1,usedefault]{\reformatted@{#1}{#2}{SEref}}
6755
6756 \newcommandx{\apprefwithpage}[2][1,usedefault]{\reformattedwithpage@
6757 {#1}{#2}{appref}}
6758 \newcommandx{\SErefwithpage}[2][1,usedefault]{\reformattedwithpage@
6759 {#1}{#2}{SEref}}
6760
6761 \newcommand{\reformatted@}[3]{%
6762   \def\do##1{%
6763     \setkeys[mac]{truefootnoteoption}{##1}%
6764   }%

```

```

6765 \notblank{#1}{\docsvlist{#1}}{ }%
6766 \xdef\@currentseries{#3}%
6767 \ifcempty{@#3prefixmore}%
6768   {\@apprefprefixsingle}%
6769   {%
6770     \IfEq{\xlineref{#2:start}}{\xlineref{#2:end}}%
6771     {\csuse{@#3prefixsingle}}%
6772     {\csuse{@#3prefixmore}}%
6773   }%
6774 \ifboolexpr{%
6775   test{\ifcsundef{the@label#2:start}}%
6776   or test{\ifcsundef{the@label#2:end}}%
6777 }%
6778 {\led@warn@pairRefUndefined{#2}\nfss@text{\reset@font\bfseries ??}}%
6779 {%
6780   \def\@this@crossref@start{#2:start}%
6781   \def\@this@crossref@end{#2:end}%
6782   \xdef\annot@start{\xannotationref{#2:start}}%
6783   \xdef\annot@end{\xannotationref{#2:end}}%
6784   \printlines\xpageref{#2:start}|\xlineref{#2:start}|\xsublineref{#2:
start}|\xpageref{#2:end}|\xlineref{#2:end}|\xsublineref{#2:end}|\relax|
xflagref{#2:start}|%
6785   \undef\@this@crossref@end%
6786   \undef\@this@crossref@start%
6787 }%
6788 \def\do##1{%
6789   \setkeys[mac]{falsefootnoteoption}{##1}%
6790 }%
6791 \notblank{#1}{\docsvlist{#1}}{ }%
6792 }%
6793
6794 \newcommand{\reformattedwithpage@}[3]{%
6795   \def\do##1{%
6796     \setkeys[mac]{truefootnoteoption}{##1}%
6797   }%
6798   \notblank{#1}{\docsvlist{#1}}{ }%
6799   \xdef\@currentseries{#3withpage}%
6800   \ifboolexpr{%
6801     test{\ifcsundef{the@label#2:start}}%
6802     or test{\ifcsundef{the@label#2:end}}%
6803   }%
6804   {\led@warn@pairRefUndefined{#2}\nfss@text{\reset@font\bfseries ??}}%
6805   {%
6806     \def\@this@crossref@start{#2:start}%
6807     \def\@this@crossref@end{#2:end}%
6808     \printendlines\xpageref{#2:start}|\xlineref{#2:start}|\xsublineref{#2:
start}|\xpageref{#2:end}|\xlineref{#2:end}|\xsublineref{#2:end}|\relax|
xflagref{#2:start}|%
6809     \undef\@this@crossref@end%
6810     \undef\@this@crossref@start%

```

```

6811 }%
6812 \def\do##1{%
6813   \setkeys[mac]{falsefootnoteoption}{##1}%
6814 }%
6815 \notblank{#1}{\docsvlist{#1}}{ }%
6816 }%
6817
6818 \newcommand{\reformattedonlypage@}[3]{%
6819   \def\do##1{%
6820     \setkeys[mac]{truefootnoteoption}{##1}%
6821   }%
6822   \notblank{#1}{\docsvlist{#1}}{ }%
6823   \xdef\@currentseries{#3onlypage}%
6824   \ifboolexpr{%
6825     test{\ifcsundef{the@label#2:start}}%
6826     or test{\ifcsundef{the@label#2:end}}%
6827   }%
6828   {\led@warn@pairRefUndefined{#2}\nfss@text{\reset@font\bfseries ??}}%
6829   {\ifnumequal{\xpageref{#2:end}}{\xpageref{#2:start}}%
6830     {%
6831       \ifcsvoid{#3onlypage@prefixsingle}%
6832       {}%
6833       {\csletcs{Xendbeforepagenumber@#3onlypage}{#3onlypage@prefixsingle
6834 }}%
6835       \printnpnum{%
6836         \wrap@edcrossref{#2:start}{\xpageref{#2:start}}%
6837       }%
6838     }%
6839     \ifcsvoid{#3onlypage@prefixmore}%
6840     {}%
6841     {\csletcs{Xendbeforepagenumber@#3onlypage}{#3onlypage@prefixmore}}%
6842     \ifdefined\linerangesep@%
6843       \printnpnum{%
6844         \wrap@edcrossref{#2:start}{\xpageref{#2:start}}%
6845         \linerangesep@%
6846         \wrap@edcrossref{#2:end}{\xpageref{#2:end}}%
6847       }%
6848     }%
6849     \else%
6850       \printnpnum{%
6851         \wrap@edcrossref{#2:start}{\xpageref{#2:start}}%
6852         \csuse{Xendlinerangeseparator@}\@currentseries%
6853         \wrap@edcrossref{#2:end}{\xpageref{#2:end}}%
6854       }%
6855     }%
6856     \fi%
6857   }%
6858   \def\do##1{%
6859     \setkeys[mac]{falsefootnoteoption}{##1}%
6860   }%

```

```

6860 \notblank{#1}{\docsvlist{#1}}{-}%
6861 }%
6862 %

```

`\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`.

```

6863 \newcommand*{\edmakelabel}[2]{\expandafter\xdef\csname the@label#1\
endcsname{#2}}
6864
6865 %

```

(If you are only going to refer to such a label using `\xxref`, then you can omit entries in the same way as with `\linenum` (see VI.3 p. 139 and V.9 p. 104), since `\xxref` makes a call to `\linenum` in order to do its work.)

XXIV.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`.

```

6866 \pretocmd{\XR@test}%
6867   {\XR@test@mac+++#1#2#3#4+++}%
6868   {}%
6869   {}%
6870 %

```

`\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`.

```

6871 \long\def\xR@test@mac+++#1+++{\XR@test@mac@test#1}
6872 %

```

`\XR@test@mac@test` And finally, `\XR@test@mac@test` does the job. This code is based on the `\XR@test` macro of the `xr` package. However, not that the `\XR@prefix` is not called here, but it is integrated directly in `\l@dmake@labels` and `\l@dmake@labelsR`.

```

6873 \long\def\xR@test@mac@test#1#2...{%The triple dots (NOT \ldots) are because
of the line 22 of xr.sty v5.02 1994/05/28
6874   \ifx#1\l@dmake@labels%

```

```

6875 \l@dmake@labels#2%
6876 \else
6877 \ifx#1\l@dmake@labelsR%
6878 \l@dmake@labelsR #2%
6879 \fi%
6880 \fi%
6881 }%
6882 %

```

XXV Side notes

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.

```

6883 \pretocmd{\@xympar}%
6884 {\ifnumberedpar@
6885 \led@warn@NoMarginpars
6886 \@esphack
6887 \else}%
6888 {}%
6889 {}%
6890
6891 \apptocmd{\@xympar}%
6892 {\fi}%
6893 {}
6894 {}
6895
6896 %

```

We provide side notes 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 side note margin:

left: 0

right: 1

outer: 2

inner: 3

```

6897 \newcount\sidenote@margin
6898 \newcommand*\sidenotemargin}[1]{%
6899   \l@dgetsidenote@margin{#1}%
6900   \ifnum\@l@dtempcntb>\m@ne
6901     \ifledRcol
6902       \global\sidenote@marginR=\@l@dtempcntb
6903     \else
6904       \global\sidenote@margin=\@l@dtempcntb
6905     \fi
6906   \fi}}
6907 \newcommand*\l@dgetsidenote@margin}[1]{%
6908   \def\@tempa{#1}\def\@tempb{left}%
6909   \ifx\@tempa\@tempb
6910     \@l@dtempcntb \z@
6911   \else
6912     \def\@tempb{right}%
6913     \ifx\@tempa\@tempb
6914       \@l@dtempcntb \@ne
6915     \else
6916       \def\@tempb{outer}%
6917       \ifx\@tempa\@tempb
6918         \@l@dtempcntb \tw@
6919       \else
6920         \def\@tempb{inner}%
6921         \ifx\@tempa\@tempb
6922           \@l@dtempcntb \thr@@
6923         \else
6924           \led@warn@BadSidenotemargin
6925           \@l@dtempcntb \m@ne
6926         \fi
6927       \fi
6928     \fi
6929   \fi}
6930 \sidenotemargin{right}
6931
6932 %

```

`\l@dlp@rbox` We need two boxes to store sidenote texts.

```

\l@drp@rbox
6933 \newbox\l@dlp@rbox
6934 \newbox\l@drp@rbox
6935
6936 %

```

`\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
6937 \newdimen\ledlsnotewidth \ledlsnotewidth=\marginparwidth
\ledrsnotesep
6938 \newdimen\ledrsnotewidth \ledrsnotewidth=\marginparwidth
\ledlsnotefontsetup
6939 \newdimen\ledlsnotesep \ledlsnotesep=\linenumsep
\ledrsnotefontsetup

```

```

6940 \newdimen\ledrsnotesep \ledrsnotesep=\linenumsep
6941 \newcommand*\ledlsnotefontsetup{\raggedleft\footnotesize}
6942 \newcommand*\ledrsnotefontsetup{\raggedright\footnotesize}
6943
6944 %

```

`\ledleftnote` `\ledrightnote`, `\ledinnernote`, `\ledouternote` are the user commands for left, right, inner and outer sidenotes. The two last one are just alias for the two first one, depending of the page number. `\ledsidenote{<text>}` is the command for a moveable sidenote.

```

\ledsidenote
6945 \newcommand*\ledleftnote}[1]{\edtext{}{\l@dlsnote{#1}}}
6946 \newcommand*\ledrightnote}[1]{\edtext{}{\l@drsnote{#1}}}
6947 \newcommand*\ledsidenote}[1]{\edtext{}{\l@dcnote{#1}}}%
6948 \newcommand*\ledinnernote}[1]{\edtext{}{\l@disnote{#1}}}%
6949 \newcommand*\ledouternote}[1]{\edtext{}{\l@dosnote{#1}}}%
6950 %

```

`\l@dlsnote` . The ‘footnotes’ for left, right, and moveable sidenotes. The whole scheme is reminiscent of the critical footnotes code.

```

\l@drsnote
\l@dcnote
\l@desnote
\l@disnote
6951 \newif\ifrighnoteup
6952 \righnoteuptrue
6953
6954 \newcommand*\l@dlsnote}[1]{%
6955   \begingroup%
6956   \newcommand{\content}{#1}%
6957   \ifnumberedpar@
6958     \ifledRcol%
6959       \xright@appenditem{\noexpand\l@dlsnote{\expandonce\content}}%
6960         \to\inserts@listR
6961       \global\advance\insert@countR \@ne%
6962     \else%
6963       \xright@appenditem{\noexpand\l@dlsnote{\expandonce\content}}%
6964         \to\inserts@list
6965       \global\advance\insert@count \@ne%
6966     \fi
6967   \fi%
6968   \ignorespaces%
6969   \endgroup%
6970 }%
6971
6972 \newcommand*\l@drsnote}[1]{%
6973   \begingroup%
6974   \newcommand{\content}{#1}%
6975   \ifnumberedpar@
6976     \ifledRcol%
6977       \xright@appenditem{\noexpand\l@drsnote{\expandonce\content}}%
6978         \to\inserts@listR
6979       \global\advance\insert@countR \@ne%

```

```

6980 \else%
6981 \xright@appenditem{\noexpand\vl@drsnote{\expandonce\content}}%
6982 \to\inserts@list
6983 \global\advance\insert@count \@ne%
6984 \fi
6985 \fi\ignorespaces%
6986 \endgroup%
6987 }%
6988
6989 \newcommand*\l@dcsnote}[1]{%
6990 \begingroup%
6991 \newcommand{\content}{#1}%
6992 \ifnumberedpar@
6993 \ifledRcol%
6994 \xright@appenditem{\noexpand\vl@dcsnote{\expandonce\content}}%
6995 \to\inserts@listR
6996 \global\advance\insert@countR \@ne%
6997 \else%
6998 \xright@appenditem{\noexpand\vl@dcsnote{\expandonce\content}}%
6999 \to\inserts@list
7000 \global\advance\insert@count \@ne%
7001 \fi
7002 \fi\ignorespaces%
7003 \endgroup%
7004 }%
7005
7006 \newcommand*\l@disnote}[1]{%
7007 \begingroup%
7008 \newcommand{\content}{#1}%
7009 \ifnumberedpar@%
7010 \ifledRcol%
7011 \xright@appenditem{\noexpand\vl@disnote{\expandonce\content}}%
7012 \to\inserts@listR%
7013 \global\advance\insert@countR \@ne%
7014 \else%
7015 \xright@appenditem{\noexpand\vl@disnote{\expandonce\content}}%
7016 \to\inserts@list%
7017 \global\advance\insert@count \@ne%
7018 \fi%
7019 \fi\ignorespaces%
7020 \endgroup%
7021 }%
7022
7023 \newcommand*\l@dosnote}[1]{%
7024 \begingroup%
7025 \newcommand{\content}{#1}%
7026 \ifnumberedpar@%
7027 \ifledRcol%
7028 \xright@appenditem{\noexpand\vl@dosnote{\expandonce\content}}%
7029 \to\inserts@listR%

```



```

7030     \global\advance\insert@countR \@ne%
7031     \else%
7032       \xright@appenditem{\noexpand\vl@dosnote{\expandonce\content}}%
7033         \to\inserts@list%
7034       \global\advance\insert@count \@ne%
7035     \fi%
7036     \fi\ignorespaces%
7037   \endgroup%
7038 }%
7039
7040 %

```

\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 side notes. We store the content in lists, because we need to loop later on them, in case many sidenote co-exist for the same line. That is there some special test `\vl@disnote` to do, in order to:

- Store the content of `\ledsidenote` to `\l@dcsnotetext` in any cases.
- Store the content of `\rightsidenote` to:
 - `\l@dcsnotetext` if `\ledsidenote` is to be put on right.
 - `\l@dcsnotetext@r` if `\ledsidenote` is to be put on left.
- Store the content of `\leftsidenote` to:
 - `\l@dcsnotetext` if `\ledsidenote` is to be put on left.
 - `\l@dcsnotetext@l` if `\ledsidenote` is to be put on right.

`\vl@disnote` and `\vl@dosnote` just call `\vl@dlsnote` or `\vl@drsnote`, depending of the page.

```

7041 \newcommand*{\vl@dlsnote}[1]{%
7042   \ifledRcol%
7043     \@l@tempcntb=\sidenote@marginR%
7044     \ifnum\@l@tempcntb>\@ne%
7045       \advance\@l@tempcntb by\page@numR%
7046     \fi%
7047   \else%
7048     \@l@tempcntb=\sidenote@margin%
7049     \ifnum\@l@tempcntb>\@ne%
7050       \advance\@l@tempcntb by\page@num%
7051     \fi%
7052   \fi%
7053   \ifodd\@l@tempcntb%
7054     \listgadd{\l@dcsnotetext@l}{#1}%
7055   \else%
7056     \listgadd{\l@dcsnotetext}{#1}%
7057   \fi

```

```

7058 }
7059 \newcommand*{\vl@drsnote}[1]{%
7060   \ifledRcol{%
7061     \@l@tempcntb=\sidenote@marginR%
7062     \ifnum\@l@tempcntb>\@ne%
7063       \advance\@l@tempcntb by\page@numR%
7064     \fi%
7065   \else%
7066     \@l@tempcntb=\sidenote@margin%
7067     \ifnum\@l@tempcntb>\@ne%
7068       \advance\@l@tempcntb by\page@num%
7069     \fi%
7070   \fi%
7071   \ifodd\@l@tempcntb%
7072     \listgadd{\l@dcsnotetext}{#1}%
7073   \else%
7074     \listgadd{\l@dcsnotetext@r}{#1}%
7075   \fi%
7076 }
7077 \newcommand*{\vl@dcsnote}[1]{\listgadd{\l@dcsnotetext}{#1}}
7078
7079 \newcommand{\vl@disnote}[1]{%
7080   \ifledRcol{%
7081     \@tempcnta=\page@numR%
7082   \else%
7083     \@tempcnta=\page@num%
7084   \fi%
7085   \ifodd\@tempcnta% ODD => right page => inner side = left side
7086     \vl@dlsnote{#1}%
7087   \else%
7088     \vl@drsnote{#1}%
7089   \fi%
7090 }%
7091
7092 \newcommand{\vl@dosnote}[1]{%
7093   \ifledRcol{%
7094     \@tempcnta=\page@numR%
7095   \else%
7096     \@tempcnta=\page@num%
7097   \fi%
7098   \ifodd\@tempcnta% ODD => right page => outer side = right side
7099     \vl@drsnote{#1}%
7100   \else%
7101     \vl@dlsnote{#1}%
7102   \fi%
7103 }%
7104
7105 %

```

`\setl@dlp@rbox` `\setl@dlprbox{<lednums>}{<tag>}{<text>}` puts `<text>` into the `\l@dlp@rbox` box.
`\setl@drpr@box`

And similarly for the right side box. It is these boxes that finally get displayed in the margins.

```

7106 \newcommand*\setl@dlp@rbox}[1]{%
7107   \begingroup%
7108     \parindent\z@\hsize=\ledlsnotewidth%
7109     \ledlsnotefontsetup%We kept it outside of the vbox, because can affect
the ragging
7110     \global\setbox\l@dlp@rbox%
7111     \ifleftnoteup%
7112       =\vbox to\z@{\ledlsnotefontsetup\vss #1}}%We put \
ledlsnotefontsetup inside footnote because required for color command. Note
the {} to keep setting local.
7113     \else%
7114       =\vbox to 0.70\baselineskip{\ledlsnotefontsetup\strut#1\vss}}%
7115     \fi%
7116   \endgroup%
7117 }
7118
7119 \newcommand*\setl@drp@rbox}[1]{%
7120   \begingroup%
7121     \parindent\z@\hsize=\ledrsnotewidth%
7122     \ledrsnotefontsetup%We kept it outside of the vbox, because can affect
the ragging
7123     \global\setbox\l@drp@rbox%
7124     \ifrightrightnoteup%
7125       =\vbox to\z@{\ledrsnotefontsetup\vss#1}}%We put \ledrsnotefontsetup
inside footnote because required for color command. Note the {} to keep
setting local.
7126     \else%
7127       =\vbox to0.7\baselineskip{\ledrsnotefontsetup\strut#1\vss}}%
7128     \fi%
7129   \endgroup%
7130 }%
7131 \newif\ifleftnoteup
7132 \leftnoteuptrue
7133 %

```

`\@sidenotesep` This macro is used to separate sidenotes of the same line.

```

7134 \newcommand{\setsidenotesep}[1]{\gdef\@sidenotesep{#1}}
7135 \newcommand{\@sidenotesep}{, }
7136 %

```

`\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 `\affixlin@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.

```

7137 \newcommand*\affixside@note}{%
7138   \prepare@edindex@fornote{\the\page@num|\the\line@num|\the\subline@num|\
the\page@num|\the\line@num|\the\subline@num}}%
7139   \def\sidenotecontent@{%
7140     \numgdef\itemcount@{0}%
7141     \def\do##1{%
7142       \ifnumequal{\itemcount@}{0}%
7143         {%
7144           \appto\sidenotecontent@{##1}}% Not print not separator before
the 1st note
7145         {\appto\sidenotecontent@{\@sidenotesep ##1}%
7146         }%
7147         \numgdef\itemcount@{\itemcount@+\@one}%
7148       }%
7149     \dolistloop{\l@dcnotetext}%
7150     \ifnumgreater{\itemcount@}{1}{\led@err@ManySidenotes}{}%
7151   %

```

And we do the same for left and right notes (not movable).

```

7152   \gdef\@templ@d{%
7153     \gdef\@templ@n{\l@dcnotetext\l@dcnotetext@1\l@dcnotetext@r}%
7154     \ifx\@templ@d\@templ@n \else%
7155       \if@twocolumn%
7156         \if@firstcolumn%
7157           \setl@dlp@rbox{##1}{\sidenotecontent@}%
7158         \else%
7159           \setl@drp@rbox{\sidenotecontent@}%
7160         \fi%
7161       \else%
7162         \@l@dttempcntb=\sidenote@margin%
7163         \ifnum\@l@dttempcntb>\@one%
7164           \advance\@l@dttempcntb by\page@num%
7165         \fi%
7166         \ifodd\@l@dttempcntb%
7167           \setl@drp@rbox{\sidenotecontent@}%
7168           \gdef\sidenotecontent@{%
7169             \numgdef\itemcount@{0}%
7170             \dolistloop{\l@dcnotetext@1}%
7171             \ifnumgreater{\itemcount@}{1}{\led@err@ManyLeftnotes}{}%
7172             \setl@dlp@rbox{\sidenotecontent@}%
7173           \else%
7174             \setl@dlp@rbox{\sidenotecontent@}%
7175             \gdef\sidenotecontent@{%
7176               \numgdef\itemcount@{0}%
7177               \dolistloop{\l@dcnotetext@r}%
7178               \ifnumgreater{\itemcount@}{1}{\led@err@ManyRightnotes}{}%
7179               \setl@drp@rbox{\sidenotecontent@}%
7180             \fi%
7181             \fi%
7182           \fi%

```

```

7183 \advance\@edindex@fornote@\m@ne%
7184 }
7185 %

```

XXVI Minipages and such

We can put footnotes into minipages. The preparatory code has been set up earlier, all that remains is to ensure that it is available inside a minipage box. This requires some alteration to the kernel code, specifically the `\@iiiminipage` and `\endminipage` macros. We will arrange this so that additional series can be easily added.

`\l@dfetbeginmini` These will be the hooks in `\@iiiminipage` and `\endminipage`.
`\l@dfetendmini` They can be extended to handle other things if necessary.

```

7186 \ifnoledgroup@else%
7187 \newcommand*\l@dfetbeginmini}{\ledgrouptrue\l@dedbeginmini\
l@dfambeginmini}
7188 \newcommand*\l@dfetendmini}{%
7189   \IfStrEq{critical-familiar}{\@mpfnpos}%
7190     {\l@dedendmini\l@dfamendmini}%
7191     {%
7192       \IfStrEq{familiar-critical}{\@mpfnpos}%
7193         {\l@dfamendmini\l@dedendmini}%
7194         {\do@feet@custom@order{mp@}{\@mpfnpos}}}%
7195     }%
7196 }%
7197 %

```

`\l@dedbeginmini` These handle the initiation and closure of critical footnotes in a minipage environment.

```

\l@dedendmini
\mp@append@Xnotes
7198 \newcommand*\l@dedbeginmini}{%
7199   \unless\ifnocritical@%
7200     \def\do##1{%
7201       \csletcs{v##1footnote}{mpv##1footnote}%
7202     }%
7203     \dolistloop{\@series}%
7204     \fi%
7205   }
7206 \newcommand*\l@dedendmini}{%
7207   \unless\ifnocritical@%
7208     \ifl@dpairing%
7209       \ifledRcol%
7210         \flush@notesR%
7211       \else%
7212         \flush@notes%
7213       \fi%
7214     \fi
7215     \def\do##1{%
7216       \mp@append@Xnotes{##1}%

```

```

7217 }%
7218 \dolistloop{\@series}%
7219 \fi%
7220 }%
7221 \newcommand{\mp@append@Xnotes}[1]{%
7222 \ifvoid\csuse{mp#1footins}\else%
7223 \ifl@dpairing%
7224 \ifparledgroup%
7225 \ifledRcol%
7226 \dingdef{\parledgroup@beforenotesR}{\parledgroup@beforenotesR+\skip
\@nameuse{mp#1footins}}%
7227 \else%
7228 \dingdef{\parledgroup@beforenotesL}{\parledgroup@beforenotesL+\
skip\@nameuse{mp#1footins}}%
7229 \fi%
7230 \fi%
7231 \fi%
7232 \ifcsstring{series@display#1}{paragraph}{}%
7233 \setbox\@nameuse{mp#1footins}=\vbox{%
7234 \csuse{Xnotefontsize@#1}%
7235 \ifcsdef{Xhsize\csuse{series@display#1}@#1}{%
7236 \hsize\csuse{Xhsize\csuse{series@display#1}@#1}%
7237 }{}%
7238 \noindent\csuse{Xtxtbeforenotes@#1}%
7239 \unvbox\@nameuse{mp#1footins}%
7240 \@parboxrestore%
7241 }%
7242 }%
7243 \csuse{mp#1footgroup}{#1}%
7244 \fi%
7245 }%
7246 %

```

`\l@dfambeginmini` These handle the initiation and closure of familiar footnotes in a minipage environment.

```

\l@dfamendmini
\mp@append@notesX
7247 \newcommand*\l@dfambeginmini}{%
7248 \unless\ifnofamiliar{%
7249 \def\do##1{\csletcs{vfootnote##1}{mpvfootnote##1}}%
7250 \dolistloop{\@series}%
7251 \fi%
7252 }%
7253
7254 \newcommand*\l@dfamendmini}{%
7255 \unless\ifnofamiliar{%
7256 \def\do##1{%
7257 \mp@append@notesX{##1}%
7258 }%
7259 \dolistloop{\@series}%
7260 \fi%
7261 }%

```

```

7262 \newcommand{\mp@append@notesX}[1]{%
7263   \ifvoid\csuse{mpfootins#1}\else%
7264     \ifcsstring{series@displayX#1}{paragraph}{-}{%
7265       \setbox\@nameuse{mpfootins#1}=\vbox{%
7266         \csuse{notefontsizeX@#1}%
7267         \ifcsdef{hsize\csuse{series@display#1}X@#1}{%
7268           \hsize\csuse{hsize\csuse{series@display#1}X@#1}%
7269         }{}%
7270         \noindent\csuse{txtbeforenotesX@#1}%
7271         \unvbox\@nameuse{mpfootins#1}%
7272         \@parboxrestore%
7273       }%
7274     }%
7275     \csuse{mpfootgroup#1}{#1}%
7276   \fi%
7277 }%
7278 %

```

`\@iiiminipage` This is our extended form of the kernel `\@iiiminipage` defined in `ltboxes.dtx`.

```

7279 \patchcmd%
7280 {\@iiiminipage}%
7281 {\let\@footnotetext\@mpfootnotetext}%
7282 {\let\@footnotetext\@mpfootnotetext\l@dfeetbeginmini}%
7283 {}%
7284 {\led@error@fail@patch@\@iiiminipage}%
7285 %

```

`\endminipage` This is our extended form of the kernel `\endminipage` defined in `ltboxes.dtx`.

```

7286 \patchcmd%
7287 {\endminipage}%
7288 {\footnoterule}%
7289 {\footnoterule\l@advance@parledgroup@beforenormalnotes}%
7290 {}%
7291 {\led@error@fail@patch@endminipage}
7292
7293 \patchcmd%
7294 {\endminipage}%
7295 {\@minipagefalse}%
7296 {\l@dfeetendmini\@minipagefalse}%
7297 {}%
7298 {\led@error@fail@patch@endminipage}
7299
7300 %

```

`\l@dunboxmpfoot` `\l@dunboxmpfoot` insert normal footnotes for ledgroup.
`edgroup@beforenormalnotes`

```

7301 \newcommand*\l@dunboxmpfoot{%
7302   \vskip\skip\@mpfootins

```

```

7303 \normalcolor
7304 \footnoterule
7305 \l@advance@parledgroup@beforenormalnotes
7306 \unvbox\@mpfootins%
7307 }
7308 %

```

When using parallel ledgroup, we need to store the vertical space added before footnote, in order to compensate them between left and right pages.

```

7309 \newcommand{\l@advance@parledgroup@beforenormalnotes}{%
7310 \ifparledgroup
7311 \ifl@pairing
7312 \ifledRcol
7313 \dimdef{\parledgroup@beforenotesR}{\parledgroup@beforenotesR+
skip\@mpfootins}
7314 \else
7315 \dimdef{\parledgroup@beforenotesL}{\parledgroup@beforenotesL+
skip\@mpfootins}
7316 \fi
7317 \fi
7318 \fi
7319 }
7320 %

```

`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.

```

7321 \newenvironment{ledgroup}{%
7322 \resetprevpage@num%
7323 \def\@mpfn{mpfootnote}\def\thempfn{\thempfootnote}\c@mpfootnote\z@%
7324 \let\@footnotetext\@mpfootnotetext
7325 \l@dfeetbeginmini%
7326 }{%
7327 \par
7328 \unskip
7329 \ifvoid\@mpfootins\else
7330 \l@dunboxmpfoot
7331 \fi
7332 \l@dfeetendmini%
7333 \@ledgroupfalse%
7334 }
7335 }
7336
7337
7338 %

```

`ledgroupsize` `\begin{ledgroupsize} [pos] {width}`

This environment puts footnotes at the end, even if that happens to be in the middle of

a page, or crossing a page boundary. It is a sort of unboxed, variable $\langle width \rangle$ minipage. The optional $\langle pos \rangle$ controls the sideways position of numbered text.

```
7339 \newenvironment{ledgroupsize}[2][1]{%
7340 %
```

Set the various text measures.

```
7341 \hsize #2\relax
7342 %
```

Initialize fills for centering.

```
7343 \let\ledllfill\hfil
7344 \let\ledrlfill\hfil
7345 \def\@tempa{#1}\def\@tempb{1}%
7346 %
```

Left adjusted numbered lines

```
7347 \ifx\@tempa\@tempb
7348 \let\ledllfill\relax
7349 \else
7350 \def\@tempb{r}%
7351 \ifx\@tempa\@tempb
7352 %
```

Right adjusted numbered lines

```
7353 \let\ledrlfill\relax
7354 \fi
7355 \fi
7356 %
```

Set up the footnoting.

```
7357 \def\@mpfn{mpfootnote}\def\thempfn{\thempfootnote}\c@mpfootnote\z@
7358 \let\@footnotetext\@mpfootnotetext
7359 \l@dfetbeginmini%
7360 }{%
7361 \par
7362 \unskip
7363 \ifvoid\@mpfootins\else
7364 \l@dunboxmpfoot
7365 \fi
7366 \l@dfetendmini%
7367 }
7368 %
7369 %
```

Close the `\ifnoledgroup@` else.

```
7370 \fi%
7371 %
```

`\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`.

```
7372 \newif\ifledgroupnotesL@
7373 \newif\ifledgroupnotesR@
7374 %
```

XXVII Indexing

Here is some code for indexing using page and line numbers.

XXVII.1 Looking on package order

First, ensure that `imakeidx` or `indextools` is loaded *before* `eledmac`.

```
7375 \AtBeginDocument{%
7376   \unless\ifl@imakeidx%
7377     \ifpackageloaded{imakeidx}{\led@error@PackageAfterEledmac{imakeidx}}{}
7378   %
7379   \fi%
7380   \unless\ifl@indextools%
7381     \ifpackageloaded{indextools}{\led@error@PackageAfterEledmac{indextools}}{}
7382   %
7383   \fi%
7384   \unless\ifl@footmisc%
7385     \ifpackageloaded{footmisc}{\led@error@PackageAfterEledmac{footmisc}}{}
7386   %
7387   \fi%
7388 }
7389 %
```

XXVII.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
7387 \newcommand{\pagelinesep}{-}
7388 \newcommand{\edindexlab}{$&}
7389 \newcounter{labidx}
7390 \setcounter{labidx}{0}
7391 %
7392 %
```

`\doedindexlabel` This macro sets an `\edlabel`.

```
7393 \newcommand{\doedindexlabel}{%
7394   \stepcounter{labidx}%
7395   \edlabel{\edindexlab\thelabidx}%
7396 }
```

```
7397
7398 %
```

`\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`.

```
7399 \newcounter{pageline}%
7400 \renewcommand{\thepageline}{%
7401   \thepage%
7402   \pagelinesep%
7403   \xlineref{\edindexlab\thelabidx}%
7404 }
7405 %
```

`\thestartpageline` `\theendpageline` These macros make up the page/line start/end number when the `\edindex` command is called in critical notes.

```
7406 \newcommand{\thestartpageline}{%
7407   \l@dparsedstartpage%
7408   \pagelinesep%
7409   \l@dparsedstartline%
7410 }
7411 \newcommand{\theendpageline}{%
7412   \l@dparsedendpage%
7413   \pagelinesep%
7414   \l@dparsedendline%
7415 }
7416 %
```

XXVII.3 Code specific to `\edindex` in critical footnotes

`\@edindex@fornote@` This counter is incremented at the beginning of each note (either a footnote or a sidenote), and decremented at the end of each note. If its value is greater than 0, that means we are inside a note.

```
7417 \newcount\@edindex@fornote@
7418 %
```

`\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 `|`.

```
7419 \newcommand{\prepare@edindex@fornote}[1]{%
7420   \l@dp@rsefootspec#1|}%
7421   \advance\@edindex@fornote@\@ne%
7422 }
7423 %
```

`\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.

```

7424 \newcommand{\get@edindex@ledinnote@command}{%
7425   \ifxindy%
7426     \gdef\@ledinnote@command{%
7427       ledinnote\thelabidx%
7428     }%
7429     \ifxindyhyperref%
7430       \immediate\write\eledmac@xindy@out{%
7431         (define-attributes ("ledinnote\thelabidx"))^^J
7432         \space\space(markup-locref^^J
7433           \eledmacmarkuplocrefdepth^^J
7434           :open "\string\ledinnote[\edindexlab\thelabidx]{\@index@command
7435         }{"^^J
7436           :close "}"^^J
7437           :attr "ledinnote\thelabidx"^^J
7438         )
7439       }%
7440     \else%
7441       \immediate\write\eledmac@xindy@out{%
7442         (define-attributes ("ledinnote\thelabidx"))^^J
7443         \space\space(markup-locref^^J
7444           \eledmacmarkuplocrefdepth^^J
7445           :open "\string\ledinnote{\@index@command}{"^^J
7446           :close "}"^^J
7447           :attr "ledinnote\thelabidx"^^J
7448         )
7449       }%
7450     \fi%
7451   }%

```

If we do not use `xindy` option, `\@ledinnote@command` will produce something like `ledinnote{formattingcommand}`.

```

7451   \else%
7452     \gdef\@ledinnote@command{%
7453       ledinnote[\edindexlab\thelabidx]{\@index@command}%
7454     }%
7455   \fi%
7456 }
7457 %

```

XXVII.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 |.

```

7458 \def\get@index@command#1|#2+{%

```

```

7459 \gdef\@index@txt{#1}%
7460 \gdef\@index@command{#2}%
7461 \xdef\@index@parenthesis{}%
7462 \IfBeginWith{\@index@command}{(}{%
7463   \StrGobbleLeft{\@index@command}{1}[\@index@command@]%
7464   \global\let\@index@command\@index@command@%
7465   \xdef\@index@parenthesis{(%}%
7466   }{)%%
7467 \IfBeginWith{\@index@command}{)}{%
7468   \StrGobbleLeft{\@index@command}{1}[\@index@command@]%
7469   \global\let\@index@command\@index@command@%
7470   \xdef\@index@parenthesis{)%}%
7471   }{)%%
7472 }
7473 %

```

XXVII.5 Code for the formatted index

`\ledinnote` These macros are used to specify that an index reference points to a note. Arguments of `\ledinnote` are: #1 (optional): the label for the hyperlink, #2: command applied to the number, #3: the number itself.

`\ledinnotehyperpage`

`\ledinnotemark`

```

7474 \newcommand{\ledinnote}[3][1,usedefault]{%
7475   \ifboolexpr{%
7476     test{\ifdefequal{\iftrue}{\ifHy@hyperindex}}%
7477     or%
7478     bool {xindyhyperref@}%
7479   }%
7480   {%
7481     \csuse{#2}{\hyperlink{#1}{\ledinnotemark{#3}}}%
7482   }%
7483   {%
7484     \csuse{#2}{\ledinnotemark{#3}}%
7485   }%
7486 }%
7487 \newcommand{\ledinnotehyperpage}[2]{\csuse{#1}{\ledinnotemark{\hyperpage
7488 {#2}}}}%
7489 \newcommand{\ledinnotemark}[1]{#1\emph{n}}%
7490 %

```

XXVII.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
7490 \newcommandx{\@wredindex}[2][1=\expandonce\jobname,usedefault]{%#1 = the
index name, #2 = the text
7491 \begingroup%
7492 \let\emph\@firstofone%
7493 \let\textbf\@firstofone%
7494 \let\textit\@firstofone%
7495 \let\textmd\@firstofone%
7496 \let\textnormal\@firstofone%
7497 \let\textrm\@firstofone%
7498 \let\textsc\@firstofone%
7499 \let\textsf\@firstofone%
7500 \let\textsl\@firstofone%
7501 \let\texttt\@firstofone%
7502 \let\textup\@firstofone%
7503 \xdef\@tmp{#2}%To be used in IfSubStr instead of #2 directly. Avoid
some expansion bugs (for example with \edindex{textsc{something}})
7504 \endgroup%
7505 \ifl@imakeidx%
7506 \ifnum\@edindex@fornote@>\z%
7507 \IfSubStr[1]{\@tmp}{|}{\get@index@command#2+}{\get@index@command#2|+}
%
7508 \get@edindex@ledinnote@command%
7509 \expandafter\imki@wrindexentry{#1}{\@index@txt|(\@ledinnote@command
}{\thestartpageline}%
7510 \expandafter\imki@wrindexentry{#1}{\@index@txt|)\@ledinnote@command
}{\theendpageline}%
7511 \else%
7512 \get@edindex@hyperref{#2}%
7513 \imki@wrindexentry{#1}{\@index@txt\@edindex@hyperref}{\thepageline}%
7514 \fi%
7515 \else%
7516 \ifnum\@edindex@fornote@>\z%
7517 \IfSubStr[1]{\@tmp}{|}{\get@index@command#2+}{\get@index@command#2|+}
%
7518 \get@edindex@ledinnote@command%
7519 \expandafter\protected@write\@indexfile{}%
7520 {\string\indexentry{\@index@txt|(\@ledinnote@command}{\thestartpageline}
}%
7521 }%
7522 \expandafter\protected@write\@indexfile{}%
7523 {\string\indexentry{\@index@txt|)\@ledinnote@command}{\theendpageline}
}%
7524 }%
7525 \else%
7526 \protected@write\@indexfile{}%
7527 {\string\indexentry{#2}{\thepageline}

```

```

7528 }%
7529 \fi%
7530 \fi%
7531 \endgroup
7532 \@esphack%
7533 }
7534 %

```

Need to add the definition of `\edindex` to `\makeindex`, and initialise `\edindex` to do nothing.

```

7535 \pretocmd{\makeindex}{%
7536 \def\edindex{%
7537 \ifboolexpr{bool{numbering} or bool{numberingR} or bool{
7538 l@dprintingpages} or bool{l@dprintingcolumns}}{%
7539 \@bsphack%
7540 \doedindexlabel%
7541 \begingroup%
7542 \@sanitize%
7543 \@wredindex%
7544 }%
7545 }%
7546 \led@warn@edinde@outsidenumbering%
7547 \index%
7548 }%
7549 }%
7550 {}%
7551 {\led@error@fail@patch@makeindex}%
7552 \newcommand{\edindex}[1]{\@bsphack\@esphack}
7553 \newcommandx{\dummy@edindex}[2][1=\expandonce\jobname,usedefault]{}%
7554 %

```

XXVII.7 *Hyperlink*

`\hyperlinkformat` `\hyperlinkformat` command is to be used to have both a internal hyperlink and a format, when indexing.

```

7555 \newcommand{\hyperlinkformat}[3]{%
7556 \ifstrempy{#1}%
7557 {\hyperlink{#2}{#3}}%
7558 {\csuse{#1}{\hyperlink{#2}{#3}}%
7559 }}
7560 %

```

`\hyperlinkR` `\hyperlinkR` command is to be used to create a internal hyperlink and `\ledRflag`, when indexing.

```

7561 \newcommand{\hyperlinkR}[2]{%
7562 \hyperlink{#1}{#2\@Rlineflag}%

```

```
7563 }%
7564
7565 %
```

`\hyperlinkformatR` `\hyperlinkformatR` command is to be used to create a internal hyperlink, a format and a `\@Rlineflag`, when indexing.

```
7566 \newcommand{\hyperlinkformatR}[3]{%
7567   \hyperlinkformat{#1}{#2}{#3\@Rlineflag}%
7568 }%
7569
7570 %
```

`\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`).

```
7571 \newcommand{\get@edindex@hyperref}[1]{%
7572 %
```

We have to disable temporary spaces to work through a `xstring` bug (or feature?)

```
7573 \edef\temp@{%
7574 \catcode`\ =9 %space need for catcode
7575 \detokenize{#1}%For active character in unicode
7576 \catcode`\ =10 % space need for catcode
7577 }%
7578 %
```

Now, we define `\@edindex@hyperref` if the `hyperindex` of `hyperref` is enabled.

```
7579 \ifdefequal{\iftrue}{\ifHy@hyperindex}{%
7580   \IfSubStr{\temp@}{|}%
7581   {\get@index@command#1+%
7582   \ifledRcol%
7583     \gdef\@edindex@hyperref{|\@index@parenthesis %space kept
7584     hyperlinkformatR{\@index@command}%
7585     {\edindexlab\thelabidx}}%
7586   \else%
7587     \gdef\@edindex@hyperref{|\@index@parenthesis %space kept
7588     hyperlinkformat{\@index@command}%
7589     {\edindexlab\thelabidx}}%
7590   \fi%
7591 }%
7592 {\get@index@command#1|+%
7593   \ifledRcol%
7594     \gdef\@edindex@hyperref{hyperlinkR{\edindexlab\thelabidx}}%
7595   \else%
7596     \gdef\@edindex@hyperref{hyperlink{\edindexlab\thelabidx}}%
7597   \fi%
7598 }%
7599 }%
7600 %
```



```

7601 % If we use both xindy and hyperref, first get the \protect\cs{
index@command} command.
7602 % Then define \protect\cs{@edindex@hyperref} in the form \verb+eledmacXXX+
7603 % \begin{macrocode}
7604 {\ifxindyhyperref%
7605 \IfSubStr{\temp@}{|}%
7606 {\get@index@command#1+}%
7607 {\get@index@command#1|+}%
7608 \gdef\@edindex@hyperref{|eledmac\thelabidx}%
7609 %

```

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`.

```

7610 \IfStrEq{\@index@parenthesis}{(}%
7611 {%
7612 \csxdef{xindyparenthesis@\@index@txt}{\thelabidx}%
7613 \gdef\@edindex@hyperref{| (eledmac\thelabidx}%
7614 }%
7615 {}%
7616 %

```

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`.

```

7617 \IfStrEq{\@index@parenthesis}{)}%
7618 {%
7619 \xdef\@edindex@hyperref{|)eledmac\csuse{xindyparenthesis@\@index@txt}}%
7620 \global\csundef{xindyparenthesis@\@index@txt}%
7621 }%
7622 %

```

Write in the `.xdy` file the attributes of the location.

```

7623 {%
7624 \immediate\write\eledmac@xindy@out{%
7625 (define-attributes ("eledmac\thelabidx"))^^J
7626 \space\space(markup-locref^^J
7627 \eledmacmarkuplocrefdepth^^J
7628 :open "\string\hyperlink%
7629 \ifledRcol R\fi%
7630 {\edindexlab\thelabidx}%
7631 {\ifdefempty{\@index@command}%
7632 }%
7633 {\@backslashchar\@index@command}%
7634 {"^^J
7635 :close "}")^^J
7636 :attr "eledmac\thelabidx"^^J
7637 )

```

```

7638     }%
7639     }%
7640 %

```

And now, in any other case.

```

7641 \else%
7642   \gdef\@index@txt{#1}%
7643   \gdef\@edindex@hyperref{}%
7644   \fi%
7645 }%
7646 }
7647 %

```

XXVII.8 ‘innote’ and ‘notenumber’ option of indextols package

`\led@set@index@fornote` The `\led@set@index@fornote` is called when a familiar footnote is inserted — and not when it is read — and changes the `\index` command depending of the option of the `indextools` package. Its only argument is the note series.

```

7648 \newcommand{\led@set@index@fornote}[1]{%
7649   \ifbool{indtl@innote}%
7650     {\let\index\nindex}%
7651     {}%
7652   \ifbool{indtl@notenumber}%
7653     {%
7654     \renewcommand{\index}[2][\indtl@jobname]{%
7655       \orig@@index[##1]{%
7656         ##2|innotenumber{\this@footnoteX@reading}%
7657       }%
7658     }%
7659   }%
7660   {}%
7661 }%
7662 %

```

`\led@reinit@index@fornote` The `\led@reinit@index@fornote` just reset the default value of `\index`.

```

7663 \newcommand{\led@reinit@index@fornote}{%
7664   \ifbool{indtl@innote}%
7665     {\let\index\orig@@index}%
7666     {}%
7667   \ifbool{indtl@notenumber}%
7668     {\let\index\orig@@index}%
7669     {}%
7670 }%
7671 %

```

XXVIII Glossaries

Here, we define the \gls-like commands prefixed by ed, only if the package glossaries is loaded.

```
7672 \AtBeginDocument{%
7673   \@ifpackageloaded{glossaries}{%
7674     %
```

First those which arguments are [*options*]{*label*}[*insert*].

```
7675   \renewcommand{\do}[1]{%
7676     \expandafter\DeclareRobustCommandx\csname ed#1\endcsname[3][1,3,
usedefault]{%
7677       \doedindexlabel%
7678       \csname#1\endcsname[counter=pageline,##1]{##2}[##3]%
7679     }%
7680   \expandafter\WithSuffix\expandafter\DeclareRobustCommandx\csname ed
#1\endcsname*[3][1,3,usedefault]{%
7681     \doedindexlabel%
7682     \csname#1\endcsname*[counter=pageline,##1]{##2}[##3]%
7683   }%
7684 }%
7685 \docsvlist{%
7686   gls,%
7687   Gls,%
7688   GLS,%
7689   glspl,%
7690   Glspl,%
7691   GLSpl,%
7692   glstext,%
7693   Glstext,%
7694   GLStext,%
7695   Glsfirst,%
7696   GLSfirst,%
7697   glsplural,%
7698   Glsplural,%
7699   GLSplural,%
7700   glsfirstplural,%
7701   Glsfirstplural,%
7702   GLSfirstplural,%
7703   glsname,%
7704   Glsname,%
7705   GLSname,%
7706   glssymbol,%
7707   Glsymbol,%
7708   GLSsymbol,%
7709   glsdesc,%
7710   Glsdesc,%
7711   GLSdesc,%
7712   glsuseri,%
```

```

7713 Glsuseri,%
7714 GLSuseri,%
7715 glsuserii,%
7716 Glsuserii,%
7717 GLSuserii,%
7718 glsuseriii,%
7719 Glsuseriii,%
7720 GLSuseriii,%
7721 glsuseriv,%
7722 Glsuseriv,%
7723 GLSuseriv,%
7724 glsuserv,%
7725 Glsuserv,%
7726 GLSuserv,%
7727 glsuservi,%
7728 Glsuservi,%
7729 GLSuservi%
7730 }%
7731 %

```

Then those which arguments are [*options*]{*label*}{*link text*}.

```

7732 \renewcommand{\do}[1]{%
7733 \expandafter\DeclareRobustCommand\csname ed#1\endcsname[3][1,
usedefault]{%
7734 \doedindexlabel%
7735 \csname#1\endcsname[counter=pageline,##1]{##2}{##3}%
7736 }%
7737 \expandafter\WithSuffix\expandafter\DeclareRobustCommand\csname ed
#1\endcsname*[3][1,usedefault]{%
7738 \doedindexlabel%
7739 \csname#1\endcsname*[counter=pageline,##1]{##2}{##3}%
7740 }%
7741 }%
7742 \docsvlist{glsdisp,glslink}%
7743 %

```

Then those which arguments are [*options*]{*label*}.

```

7744 \renewcommand{\do}[1]{%
7745 \expandafter\DeclareRobustCommand\csname ed#1\endcsname[2][1,
usedefault]{%
7746 \doedindexlabel%
7747 \csname#1\endcsname[counter=pageline,##1]{##2}%
7748 }%
7749 \expandafter\WithSuffix\expandafter\DeclareRobustCommand\csname ed
#1\endcsname*[2][1,usedefault]{%
7750 \doedindexlabel%
7751 \csname#1\endcsname*[counter=pageline,##1]{##2}%
7752 }%
7753 }%
7754 \docsvlist{glsadd}%

```

```

7755 }{}%
7756 }%
7757 %

```

XXIX 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`.

XXIX.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 the next code:

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

The `\ifinstanza` boolean is used to be sure that we are in a stanza part.

```

7758 \def\@hangingsymbol{}
7759 \newcommand*\sethangingsymbol[1]{%
7760   \gdef\@hangingsymbol{#1}%
7761 }%
7762 \newif\ifinstanza
7763 %

```

`\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.

```

7764 \newif\ifinserthangingsymbol
7765 \newcommand\inserthangingsymbol{%
7766   \ifinserthangingsymbol%
7767     \ifinstanza%
7768       \@hangingsymbol%
7769     \fi%
7770 \fi%
7771 }
7772 %

```

XXIX.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.

```

7773 \newcommand*{\ampersand}{\char`\&}
7774
7775 %

```

XXIX.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.

```

7776 \chardef\body=\catcode`\@
7777 \catcode`\@=11
7778 \chardef\next=\catcode`\&
7779 \catcode`\&=\active
7780
7781 %

```

XXIX.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.

```

7782 \newcount\stanza@count
7783 \newlength{\stanzaindentbase}
7784 \setlength{\stanzaindentbase}{20pt}
7785
7786 %

```

`\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.

```

7787 \def\strip@szacnt#1,#2|{\def\@tempb{#1}\def\@tempa{#2|}}
7788 \newcommand*{\setstanzavalues}[2]{\def\@tempa{#2, ,|}%
7789 \stanza@count\z@
7790 \def\next{\expandafter\strip@szacnt\@tempa
7791 \ifx\@tempb\empty\let\next\relax\else
7792 \expandafter\mathchardef\csname #1@number\stanza@count
7793 \@endcsname\@tempb\relax
7794 \advance\stanza@count\@ne\fi\next}%
7795 \next}
7796
7797 %

```

`\setstanzaindents` In the original edmac, `\setstanzavalues{sza}{⟨...⟩}` had to be called to set the in-
`\setstanzapenalties` dents, and similarly `\setstanzavalues{szp}{⟨...⟩}`. to set the penalties. `\setstanzaindents`
and `\setstanzapenalties` macros are a convenience to give the user one less thing
to worry about (misspelling the first argument).

```
7798 \newcommand*\setstanzaindents[1]{\setstanzavalues{sza}{#1}}
7799 \newcommand*\setstanzapenalties[1]{\setstanzavalues{szp}{#1}}
7800 %
7801 %
```

`\managestanza@modulo` Since version 0.13, the `stanzaindentsrepetition` counter can be used when the in-
dentation is repeated every *n* verses. The `\managestanza@modulo` is a command which
modifies the counter `stanza@modulo`. The command adds 1 to `stanza@modulo`, but if
`stanza@modulo` is equal to the `stanzaindentsrepetition` counter, the command restarts
it.

```
7802 \newcounter{stanzaindentsrepetition}
7803 \newcount\stanza@modulo
7804
7805 \newcommand*\managestanza@modulo[0]{%
7806   \advance\stanza@modulo\@ne%
7807   \ifnum\stanza@modulo>\value{stanzaindentsrepetition}%
7808     \stanza@modulo\@ne%
7809   \fi%
7810 }
7811 %
```

`\stanzaindent` The macro `\stanzaindent`, when called at the beginning of a verse, changes the in-
`\stanzaindent*` dentation normally defined for this verse by `\setstanzaindent`. The starred version
skips the current verse for the repetition of stanza indent.

```
7812 \newcommand{\stanzaindent}[1]{%
7813   \hspace{\dimexpr#1\stanzaindentbase-\parindent\relax}%
7814   \ignorespaces%
7815 }%
7816 \WithSuffix\newcommand\stanzaindent*[1]{%
7817   \stanzaindent{#1}%
7818   \global\advance\stanza@modulo-\@ne%
7819   \ifnum\stanza@modulo=0%
7820     \global\stanza@modulo=\value{stanzaindentsrepetition}%
7821   \fi%
7822   \ignorespaces%
7823 }%
7824 %
```

XXIX.5 Numbering stanza

Here, macro for numbering stanza. First, the stanza counter.

```

\thestanzaa25 \newcounter{stanza}
7826 \renewcommand{\thestanza}{%
7827 \textbf{\arabic{stanza}}%
7828 }
7829 %

```

\ifnumberstanza Then, macro to activate automatically numbering of stanza.

```

7830 \newif\ifnumberstanza%
7831 %

```

\@insertstanzanumber Now, macro called at the first line of of verse of a stanza.

```

7832 \newcommand{\@insertstanzanumber}[0]{%
7833 \ifnumberstanza%
7834 \ifl@pairing%
7835 \ifledRcol%
7836 \stanzanumwrapper{\thestanzaR}%
7837 \else%
7838 \stanzanumwrapper{\thestanzaL}%
7839 \fi%
7840 \else%
7841 \stanzanumwrapper{\thestanza}%
7842 \fi%
7843 \setline{1}%
7844 \fi%
7845 }%
7846 %

```

\@advancestanzanumber Also a command to advance the counter of stanza.

```

7847 \newcommand{\@advancestanzanumber}[0]{%
7848 \ifnumberstanza%
7849 \ifl@pairing%
7850 \ifledRcol%
7851 \addtocounter{stanzaR}{1}%
7852 \else%
7853 \addtocounter{stanzaL}{1}%
7854 \fi%
7855 \else%
7856 \addtocounter{stanza}{1}%
7857 \fi%
7858 \fi%
7859 }%
7860 %

```

\stanzanumwrapper And finally, the wrapper for stanza number


```

7861 \newcommand{\stanzanumwrapper}[1]{%
7862   \flagstanza{#1}%
7863 }%
7864 %

```

XXIX.6 Stanza number in note

Here, the command called when printing stanza number in notes.

```

7865 \newcommand{\printstanza}[0]{%
7866   \ifboolexpr{bool{l@dpairing} or bool{l@dprintingpages} or bool{
7867     l@dprintingcolumns}}{%
7868     \ifledRcol{%
7869       \thestanzaR%
7870     }%
7871     \ifl{%
7872       \thestanzaL%
7873     }%
7874     \thestanza%
7875   }%
7876 %

```

XXIX.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. `\sza@penalty`

If it is known that each stanza line will fit on one print line, it is advisable to set the hanging indentation to zero. `\sza@penalty` places the specified penalty following each stanza line. By default, this facility is turned off so that no penalty is included. However, the user may initiate these penalties to indicate good and bad places in the stanza for page breaking.

```

7877 \newcommandx{\stanza@line}[2][1,2,usedefault]{%
7878   \ifnum\value{stanzaindentsrepetition}=0
7879     \ifcsdef{sza@number\stanza@count @}{%
7880       {%
7881         \parindent=\csname sza@number\stanza@count @\endcsname\
7882         stanzaindentbase%
7883       }%
7884       \led@err@StanzaIndentNotDefined%
7885     }%
7886   \else
7887     \ifcsdef{sza@number\stanza@modulo @}{%
7888       \parindent=\csname sza@number\stanza@modulo @\endcsname\
7889       stanzaindentbase%
7890       \managestanza@modulo%
7891     }%

```

```

7890     {%
7891     \led@err@StanzaIndentNotDefined%
7892     }%
7893     \fi
7894     \pstart[#1][#2]\stanza@hang\ignorespaces%
7895 }%
7896 \xdef\stanza@hang{\noexpand\leavevmode\noexpand\startlock
7897     \hangindent\expandafter
7898     \noexpand\csname sza@0@\endcsname\stanzaindentbase
7899     \hangafter\@ne}
7900 \def\sza@penalty{\count@\csname szp@\number\stanza@count @\endcsname
7901     \ifnum\count@>\@M\advance\count@-\@M\penalty-\else
7902     \penalty\fi\count@}
7903 %

```

\@startstanza Now we have the components of the `\stanza` macro, which appears at the start of a group of lines. This macro initializes the count and checks to see if hanging indentation and penalties are to be included. Hanging indentation suspends the line count, so that the enumeration is by verse line rather than by print line. If the print line count is desired, invoke `\let\startlock\relax` and do the same for `\endlock`. Here and above we have used `\xdef` to make the stored macros take up a bit less space, but it also makes them more obscure to the reader. Lines of the stanza are delimited by ampersands `&`. The last line of the stanza must end with `\&`.

`\stanza`
`\@stopstanza`
`\AtEveryStopStanza`
`\AtEveryStanza`
`\AtStartEveryStanza`
`\BeforeEveryStopStanza`
`\newverse`

```

7904 \xdef\@startstanza[#1][#2]{%
7905     \noexpand\instanzatrue\expandafter
7906     \begingroup%
7907     \catcode`\noexpand\&\active%
7908     \global\stanza@count@\ne\stanza@modulo@\ne
7909     \noexpand\ifnum\expandafter\noexpand
7910     \csname sza@0@\endcsname=\z@\let\noexpand\stanza@hang\relax
7911     \let\noexpand\endlock\relax\noexpand\else\interlinepenalty
7912     \@M\rightskip\z@ plus 1fil\relax\noexpand\fi\noexpand\ifnum
7913     \expandafter\noexpand\csname szp@0@\endcsname=\z@
7914     \let\noexpand\sza@penalty\relax\noexpand\fi%
7915     \def\noexpand&{%
7916         \noexpand\newverse [] []}%
7917     \def\noexpand\&{\noexpand\@stopstanza}%
7918     \noexpand\@advancestanzanumber%
7919     \noexpand\stanza@line[#1][#2]%
7920     \noexpand\@insertstanzanumber%
7921     \let\par\relax\ignorespaces%No paragraph in verses
7922 }
7923
7924 \newcommandx{\stanza}[2][1,2,usedefault]{%
7925     \ifboolexpr{%
7926         not test{\ifdefvoid{\at@every@stanza}}%
7927         and test{\ifstrempy{#1}}%
7928         and test{\ifstrempy{#2}}}%

```

```

7929   {\@startstanza [] [\at@every@stanza]\at@start@every@stanza}%
7930   {\@startstanza [#1] [#2] \at@start@every@stanza}%
7931 }%
7932
7933 \newcommandx{\@stopstanza} [2] [1,2,usedefault] {%
7934   \unskip%
7935   \endlock%
7936   \ifboolexpr{%
7937     not test{\ifdefvoid{\at@every@stop@stanza}}%
7938     and test{\ifstrempy{#1}}%
7939     and test{\ifstrempy{#2}}}%
7940     {\before@every@stop@stanza\pend [] [\at@every@stop@stanza]}%
7941     {\before@every@stop@stanza\pend [#1] [#2]}%
7942   \endgroup%
7943   \instanzafalse%
7944 }
7945
7946 \newcommand{\AtEveryStopStanza} [1] {%
7947   \ifstrempy{#1}%
7948     {\gdef\at@every@stop@stanza{}}%
7949     {\gdef\at@every@stop@stanza{\noindent#1}}%
7950 }%
7951 \WithSuffix\newcommand\AtEveryStopStanza* [1] {%
7952   \ifstrempy{#1}%
7953     {\gdef\at@every@stop@stanza{}}%
7954     {\gdef\at@every@stop@stanza{#1}}%
7955 }%
7956 \def\at@every@stop@stanza{}%
7957
7958 \newcommand{\AtEveryStanza} [1] {%
7959   \ifstrempy{#1}%
7960     {\gdef\at@every@stanza{}}%
7961     {\gdef\at@every@stanza{\noindent#1}}%
7962 }%
7963 \WithSuffix\newcommand\AtEveryStanza* [1] {%
7964   \ifstrempy{#1}%
7965     {\gdef\at@every@stanza{}}%
7966     {\gdef\at@every@stanza{#1}}%
7967 }%
7968
7969
7970
7971 \newcommand{\AtStartEveryStanza} [1] {%
7972   \ifstrempy{#1}%
7973     {\gdef\at@start@every@stanza{}}%
7974     {\gdef\at@start@every@stanza{#1}}%
7975 }%
7976 \def\at@start@every@stanza{}%
7977
7978 \newcommand{\BeforeEveryStopStanza} [1] {%

```

```

7979 \ifstrempy{#1}%
7980   {\gdef\before@every@stop@stanza{}}%
7981   {\gdef\before@every@stop@stanza{#1}}%
7982 }%
7983 \def\before@every@stop@stanza{}%
7984
7985 \newcommand*{\newverse}[4][1,2,3,4,usedefault]{%
7986   \unskip%
7987   \endlock\pend[#1][#3]\sza@penalty\global%
7988   \advance\stanza@count\@ne\stanza@line[#2][#4]%
7989   }
7990
7991 %

```

\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`.

```

7992 \newcommand*{\flagstanza}[2][\stanzaindentbase]{%
7993   \hskip -#1\llap{#2}\hskip #1\ignorespaces}
7994
7995 %

```

XXIX.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.

```

7996 \catcode`\&=\next
7997 \catcode`\@=\body
7998 \setstanzavalues{szp}{0}
7999
8000 %

```

XXX Apparatus of Manuscripts

XXX.1 User level macro

\msdata The user level `\msdata` command only writes the manuscripts data in a numbered auxiliary file. There are two associated etoolbox counters.

```

\msdata@c
\msdata@cR
8001 \def\msdata@c{}%
8002 \def\msdata@cR{}%
8003 \newcommand{\msdata}[1]{%
8004   \leavevmode%

```

```

8005 \unless\ifstopmsdata@inserted%
8006 \stopmsdata%
8007 \led@warning@msdatawithoutstop%
8008 \fi%
8009 \global\stopmsdata@inserted@false%
8010 \unless\ifledRcol%
8011 \numgdef{\msdata@c}{\msdata@c+1}%
8012 \ifdef{\hypertarget}{%
8013 \edlabel{\msdata@c:start:msdata}%
8014 }{}%
8015 \protected@write\linenum@out{}{%
8016 \string\@msd{#1}%
8017 }%
8018 \else%
8019 \numgdef{\msdata@cR}{\msdata@cR+1}%
8020 \ifdef{\hypertarget}{%
8021 \edlabel{\msdata@cR:start:msdata}%
8022 }{}%
8023 \protected@write\linenum@outR{}{%
8024 \string\@msd{#1}%
8025 }%
8026 \fi%
8027 }%
8028 %

```

`\stopmsdata` The user level `\stopmsdata` command only writes information about the end of manuscripts data in numbered auxiliary file.

```

8029 \newcommand{\stopmsdata}[0]{%
8030 \leavevmode%
8031 \unless\ifledRcol%
8032 \protected@write\linenum@out{}{%
8033 \string\@stopmsd%
8034 }%
8035 \ifdef{\hypertarget}{%
8036 \edlabel{\msdata@c:end:msdata}%
8037 }{}%
8038 \else%
8039 \protected@write\linenum@outR{}{%
8040 \string\@stopmsd%
8041 }%
8042 \ifdef{\hypertarget}{%
8043 \edlabel{\msdata@cR:end:msdata}%
8044 }{}%
8045 \fi%
8046 \global\stopmsdata@inserted@true%
8047 }%
8048 %

```

`\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`

```
8049 \newif\ifstopmsdata@inserted@%
8050 %
```

XXX.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?

```
8051 \newcommand{\setmsdataseries}[1]{%
8052   \gdef\@msdata@series{#1}%
8053 }%
8054 \def\@msdata@series{A}%
8055 %
```

`\setmsdataposition` The label for the manuscripts data.

```
8056 \def\ms@data@position{msdata-regular}%
8057 \newcommand{\setmsdataposition}[1]{%
8058   \gdef\ms@data@position{#1}%
8059 }%
8060 %
```

`\setmsdatalabel` The label for the manuscripts data.

```
8061 \def\ms@data@label{Ms.}%
8062 \newcommand{\setmsdatalabel}[1]{%
8063   \gdef\ms@data@label{#1}%
8064 }%
8065 %
```

XXX.3 Counters and lists

`\@msd@c` `\@msd@c` is a counter incremented at each `\@msd` read in auxiliary file.

```
8066 \numdef{\@msd@c}{0}
8067 \numdef{\@msd@cR}{0}
8068 %
```

`\add@msd@` `\add@msd@` is a counter incremented at each `\add@msd@data`, that is at each time we prepare the insertion of manuscripts data footnote.

```
8069 \numdef{\add@msd@c}{0}%
8070 \numdef{\add@msd@cR}{0}%
8071 %
```

`\@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.

```
8072 \def\@msdata@list{}%
8073 %
```

XXX.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.

```
8074 \newcommand{\@msd}[1]{%
8075   \unless\ifledRcol%
8076     \global\newdef{\@msd@c}{\@msd@c+\@ne}%
8077     \csgdef{\@msdata@\@msd@c @data}{#1}%
8078     \csxdef{\@msdata@\@msd@c @linenumber}{\the\line@num}%
8079     \csxdef{\@msdata@\@msd@c @abslinenumber}{\the\absline@num}%
8080     \xright@appenditem{\the\absline@num}\to\actionlines@list%
8081     \xright@appenditem{-1010}\to\actions@list%
8082   \else%
8083     \global\newdef{\@msd@cR}{\@msd@cR+\@ne}%
8084     \csgdef{\@msdata@\@msd@cR @dataR}{#1}%
8085     \csxdef{\@msdata@\@msd@cR @linenumberR}{\the\line@numR}%
8086     \csxdef{\@msdata@\@msd@cR @abslinenumberR}{\the\absline@numR}%
8087     \xright@appenditem{\the\absline@numR}\to\actionlines@listR%
8088     \xright@appenditem{-1010}\to\actions@listR%
8089   \fi%
8090 }%
8091 %
```

`\@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.

```
8092 \newcommand{\@stopmsd}[0]{%
8093   \unless\ifledRcol%
8094     \ifcsundef{\@msdata@\@msd@c @stoplinenumber}{%
8095       \csxdef{\@msdata@\@msd@c @stopabslinenumber}{\the\absline@num}%
8096       \csxdef{\@msdata@\@msd@c @stoplinenumber}{\the\line@num}%
8097     }%
8098   \else%
8099     \ifcsundef{\@msdata@\@msd@cR @stoplinenumberR}{%
8100       \csxdef{\@msdata@\@msd@cR @stopabslinenumberR}{\the\absline@numR}%
8101       \csxdef{\@msdata@\@msd@cR @stoplinenumberR}{\the\line@numR}%
8102     }%
8103   }%
8104 \fi%
8105 }%
```

8106 %

XXX.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`.

```

8107 \newcommand{\add@msdata}{%
8108   \bgroup%
8109   \normalfont%
8110   \unless\ifledRcol@%
8111     \numgdef{\add@msd@c}{\add@msd@c+\@ne}%
8112     \ifcsdef{@msdata@\add@msd@c @data}{%
8113       \letcs{\@data}{@msdata@\add@msd@c @data}%
8114       \edef\l@d@nums{%
8115         000|% Start page = we don't print it
8116         \csuse{@msdata@\add@msd@c @linenumber}|% Start line number
8117         000|% Start subline number, for now, not used
8118         000|% End page number, we don't print it
8119         \ifnumless{\csuse{@msdata@\add@msd@c @stopabslinenumber}}{\csuse{
@lastabsline@forpage@\the\page@num}}%
8120         {\csuse{@msdata@\add@msd@c @stoplinenumber}}%End line number if
in the same page
8121         {\csuse{@lastline@forpage@\the\page@num}}%Otherwiser, last
number of the page
8122         |%
8123         000|% End sub line number, for now, not used
8124         \edfont@info%Font
8125       }%
8126       \@msd@options@fullpagefalse%
8127       \if@firstlineofpage%Try if the data are for the full page. If yes
, will add options to the list.
8128         \unless\if@msdata@insertedfrompreviouspage%
8129         \ifnumless{\csuse{@lastabsline@forpage@\the\page@num}}{\csuse{
@msdata@\add@msd@c @stopabslinenumber}+\@ne}%
8130         {%
8131         \numdef{\@tmp}{\add@msd@c+\@ne}%
8132         \ifcsdef{@msdata@\@tmp @abslinenumber}%
8133         {\ifnumequal{\csuse{@msdata@\@tmp @abslinenumber}}{\csuse{
@lastabsline@forpage@\the\page@num}}%
8134         {}%
8135         {\@msd@options@fullpagetrue}%
8136         }%
8137         {\@msd@options@fullpagetrue}%
8138         }%
8139         {}%
8140         \fi%

```



```

8141     \fi%
8142     \listxadd{\@msdata@list}{%
8143       \@msd@options@iffullpage%
8144       \ifluatex%
8145         \csxdef{footnote@luatextextdir}{\the\textdir}%
8146         \csxdef{footnote@luatexpardir}{\the\pardir}%
8147       \fi%
8148       \csdef{@this@crossref@start}{\add@msd@c:start:msdata}%
8149       \csdef{@this@crossref@end}{\add@msd@c:end:msdata}%
8150       \noexpand\csuse{v\@msdata@series footnote}{\@msdata@series}{\@
expandonce\l@d@nums}{\ms@data@label}{\expandonce\data}}%
8151       \reset@msd@options@iffullpage%
8152     }%
8153   }%
8154   }%
8155 \else%
8156   \numgdef{\add@msd@cR}{\add@msd@cR+\@ne}%
8157   \ifcsdef{@msdata@\add@msd@cR @dataR}{%
8158     \letcs{@data}{@msdata@\add@msd@cR @dataR}%
8159     \edef\l@d@nums{%
8160       000|% Start page = we don't print it
8161       \csuse{@msdata@\add@msd@cR @linenumberR}|% Start line number
8162       000|% Start subline number, for now, not used
8163       000|% End page number, we don't print it
8164       \ifnumless{\csuse{@msdata@\add@msd@cR @stopabslinenumberR}}{\
csuse{@lastline@forpageR@\the\page@numR}}%
8165         {\csuse{@msdata@\add@msd@cR @stoplinenumberR}}%End line number
if in the same page
8166         {\csuse{@lastline@forpageR@\the\page@numR}}%Otherwiser, last
number of the page
8167       |%
8168       000|% End sub line number, for now, not used
8169       \edfont@info%Font
8170     }%
8171     \@msd@options@fullpagefalse%
8172     \if@firstlineofpageR%
8173       \unless@if@msdata@insertedfrompreviouspage%
8174         \ifnumless{\csuse{@lastabsline@forpageR@\the\page@numR}}{\
csuse{@msdata@\add@msd@c @stopabslinenumberR}+\@ne}%
8175         {%
8176           \numdef{\@tmp}{\add@msd@cR+\@ne}%
8177           \ifcsdef{@msdata@\@tmp @abslinenumberR}%
8178             {\ifnumequal{\csuse{@msdata@\@tmp @abslinenumberR}}{\csuse{
@lastabsline@forpageR@\the\page@numR}}%
8179               {}%
8180             {\@msd@options@fullpagetrue}%
8181           }%
8182           {\@msd@options@fullpagetrue}%
8183         }%
8184       {}%

```

```

8185     \fi%
8186     \fi%
8187     \listxadd{\@msdata@list}{%
8188       \@msd@options@iffullpage%
8189       \ifluatex%
8190         \csxdef{footnote@luatextextdir}{\the\textdir}%
8191         \csxdef{footnote@luatexpardir}{\the\pardir}%
8192     \fi%
8193     \csdef{@this@crossref@start}{\add@msd@cR:start:msdata}%
8194     \csdef{@this@crossref@end}{\add@msd@cR:end:msdata}%
8195     \noexpand\csuse{v\@msdata@series footnote}{\@msdata@series}{\{
expandonce\l@d@nums}{\ms@data@label}{\expandonce\data}}%
8196     \reset@msd@options@iffullpage%
8197   }%
8198 }%
8199 {}%
8200 \fi%
8201 \egroup%
8202 }%
8203 %

```

`\if@msdata@insertedfrompreviouspage` The `\if@msdata@insertedfrompreviouspage` boolean is set to TRUE if `reledmac` automatically inserts data from previous page in the first line of a page.

```

8204 \newif\if@msdata@insertedfrompreviouspage%
8205 %

```

`\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.

```

8206 \newcommand{\add@msdata@firstlineofpage}{%
8207   \bgroup%
8208   \normalfont%
8209   \unless\ifledRcol{%
8210     \ifcsdef{@msdata@\add@msd@c @data}{%
8211       \ifnumless{\the\absline@num-\@ne}{\csuse{@msdata@\add@msd@c
@stopabslinenumber}}%
8212       {%
8213         \global\@msdata@insertedfrompreviouspagetrue%
8214         \letcs{\@data}{@msdata@\add@msd@c @data}%
8215         \edef\l@d@nums{%
8216           000| Start page = we don't print it
8217           \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
8218           000| Start subline number, for now, not used
8219           000| End page number, we don't print it
8220           \ifnumless{\csuse{@msdata@\add@msd@c @stopabslinenumber}}{\csuse{
@lastabsline@forpage@\the\page@num}}%

```

```

8221         {\csuse{@msdata@\add@msd@c @stoplinenumber}}}%End line number if
in the same page
8222         {\csuse{@lastline@forpage@\the\page@num}}}%Otherwise, last
number of the page
8223         |%
8224         000|% End sub line number, for now, not used
8225         \edfont@info%Font
8226         }%
8227         \msd@options@fullpagefalse%
8228         \ifnumless{\csuse{@lastabsline@forpage@\the\page@num}}{\csuse{
@msdata@\add@msd@c @stopabslinenumber}+\@one}}%We will test if the ms data is
for the full page
8229         {%
8230         \numdef{\@tmp}{\add@msd@c+\@one}%
8231         \ifcsdef{@msdata@\@tmp @abslinenumber}%
8232         {\ifnumequal{\csuse{@msdata@\@tmp @abslinenumber}}{\csuse{
@lastabsline@forpage@\the\page@num}}}%
8233         }%
8234         {\msd@options@fullpagetrue}%
8235         }%
8236         {\msd@options@fullpagetrue}%
8237         }%
8238         {}%
8239         \listxadd{\@msdata@list}{%
8240         \msd@options@iffullpage%
8241         \ifluatex%
8242         \csxdef{footnote@luatextextdir}{\the\textdir}%
8243         \csxdef{footnote@luatexpardir}{\the\pardir}%
8244         \fi%
8245         \csdef{@this@crossref@start}{\add@msd@c:start:msdata}%
8246         \csdef{@this@crossref@end}{\add@msd@c:end:msdata}%
8247         \noexpand\csuse{v\@msdata@series footnote}{\@msdata@series}{\
expandonce\l@d@nums}{\ms@data@label}{\expandonce\data}}%
8248         \reset@msd@options@iffullpage%
8249         }%
8250         }%
8251         {\global\@msdata@insertedfrompreviouspagefalse}%
8252         }{}%
8253         \else%
8254         \ifcsdef{@msdata@\add@msd@cR @dataR}{%
8255         \ifnumless{\the\absline@numR-\@one}{\csuse{@msdata@\add@msd@cR
@stopabslinenumberR}}}%
8256         {%
8257         \global\@msdata@insertedfrompreviouspagetrue%
8258         \letcs{\@data}{@msdata@\add@msd@cR @dataR}%
8259         \edef\l@d@nums{%
8260         000|% Start page = we don't print it
8261         \numexpr\the\line@numR+\@one\relax|}% Start line number = first
line of the page. As \add@msdata@firstlineofpage is called before line
number has been incremented, we increment it for printing

```

```

8262     000|% Start subline number, for now, not used
8263     000|% End page number, we don't print it
8264     \ifnumless{\csuse{@msdata@add@msd@cR @stopabslinenumberR}}{\
csuse{@lastline@forpageR@the\page@numR}}%
8265         {\csuse{@msdata@add@msd@cR @stoplinenumberR}}%End line number
if in the same page
8266         {\csuse{@lastline@forpageR@the\page@numR}}%Otherwise, last
number of the page
8267         |%
8268     000|% End sub line number, for now, not used
8269     \edfont@info%Font
8270     }%
8271     \@msd@options@fullpagefalse%
8272     \ifnumless{\csuse{@lastabsline@forpageR@the\page@numR}}{\csuse{
@msdata@add@msd@cR @stopabslinenumberR}+\@one}%
8273     {%
8274     \numdef{\@tmp}{\add@msd@cR+\@one}%
8275     \ifcsdef{@msdata@\@tmp @abslinenumberR}%
8276     {\ifnumequal{\csuse{@msdata@\@tmp @abslinenumberR}}{\csuse{
@lastabsline@forpageR@the\page@numR}}%
8277     }%
8278     {\@msd@options@fullpagetrue}%
8279     }%
8280     {\@msd@options@fullpagetrue}%
8281     }%
8282     {}%
8283     \listxadd{\@msdata@list}{%
8284     \@msd@options@iffullpage%
8285     \ifluatex%
8286     \csxdef{footnote@luatextextdir}{\the\textdir}%
8287     \csxdef{footnote@luatexpardir}{\the\pardir}%
8288     \fi%
8289     \csdef{@this@crossref@start}{\add@msd@cR:start:msdata}%
8290     \csdef{@this@crossref@end}{\add@msd@cR:end:msdata}%
8291     \noexpand\csuse{v@msdata@series footnote}{\@msdata@series}{\
expandonce\l@d@nums}{\ms@data@label}{\expandonce\data}}%
8292     \reset@msd@options@iffullpage%
8293     }%
8294     }%
8295     {\global\@msdata@insertedfrompreviouspagefalse}%
8296     }{}%
8297     \fi%
8298     \egroup%
8299     }%
8300     %

```

XXX.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 01 \newcommand{\insert@msdata}{%
8302 \def\do##1{##1}%
8303 \dolistloop{\@msdata@list}%
8304 \global\let\@msdata@list\relax%
8305 }%
8306 %
```

XXX.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.

```
8307 \newif\if@msd@options@fullpage%
8308 \newcommand{\@msd@options@iffullpage}[0]{%
8309 \if@msd@options@fullpage%
8310 \noexpand\toggletrue{nonum@}%
8311 \ifdefvoid{\ms@data@label}%
8312 {\noexpand\toggletrue{nosep@}}%
8313 }%
8314 \fi%
8315 }%
8316 \newcommand{\reset@msd@options@iffullpage}[0]{%
8317 \noexpand\togglefalse{nonum@}%
8318 \noexpand\togglefalse{nosep@}%
8319 }%
8320 %
```

XXXI Arrays and tables

XXXI.1 Preamble: macro as environment

The following is borrowed, and renamed, from the `amsmath` package. See also the CTT thread 'eq and amstex', 1995/08/31, started by Keith Reckdahl and ended definitively by David M. Jones.

Several of the `[math]` macros scan their body twice. This means we must collect all text in the body of an environment form before calling the macro.

`\@emptytoks` This is actually defined in the `amsgen` package.

```
8321 \newtoks\@emptytoks
8322
8323 %
```

The rest is from amsmath.

`\l@denbody` A token register to contain the body.

```
8324 \newtoks\l@denbody
8325
8326 %
```

`\addtol@denbody` `\addtol@denbody{arg}` adds `arg` to the token register `\l@denbody`.

```
8327 \newcommand{\addtol@denbody}[1]{%
8328   \global\l@denbody\expandafter{\the\l@denbody#1}}
8329
8330 %
```

`\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`.

```
8331 \newcommand{\l@dcollect@body}[1]{%
8332   \l@denbody{\expandafter#1\expandafter{\the\l@denbody}}%
8333   \edef\processl@denbody{\the\l@denbody\noexpand\end{\@currentenv}}%
8334   \l@denbody\emptytoks \def\l@dbegin@stack{b}%
8335   \begingroup
8336     \expandafter\let\csname\@currentenv\endcsname\l@dcollect@@body
8337     \edef\processl@denbody{\expandafter\noexpand\csname\@currentenv\
8338   endcsname}%
8338     \processl@denbody%
8339   }%
8340
8341 %
```

`\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.

```
8342 \def\l@dpush@begins#1\begin#2{%
8343   \ifx\end#2\else b\expandafter\l@dpush@begins\fi}
8344
8345 %
```

`\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.

```

8346 \def\l@dcollect@body#1\end#2{%
8347   \edef\l@dbegin@stack{\l@dpush@begins#1\begin\end
8348     \expandafter\@gobble\l@dbegin@stack}%
8349   \ifx\@empty\l@dbegin@stack
8350     \endgroup
8351     \@checkend{#2}%
8352     \addtol@denvbody{#1}%
8353   \else
8354     \addtol@denvbody{#1\end{#2}}%
8355   \fi
8356   \processl@denvbody % A little tricky! Note the grouping
8357 }
8358
8359 %

```

There was a question on CTT about how to use `\collect@body` for a macro taking an argument. The following is part of that thread.

```

From: Heiko Oberdiek <oberdiek@uni-freiburg.de>
Newsgroups: comp.text.tex
Subject: Re: Using \collect@body with commands that take >1 argument
Date: Fri, 08 Aug 2003 09:03:20 +0200

```

```

eed132@psu.edu (Evan) wrote:
> I'm trying to make a new Latex environment that acts like the>
> \colorbox command that is part of the color package. I looked through
> the FAQ and ran across this bit about using the \collect@body command
> that is part of AMSLaTeX:
> http://www.tex.ac.uk/cgi-bin/texfaq2html?label=cmdasenv
>
> It almost works. If I do something like the following:
> \newcommand{\redbox}[1]{\colorbox{red}{#1}}
>
> \makeatletter
> \newenvironment{redbox}{\collect@body \redbox}{\}

```

You will get an error message: Command `\redbox` already defined. Thus you must rename either the command `\redbox` or the environment name.

```

> \begin{coloredbox}{blue}
>   Yadda yadda yadda... this is on a blue background...
> \end{coloredbox}
> and can't figure out how to make the \collect@body take this.

> \collect@body \colorbox{red}
> \collect@body {\colorbox{red}}

```

The argument of `\collect@body` has to be one token exactly.

```

\documentclass{article}
\usepackage{color}
\usepackage{amsmath}

\newcommand{\redbox}[1]{\colorbox{red}{#1}}
\makeatletter
\newenvironment{coloredbox}[1]{%
  \def\next@{\colorbox{#1}}%
  \collect@body\next@
}{%

% ignore spaces at begin and end of environment
\newenvironment{coloredboxII}[1]{%
  \def\next@{\mycoloredbox{#1}}%
  \collect@body\next@
}{%
\newcommand{\mycoloredbox}[2]{%
  \colorbox{#1}{\ignorespaces#2\unskip}%
}

% support of optional color model argument
\newcommand\coloredboxIII\endcsname{}
\def\coloredboxIII#1#2{%
  \@coloredboxIII{#1}%
}
\def\@coloredboxIII#1#2{%
  \def\next@{\mycoloredboxIII{#1}{#2}}%
  \collect@body\next@
}
\newcommand{\mycoloredboxIII}[3]{%
  \colorbox#1{#2}{\ignorespaces#3\unskip}%
}

\makeatother

\begin{document}
Black text before
\begin{coloredbox}{blue}
Hello World
\end{coloredbox}
Black text after

Black text before
\begin{coloredboxII}{blue}
Hello World
\end{coloredboxII}
Black text after

Black text before
\begin{coloredboxIII}[rgb]{0,0,1}

```



```

Hello World
\end{coloredboxIII}
Black text after

\end{document}

Yours sincerely
Heiko <oberdiek@uni-freiburg.de>

```

XXXI.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`.

XXXI.2.1 Disabling and restoring commands

`\l@dtabnoexpands` More no expansion for critical and familiar footnotes in tabular environment.

```

8360 \newcommand*{\l@dtabnoexpands}{%
8361   \let\rtab=0%
8362   \let\ctab=0%
8363   \let\ltab=0%
8364   \let\rtabtext=0%
8365   \let\ltabtext=0%
8366   \let\ctabtext=0%
8367   \let\edbeforetab=0%
8368   \let\edaftertab=0%
8369   \let\edatleft=0%
8370   \let\edatright=0%
8371   \let\edvertline=0%
8372   \let\edvertdots=0%
8373   \let\edrowfill=0%
8374 }
8375
8376 %

```

`\disable@familiarnotes` Macros to disable and restore familiar notes, to prevent them from printing multiple times in `edtabularx` and `edarray` environments.

```

8377 \newcommand{\disable@familiarnotes}{%
8378   \unless\ifnofamiliar%
8379     \def\do##1{%
8380       \cslletcs{footnote@##1}{footnote##1}%

```

```

8381     \expandafter\renewcommand \csname footnote##1\endcsname[1]{%
8382     \protected@csxdef{@thefnmark##1}{\csuse{thefootnote##1}}%
8383     \csuse{@footnotemark##1}%
8384     }%
8385     }%
8386     \dolistloop{\@series}%
8387     \fi%
8388 }%
8389 \newcommand{\restore@familiarnotes}{%
8390 \unless\ifnofamiliar%
8391   \def\do##1{%
8392     \csletcs{footnote##1}{footnote@##1}%
8393   }%
8394   \dolistloop{\@series}%
8395   \fi%
8396 }%
8397 %
8398 %

```

`\disable@sidenotes` The same, for side notes.

`\restore@sidenotes`

```

8399 \newcommand{\disable@sidenotes}{%
8400 \let\@@ledrightnote\ledrightnote%
8401 \let\@@ledleftnote\ledleftnote%
8402 \let\@@ledsidenote\ledsidenote%
8403 \let\ledrightnote@gobble%
8404 \let\ledleftnote@gobble%
8405 \let\ledsidenote@gobble%
8406 }%
8407 \newcommand{\restore@sidenotes}{%
8408 \let\ledrightnote\@@ledrightnote%
8409 \let\ledleftnote\@@ledleftnote%
8410 \let\ledsidenote\@@ledsidenote%
8411 }%
8412 %

```

`\disable@endnotes` The same for endnotes.

`\restore@endnotes`

```

8413 \newcommand{\disable@endnotes}{%
8414 \unless\ifnoend%
8415   \def\do##1{%
8416     \csletcs{##1@endnote}{##1endnote}%
8417     \expandafter\renewcommand \csname ##1endnote\endcsname[1]{}%
8418   }%
8419   \dolistloop{\@series}%
8420   \fi%
8421 }%
8422 \newcommand{\restore@endnotes}{%
8423 \unless\ifnofamiliar%
8424   \def\do##1{%

```

```

8425     \csletcs{##1endnote}{##1@endnote}%
8426     }%
8427     \dolistloop{\@series}%
8428     \fi%
8429 }%
8430 %

```

\disable@notes Disable/restore side, familiar and end notes.

\restore@notes

```

8431 \newcommand{\disable@notes}{%
8432   \disable@sidenotes%
8433   \disable@familiarnotes%
8434   \disable@endnotes%
8435 }%
8436 \newcommand{\restore@notes}{%
8437   \restore@sidenotes%
8438   \restore@familiarnotes%
8439   \restore@endnotes%
8440 }%
8441 %

```

\EDTEXT We need to be able to modify the `\edtext` macros and also restore their original definitions.

\xedtext

```

8442 \let\EDTEXT=\edtext
8443 \newcommand{\xedtext}[2]{\EDTEXT{#1}{#2}}
8444 %

```

\EDLABEL We need to be able to modify and restore the `\edlabel` macro.

\xedlabel

```

8445 \let\EDLABEL=\edlabel
8446 \newcommand*{\xedlabel}[1]{\EDLABEL{#1}}
8447 %

```

\xedindex Macros supporting modification and restoration of `\edindex`.

\nulledindex

```

8448 \AtBeginDocument{\let\xedindex\edindex}%
8449 \newcommand{\nulledindex}[2][\jobname]{\@bsphack\@esphack}
8450
8451 %

```

\@line@@num Macro supporting restoration of `\linenum`.

```

8452 \let\@line@@num=\linenum
8453 %

```

\l@dgobblearg `\l@dgobbleoptarg[⟨arg⟩]{⟨arg⟩}` replaces these two arguments (first is optional) by `\relax`.

```
8454 \newcommand*\l@dgobbleoptarg}[2] []{\relax}%
8455
8456 %
```

```
\Relax57 \let\Relax=\relax
\NEXT58 \let\NEXT=\next
8459
8460 %
```

\l@dmodforedtext Modify and restore various macros for when `\edtext` is used.
\l@drestoreforedtext

```
8461 \newcommand{\l@dmodforedtext}{%
8462   \let\edtext\relax
8463   \def\do##1{\global\csletcs{##1footnote}{\l@dgobbleoptarg}}%
8464   \dolistloop{@series}%
8465   \let\edindex\nulledindex
8466   \let\linenum@gobble}
8467 \newcommand{\l@drestoreforedtext}{%
8468   \def\do##1{\global\csletcs{##1footnote}{##1@footnote}}
8469   \dolistloop{@series}%
8470   \let\edindex\xedindex}
8471 %
```

\l@dnullfills Nullify and restore some column fillers, etc.
\l@drestorefills

```
8472 \newcommand{\l@dnullfills}{%
8473   \def\edlabel##1{%
8474     \def\edrowfill##1##2##3{%
8475     }
8476   \newcommand{\l@drestorefills}{%
8477     \def\edrowfill##1##2##3{\@EDROWFILL@{##1}{##2}{##3}}%
8478   }
8479
8480 %
```

\letsforverteilen Gathers some lets and other code that is common to the `*verteilen*` macros.

```
8481 \newcommand{\letsforverteilen}{%
8482   \let\edtext\xedtext
8483   \let\edindex\xedindex
8484   \def\do##1{\global\csletcs{##1footnote}{##1@footnote}}
8485   \dolistloop{@series}%
8486   \let\linenum@line@num
8487   \hilfsskip=\l@dcolwidth%
8488   \advance\hilfsskip by -\wd\hilfsbox
8489   \def\edlabel##1{\xedlabel{##1}}
8490
8491 %
```

`\disablel@dtabfeet` Declarations for using or using `\edtext` inside tabulars. The default at this point is for
`\enablel@dtabfeet` `\edtext`.

```
8492 \newcommand\disablel@dtabfeet{\l@modforedtext}%
8493 \newcommand\enablel@dtabfeet{\l@drestoreforedtext}%
8494 %
```

XXXI.2.2 Counters, boxes and lengths

`\l@dampcount` `\l@dampcount` is a counter for the & column dividers and `\l@dcolcount` is a counter
`\l@dcolcount` for the columns.

```
8495 \newcount\l@dampcount
8496 \l@dampcount=1\relax
8497 \newcount\l@dcolcount
8498 \l@dcolcount=0\relax
8499
8500 %
```

`\hilfsbox` Some (temporary) helper items.

```
\hilfsskip
\Hilfsbox
\hilfscount
8501 \newbox\hilfsbox
8502 \newskip\hilfsskip
8503 \newbox\Hilfsbox
8504 \newcount\hilfscount
8505
8506 %
```

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).

```
8507 \newdimen\dcoli
8508 \newdimen\dcolii
8509 \newdimen\dcoliii
8510 \newdimen\dcoliv
8511 \newdimen\dcolv
8512 \newdimen\dcolvi
8513 \newdimen\dcolvii
8514 \newdimen\dcolviii
8515 \newdimen\dcolix
8516 \newdimen\dcolx
8517 \newdimen\dcolxi
8518 \newdimen\dcolxii
8519 \newdimen\dcolxiii
8520 \newdimen\dcolxiv
8521 \newdimen\dcolxv
8522 \newdimen\dcolxvi
8523 \newdimen\dcolxvii
8524 \newdimen\dcolxviii
8525 \newdimen\dcolxix
```

```

8526 \newdimen\dcollx
8527 \newdimen\dcollxi
8528 \newdimen\dcollxii
8529 \newdimen\dcollxiii
8530 \newdimen\dcollxiv
8531 \newdimen\dcollxv
8532 \newdimen\dcollxvi
8533 \newdimen\dcollxvii
8534 \newdimen\dcollxviii
8535 \newdimen\dcollxix
8536 \newdimen\dcollxxx
8537 \newdimen\dcollerr % added for error handling
8538
8539 %

```

`\l@dcollwidth` This is a cunning way of storing the columnwidths indexed by the column number `\l@dcollcount`, like an array. (was `\Dimenzuordnung`)

```

8540 \newcommand{\l@dcollwidth}{\ifcase \the\l@dcollcount \dcoli %???
8541 \or \dcoli \or \dcolii \or \dcoliii
8542 \or \dcoliv \or \dcolv \or \dcolvi
8543 \or \dcolvii \or \dcolviii \or \dcolix \or \dcolx
8544 \or \dcolxi \or \dcolxii \or \dcolxiii
8545 \or \dcolxiv \or \dcolxv \or \dcolxvi
8546 \or \dcolxvii \or \dcolxviii \or \dcolxix \or \dcolxx
8547 \or \dcolxxi \or \dcolxxii \or \dcolxxiii
8548 \or \dcolxxiv \or \dcolxxv \or \dcolxxvi
8549 \or \dcolxxvii \or \dcolxxviii \or \dcolxxix \or \dcolxxx
8550 \else \dcollerr \fi}
8551
8552 %

```

`\stepl@dcollcount` This increments the column counter, and issues an error message if it is too large.

```

8553 \newcommand*{\stepl@dcollcount}{\advance\l@dcollcount\@ne
8554 \ifnum\l@dcollcount>30\relax
8555 \led@err@TooManyColumns
8556 \fi}
8557
8558 %

```

`\l@dsetmaxcolwidth` Sets the column width to the maximum value seen so far.

```

8559 \newcommand{\l@dsetmaxcolwidth}{%
8560 \ifdim\l@dcollwidth < \wd\hilfsbox
8561 \l@dcollwidth = \wd\hilfsbox
8562 \else \relax \fi}
8563
8564 %

```

`\measurecell` Measure (recursively) the width required for a math cell.

```

8565 \def\measurecell #1&{%
8566   \ifx #1\ \ifnum\l@dcolcount=0\let\NEXT\relax%
8567     \else\l@dcheckcols%
8568       \l@dcolcount=0%
8569       \let\NEXT\measurecell%
8570     \fi%
8571   \else\setbox\hilfsbox=\hbox{\$displaystyle{#1}$}%
8572     \step1@dcolcount%
8573     \l@dsetmaxcolwidth%
8574     \let\NEXT\measurecell%
8575   \fi\NEXT}
8576
8577 %

```

`\measuretcell` Measure (recursively) the width required for a text cell.

```

8578 \def\measuretcell #1&{%
8579   \ifx #1\ \ifnum\l@dcolcount=0\let\NEXT\relax%
8580     \else\l@dcheckcols%
8581       \l@dcolcount=0%
8582       \let\NEXT\measuretcell%
8583     \fi%
8584   \else\setbox\hilfsbox=\hbox{#1}%
8585     \step1@dcolcount%
8586     \l@dsetmaxcolwidth%
8587     \let\NEXT\measuretcell%
8588   \fi\NEXT}
8589
8590 %

```

`\measuremrow` Measure (recursively) the width required for a math row.

```

8591 \def\measuremrow #1\{&%
8592   \ifx #1&\let\NEXT\relax%
8593   \else\measurecell #1&\&\&%
8594     \let\NEXT\measuremrow%
8595   \fi\NEXT}
8596 %

```

`\measuretrow` Measure (recursively) the width required for a text row.

```

8597 \def\measuretrow #1\{&%
8598   \ifx #1&\let\NEXT\relax%
8599   \else\measuretcell #1&\&\&%
8600     \let\NEXT\measuretrow%
8601   \fi\NEXT}
8602
8603 %

```

`\edtabcolsep` The length `\edtabcolsep` controls the distance between columns.

```
8604 \newskip\edtabcolsep
8605 \global\edtabcolsep=10pt
8606
8607 %
```

`\variab` `\newcommand{\variab}{\relax}`

```
8609
8610 %
```

`\l@dcheckcols` Check that the number of columns is consistent.

```
8611 \newcommand*\l@dcheckcols{%
8612   \ifnum\l@dcolcount=1\relax
8613   \else
8614     \ifnum\l@dampcount=1\relax
8615     \else
8616       \ifnum\l@dcolcount=\l@dampcount\relax
8617       \else
8618         \l@d@err@UnequalColumns
8619       \fi
8620     \fi
8621     \l@dampcount=\l@dcolcount
8622   \fi}
8623
8624 %
```

`\edfilldimen` A length.

```
8625 \newdimen\edfilldimen
8626 \edfilldimen=0pt
8627
8628 %
```

`\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`

```
8629 \newcounter{addcolcount}
8630 \renewcommand{\theadcolcount}{\romannumeral \c@addcolcount}
8631 %
```

XXXI.2.3 Tabular typesetting

`\setmcellright` Typeset (recursively) cells of display math right justified.


```

8632 \def\setmcellright #1{\def\edlabel##1{}}%
8633 \let\edindex\nulledindex
8634 \ifx #1\ \ifnum\l@dc@lcount=0%\removelastskip
8635 \let\Next\relax%
8636 \else\l@dc@lcount=0%
8637 \let\Next=\setmcellright%
8638 \fi%
8639 \else%
8640 \disablel@dtabfeet%
8641 \step1@dc@lcount%
8642 \disable@notes%
8643 \setbox\hilfsbox=\hbox{$\displaystyle{#1}$}%
8644 \restore@notes%
8645 \letsforverteilen%
8646 \hskip\hilfsskip$\displaystyle{#1}$%
8647 \hskip\edtabcolsep%
8648 \let\Next=\setmcellright%
8649 \fi\Next}
8650
8651 %

```

`\settcellright` Typeset (recursively) cells of text right justified.

```

8652 \def\settcellright #1{\def\edlabel##1{}}%
8653 \let\edindex\nulledindex
8654 \ifx #1\ \ifnum\l@dc@lcount=0%\removelastskip
8655 \let\Next\relax%
8656 \else\l@dc@lcount=0%
8657 \let\Next=\settcellright%
8658 \fi%
8659 \else%
8660 \disablel@dtabfeet%
8661 \step1@dc@lcount%
8662 \disable@notes%
8663 \setbox\hilfsbox=\hbox{#1}%
8664 \restore@notes%
8665 \letsforverteilen%
8666 \hskip\hilfsskip#1%
8667 \hskip\edtabcolsep%
8668 \let\Next=\settcellright%
8669 \fi\Next}
8670 %

```

`\setmcelleft` Typeset (recursively) cells of display math left justified.

```

8671 \def\setmcelleft #1{\def\edlabel##1{}}%
8672 \let\edindex\nulledindex
8673 \ifx #1\ \ifnum\l@dc@lcount=0 \let\Next\relax%
8674 \else\l@dc@lcount=0%
8675 \let\Next=\setmcelleft%

```

```

8676     \fi%
8677 \else  \disablel@dtabfeet%
8678         \stepl@dcolcount%
8679         \disable@notes%
8680         \setbox\hilfsbox=\hbox{\displaystyle{#1}}%
8681         \restore@notes%
8682         \letsforverteilen%
8683         $\displaystyle{#1}$\hskip\hilfsskip\hskip\edtabcolsep%
8684         \let\Next=\setmcellleft%
8685 \fi\Next}
8686
8687 %

```

`\setmcellleft` Typeset (recursively) cells of text left justified.

```

8688 \def\setmcellleft #1&{\def\edlabel##1{}}%
8689     \let\edindex\nulledindex
8690     \ifx #1\ \ifnum\l@dcolcount=0 \let\Next\relax%
8691         \else\l@dcolcount=0%
8692             \let\Next=\setmcellleft%
8693         \fi%
8694 \else  \disablel@dtabfeet%
8695         \stepl@dcolcount%
8696         \disable@notes%
8697         \setbox\hilfsbox=\hbox{#1}%
8698         \restore@notes%
8699         \letsforverteilen%
8700         #1\hskip\hilfsskip\hskip\edtabcolsep%
8701         \let\Next=\setmcellleft%
8702 \fi\Next}
8703 %

```

`\setmcellcenter` Typeset (recursively) cells of display math centered.

```

8704 \def\setmcellcenter #1&{\def\edlabel##1{}}%
8705     \let\edindex\nulledindex
8706     \ifx #1\ \ifnum\l@dcolcount=0\let\Next\relax%
8707         \else\l@dcolcount=0%
8708             \let\Next=\setmcellcenter%
8709         \fi%
8710 \else  \disablel@dtabfeet%
8711         \stepl@dcolcount%
8712         \disable@notes%
8713         \setbox\hilfsbox=\hbox{\displaystyle{#1}}%
8714         \restore@notes%
8715         \letsforverteilen%
8716         \hskip 0.5\hilfsskip$\displaystyle{#1}$\hskip0.5\hilfsskip%
8717         \hskip\edtabcolsep%
8718         \let\Next=\setmcellcenter%
8719 \fi\Next}

```

```
8720
8721 %
```

\settcclcenter Typeset (recursively) cells of text centered.

```
8722 \def\settcclcenter #1{\def\edlabel##1{}}%
8723   \let\edindex\nulledindex
8724   \ifx #1\ \ifnum\l@dcolcount=0 \let\Next\relax%
8725     \else\l@dcolcount=0%
8726       \let\Next=\settcclcenter%
8727     \fi%
8728   \else \disablel@dtabfeet%
8729     \stepl@dcolcount%
8730     \disable@notes%
8731     \setbox\hilfsbox=\hbox{#1}%
8732     \restore@notes%
8733     \letsforverteilen%
8734     \hskip 0.5\hilfsskip #1\hskip 0.5\hilfsskip%
8735     \hskip\edtabcolsep%
8736     \let\Next=\settcclcenter%
8737   \fi\Next}
8738
8739 %
```

\NEXT₄₀ \let\NEXT=\relax

```
8741
8742 %
```

\setmrowright Typeset (recursively) rows of right justified math.

```
8743 \def\setmrowright #1\{\%
8744   \ifx #1& \let\NEXT\relax
8745   \else \centerline{\setmcellright #1&\&\&}
8746     \let\NEXT=\setmrowright
8747   \fi\NEXT}
8748 %
```

\settroright Typeset (recursively) rows of right justified text.

```
8749 \def\settroright #1\{\%
8750   \ifx #1& \let\NEXT\relax
8751   \else \centerline{\settcclright #1&\&\&}
8752     \let\NEXT=\settroright
8753   \fi\NEXT}
8754
8755 %
```

\setmrowleft Typeset (recursively) rows of left justified math.

```

8756 \def\setmrowleft #1\{\%
8757   \ifx #1& \let\NEXT\relax
8758   \else \centerline{\setmcellleft #1&\&\&}
8759     \let\NEXT=\setmrowleft
8760   \fi\NEXT}
8761 %

```

\settrorleft Typeset (recursively) rows of left justified text.

```

8762 \def\settrorleft #1\{\%
8763   \ifx #1& \let\NEXT\relax
8764   \else \centerline{\settrcellleft #1&\&\&}
8765     \let\NEXT=\settrorleft
8766   \fi\NEXT}
8767
8768 %

```

\setmrowcenter Typeset (recursively) rows of centered math.

```

8769 \def\setmrowcenter #1\{\%
8770   \ifx #1& \let\NEXT\relax%
8771   \else \centerline{\setmcellcenter #1&\&\&}
8772     \let\NEXT=\setmrowcenter
8773   \fi\NEXT}
8774 %

```

\settrorcenter Typeset (recursively) rows of centered text.

```

8775 \def\settrorcenter #1\{\%
8776   \ifx #1& \let\NEXT\relax
8777   \else \centerline{\settrcellcenter #1&\&\&}
8778     \let\NEXT=\settrorcenter
8779   \fi\NEXT}
8780
8781 %

```

\nullsetzen⁸² \newcommand{\nullsetzen}{\%

```

8783   \step1@dcolcount%
8784   \l@dcolwidth=0pt%
8785   \ifnum\l@dcolcount=30\let\NEXT\relax%
8786     \l@dcolcount=0\relax
8787   \else\let\NEXT\nullsetzen%
8788   \fi\NEXT}
8789
8790 %

```

\edatleft \edatleft [*math*]{*symbol*}{*len*}. Left *symbol*, 2*len* high with prepended *math* vertically centered.

```

8791 \newcommand{\edatleft}[3][\@empty]{%
8792   \ifx#1\@empty
8793     \vbox to 10pt{\vss\hbox{\left#2\vrule width0pt height #3
8794       depth 0pt \right. $\hss}\vfil}
8795   \else
8796     \vbox to 4pt{\vss\hbox{\left#2\vrule width0pt height #3
8797       depth 0pt \right. $\vfil}
8798   \fi}
8799 %

```

`\edatright` `\edatright` [*math*]{*symbol*}{*len*}. Right *symbol*, 2*len* high with appended *math* vertically centered.

```

8800 \newcommand{\edatright}[3][\@empty]{%
8801   \ifx#1\@empty
8802     \vbox to 10pt{\vss\hbox{\left.\vrule width0pt height #3
8803       depth 0pt \right#2 $\hss}\vfil}
8804   \else
8805     \vbox to 4pt{\vss\hbox{\left.\vrule width0pt height #3
8806       depth 0pt \right#2 #1 $\vfil}
8807   \fi}
8808 %
8809 %

```

`\edvertline` `\edvertline`{*len*} vertical line *len* high.

```

8810 \newcommand{\edvertline}[1]{\vbox to 8pt{\vss\hbox{\vrule height #1}\vfil}}
8811 %
8812 %

```

`\edvertdots` `\edvertdots`{*len*} vertical dotted line *len* high.

```

8813 \newcommand{\edvertdots}[1]{\vbox to 1pt{\vss\vbox to #1%
8814   {\cleaders\hbox{\m@th\hbox{.}\vbox to 0.5em{ }}\vfil}}}
8815 %
8816 %

```

`\l@dtabaddcols` `\l@dtabaddcols`{*startcol*}{*endcol*} adds the widths of the columns *startcol* through *endcol* to `\edfilldimen`. It is a \LaTeX style reimplement of the original `\@add@`.

```

8817 \newcommand{\l@dtabaddcols}[2]{%
8818   \l@dcheckstartend{#1}{#2}%
8819   \ifl@dstartendok
8820     \setcounter{addcolcount}{#1}%
8821     \@whilenum \value{addcolcount}<#2\relax \do
8822       {\advance\edfilldimen by \the \csname dcol\theaddcolcount\endcsname
8823         \advance\edfilldimen by \edtabcolsep
8824         \stepcounter{addcolcount}}%
8825     \advance\edfilldimen by \the \csname dcol\theaddcolcount\endcsname
8826   \fi

```

```
8827 }
8828
8829 %
```

`\ifl@dstartendok` `\l@dcheckstartend{⟨startcol⟩}{⟨endcol⟩}` checks that the values of `⟨startcol⟩` and `⟨endcol⟩` are sensible. If they are then `\ifl@dstartendok` is set TRUE, otherwise it is set FALSE.

```
8830 \newif\ifl@dstartendok
8831 \newcommand{\l@dcheckstartend}[2]{%
8832   \l@dstartendoktrue
8833   \ifnum #1<\@ne
8834     \l@dstartendokfalse
8835     \led@err@LowStartColumn
8836   \fi
8837   \ifnum #2>30\relax
8838     \l@dstartendokfalse
8839     \led@err@HighEndColumn
8840   \fi
8841   \ifnum #1>#2\relax
8842     \l@dstartendokfalse
8843     \led@err@ReverseColumns
8844   \fi
8845 }
8846
8847 %
```

`\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.

```
8848 \newcommand*\edrowfill}[3]{%
8849   \l@dtabaddcols{#1}{#2}%
8850   \hb@xt@ \the\l@dcolwidth{\hb@xt@ \the\edfilldimen{#3}\hss}}
8851 \let\@edrowfill@=\edrowfill
8852 \def\@EDROWFILL@#1#2#3{\@edrowfill@{#1}{#2}{#3}}
8853
8854 %
```

`\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`.

```
8855 \newcommand{\leftltab}[1]{%
8856   \hb@xt@\z@{\vbox{\edtabindent%
```

```

8857 \moveleft\Hilfsskip\hbox{\ #1}\hss}}
8858
8859 %

```

\lefttab `\lefttab{<text>}{<math>}` for `\edbeforetab` in `\rtab`.

```

8860 \newcommand{\lefttab}[2]{%
8861 #2\hb@xt@z@{\vbox{\edtabindent%
8862 \advance\Hilfsskip by\dcoli%
8863 \moveleft\Hilfsskip\hbox{\ #1}\hss}}
8864
8865 %

```

\leftctab `\leftctab{<text>}{<math>}` for `\edbeforetab` in `\ctab`.

```

8866 \newcommand{\leftctab}[2]{%
8867 \hb@xt@z@{\vbox{\edtabindent\l@dcolcount=\l@dampcount%
8868 \advance\Hilfsskip by 0.5\dcoli%
8869 \setbox\hilfsbox=\hbox{\def\edlabel##1{}}%
8870 \disablel@dtabfeet$\displaystyle{#2}$}%
8871 \advance\Hilfsskip by -0.5\wd\hilfsbox%
8872 \moveleft\Hilfsskip\hbox{\ #1}\hss}}%
8873 #2}
8874
8875 %

```

\rightctab `\rightctab{<math>}{<text>}` for `\edaftertab` in `\ctab`.

```

8876 \newcommand{\rightctab}[2]{%
8877 \setbox\hilfsbox=\hbox{\def\edlabel##1{}}%
8878 \disablel@dtabfeet#2}\l@dampcount=\l@dcolcount%
8879 #1\hb@xt@z@{\vbox{\edtabindent\l@dcolcount=\l@dampcount%
8880 \advance\Hilfsskip by 0.5\l@dcolwidth%
8881 \advance\Hilfsskip by -\wd\hilfsbox%
8882 \setbox\hilfsbox=\hbox{\def\edlabel##1{}}%
8883 \disablel@dtabfeet$\displaystyle{#1}$}%
8884 \advance\Hilfsskip by -0.5\wd\hilfsbox%
8885 \advance\Hilfsskip by \edtabcolsep%
8886 \moveright\Hilfsskip\hbox{ #2}\hss}}%
8887 }
8888
8889 %

```

\rightltab `\rightltab{<math>}{<text>}` for `\edaftertab` in `\ltab`.

```

8890 \newcommand{\rightltab}[2]{%
8891 \setbox\hilfsbox=\hbox{\def\edlabel##1{}}%
8892 \disablel@dtabfeet#2}\l@dampcount=\l@dcolcount%
8893 #1\hb@xt@z@{\vbox{\edtabindent\l@dcolcount=\l@dampcount%
8894 \advance\Hilfsskip by\l@dcolwidth%

```

```

8895 \advance\Hilfsskip by-\wd\hilfsbox%
8896 \setbox\hilfsbox=\hbox{\def\edlabel##1{}}%
8897 \disablel@dtabfeet$\displaystyle{#1}$}%
8898 \advance\Hilfsskip by-\wd\hilfsbox%
8899 \advance\Hilfsskip by\edtabcolsep%
8900 \moveright\Hilfsskip\hbox{ #2}}\hss}%
8901 }
8902
8903 %

```

\righttab `\righttab{math}{text}` for `\edaftertab` in `\rtab`.

```

8904 \newcommand{\righttab}[2]{%
8905 \setbox\hilfsbox=\hbox{\def\edlabel##1{}}%
8906 \disablel@dtabfeet#2}%
8907 #1\hb@xt@\z@\vbox{\edtabindent%
8908 \advance\Hilfsskip by-\wd\hilfsbox%
8909 \advance\Hilfsskip by\edtabcolsep%
8910 \moveright\Hilfsskip\hbox{ #2}}\hss}%
8911 }
8912
8913 %

```

\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.

```

8914 \newcommand{\rtab}[1]{%
8915 \l@dnnullfills
8916 \def\edbeforetab##1##2{\lefttab{##1}{##2}}%
8917 \def\edaftertab##1##2{\righttab{##1}{##2}}%
8918 \measurebody{#1}%
8919 \l@drestorefills
8920 \variab
8921 \setmrowright #1\&\%
8922 \enablel@dtabfeet}
8923
8924 %

```

\measurebody `\measurebody{body}` measures the array *body*.

```

8925 \newcommand{\measurebody}[1]{%
8926 \disablel@dtabfeet%
8927 \l@dcolcount=0%
8928 \nullsetzen%
8929 \l@dcolcount=0
8930 \measuremrow #1\&\%
8931 \global\l@dampcount=1}
8932
8933 %

```


`\rtabtext` `\rtabtext{<body>}` typesets *<body>* as a tabular with the entries right justified.

```

8934 \newcommand{\rtabtext}[1]{%
8935 \l@dnnullfills
8936 \measuretbody{#1}%
8937 \l@drestorefills
8938 \variab
8939 \settroright #1\\&\\%
8940 \enablel@dtabfeet}
8941
8942 %

```

`\measuretbody` `\measuretbody{<body>}` measures the tabular *<body>*.

```

8943 \newcommand{\measuretbody}[1]{%
8944 \disable@notes%
8945 \disablel@dtabfeet%
8946 \l@dcolcount=0%
8947 \nullsetzen%
8948 \l@dcolcount=0
8949 \measuretrorow #1\\&\\%
8950 \restore@notes%
8951 \global\l@dampcount=1}
8952
8953 %

```

`\ltab` Array with entries left justified.

```

\edbeforetab
\edaftertab
8954 \newcommand{\ltab}[1]{%
8955 \l@dnnullfills
8956 \def\edbeforetab##1##2{\leftltab{##1}{##2}}%
8957 \def\edaftertab##1##2{\rightltab{##1}{##2}}%
8958 \measuretbody{#1}%
8959 \l@drestorefills
8960 \variab
8961 \settrorowleft #1\\&\\%
8962 \enablel@dtabfeet}
8963
8964 %

```

`\ltabtext` Tabular with entries left justified.

```

8965 \newcommand{\ltabtext}[1]{%
8966 \l@dnnullfills
8967 \measuretbody{#1}%
8968 \l@drestorefills
8969 \variab
8970 \settrorowleft #1\\&\\%
8971 \enablel@dtabfeet}
8972
8973 %

```

`\ctab` Array with centered entries.

```

\edbeforetab
\edaftertab
8974 \newcommand{\ctab}[1]{%
8975   \l@dnnullfills
8976   \def\edbeforetab##1##2{\leftctab{##1}{##2}}%
8977   \def\edaftertab##1##2{\rightctab{##1}{##2}}%
8978   \measurebody{#1}%
8979   \l@drestorefills
8980   \variab
8981   \setmrowcenter #1\&\&%
8982   \enablel@dtabfeet}
8983
8984 %

```

`\ctabtext` Tabular with entries centered.

```

8985 \newcommand{\ctabtext}[1]{%
8986   \l@dnnullfills
8987   \measuretbody{#1}%
8988   \l@drestorefills
8989   \variab
8990   \settrrowcenter #1\&\&%
8991   \enablel@dtabfeet}
8992
8993 %

```

```

\spreadtext94 \newcommand{\spreadtext}[1]{%\l@dcolcount=\l@dampcount%
8995   \hb@xt@ \the\l@dcolwidth{\hbox{#1}\hss}}
8996 %

```

```

\spreadmath97 \newcommand{\spreadmath}[1]{%
8998   \hb@xt@ \the\l@dcolwidth{\hbox{$\displaystyle{#1}$}\hss}}
8999
9000 %

```

`\HILFSskip` More helpers.

```

\Hilfsskip
9001 \newskip\HILFSskip
9002 \newskip\Hilfsskip
9003
9004 %

```

```

\EDTABINDENT05 \newcommand{\EDTABINDENT}{%
9006   \ifnum\l@dcolcount=30\let\NEXT\relax\l@dcolcount=0%
9007   \else\step\l@dcolcount%
9008   \advance\Hilfsskip by\l@dcolwidth%
9009   \ifdim\l@dcolwidth=0pt\advance\hilfscout\@ne

```

```

9010     \else\advance\Hilfsskip by \the\hilfscout\edtabcolsep%
9011     \hilfscout=1\fi%
9012     \let\NEXT=\EDTABINDENT%
9013 \fi\NEXT}%
9014 %

```

`\edtabindent` (was `\tabindent`)

```

9015 \newcommand{\edtabindent}{%
9016   \l@dcolcount=0\relax
9017   \Hilfsskip=0pt%
9018   \hilfscout=1\relax
9019   \EDTABINDENT%
9020   \hilfsskip=\hsize%
9021   \advance\hilfsskip -\Hilfsskip%
9022   \Hilfsskip=0.5\hilfsskip%
9023 }%
9024 %
9025 %

```

`\EDTAB` (was `\TAB`)

```

9026 \def\EDTAB #1|#2|{%
9027   \setbox\tabhilfbox=\hbox{$\displaystyle{#1}$}%
9028   \setbox\tabHilfbox=\hbox{$\displaystyle{#2}$}%
9029   \advance\tabelskip -\wd\tabhilfbox%
9030   \advance\tabelskip -\wd\tabHilfbox%
9031   \unhbox\tabhilfbox\hskip\tabelskip%
9032   \unhbox\tabHilfbox}%
9033 %
9034 %

```

`\EDTABtext` (was `\TABtext`)

```

9035 \def\EDTABtext #1|#2|{%
9036   \setbox\tabhilfbox=\hbox{#1}%
9037   \setbox\tabHilfbox=\hbox{#2}%
9038   \advance\tabelskip -\wd\tabhilfbox%
9039   \advance\tabelskip -\wd\tabHilfbox%
9040   \unhbox\tabhilfbox\hskip\tabelskip%
9041   \unhbox\tabHilfbox}%
9042 %

```

`\tabhilfbox` Further helpers.

```

\-tabHilfbox
9043 \newbox\tabhilfbox
9044 \newbox\tabHilfbox
9045 %
9046 %

```

XXXI.2.4 Environments

`edarrayl edarrayc edarrayr` The 'environment' forms for `\ltab`, `\ctab` and `\rtab`.

```
9047 \newenvironment{edarrayl}{\l@dcollect@body\ltab}{\}
9048 \newenvironment{edarrayc}{\l@dcollect@body\ctab}{\}
9049 \newenvironment{edarrayr}{\l@dcollect@body\rtab}{\}
9050
9051 %
```

`edtabularl edtabularc edtabularr` The 'environment' forms for `\ltabtext`, `\ctabtext` and `\rtabtext`.

```
9052 \newenvironment{edtabularl}{\l@dcollect@body\ltabtext}{\}
9053 \newenvironment{edtabularc}{\l@dcollect@body\ctabtext}{\}
9054 \newenvironment{edtabularr}{\l@dcollect@body\rtabtext}{\}
9055
9056 %
```

XXXII 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
9057 \newcommand{\initnumbering@quote}{
9058   \ifnoquotation@else
9059   \renewcommand{\quotation}{\par\leavevmode%
9060     \parindent=1.5em%
9061     \skipnumbering%
9062     \ifautopar%
9063       \vskip-\parskip%
9064     \else%
9065       \vskip\topsep%
9066     \fi%
9067     \global\leftskip=\leftmargin%
9068     \global\rightskip=\leftmargin%
9069   }
9070   \renewcommand{\endquotation}{\par%
9071     \global\leftskip=0pt%
9072     \global\rightskip=0pt%
9073     \leavevmode%
9074     \skipnumbering%
9075     \ifautopar%
9076       \vskip-\parskip%
9077     \else%
9078       \vskip\topsep%
9079     \fi%
```

```

9080     }
9081     \renewcommand{\quote}{\par\leavevmode%
9082         \parindent=0pt%
9083         \skipnumbering%
9084         \ifautopar%
9085             \vskip-\parskip%
9086         \else%
9087             \vskip\topsep%
9088         \fi%
9089         \global\leftskip=\leftmargin%
9090         \global\rightskip=\leftmargin%
9091     }
9092     \renewcommand{\endquote}{\par%
9093         \global\leftskip=0pt%
9094         \global\rightskip=0pt%
9095         \leavevmode%
9096         \skipnumbering%
9097         \ifautopar%
9098             \vskip-\parskip%
9099         \else%
9100             \vskip\topsep%
9101         \fi%
9102     }
9103     \fi
9104 }
9105 %

```

XXXIII Section's title commands

XXXIII.1 Commands to disable some feature

`\ledsectnotoc` The `\ledsectnotoc` only disables the `\addcontentsline` macro.

```

9106 \newcommand{\ledsectnotoc}{\let\addcontentsline@gobblethree}
9107 %

```

`\ledsectnomark` The `\ledsectnomark` only disables the `\chaptermark`, `\sectionmark` and `\subsectionmark` macros.

```

9108 \newcommand{\ledsectnomark}{%
9109     \let\chaptermark@gobble%
9110     \let\sectionmark@gobble%
9111     \let\subsectionmark@gobble%
9112 }
9113 %

```

XXXIII.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 `eledpar` 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.

XXXIII.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).

```
9114 \catcode`\#=12
9115 \notbool{@noeled@sec}{%
9116 %
```

`\beforeeledchapter` For technical reasons, not yet solved, page-breaking before chapters can't be made automatically by `eledmac`. Users have to use `\beforeeledchapter`.

```
9117 \ifl@dmemoir
9118   \newcommand\beforeeledchapter{%
9119     \clearforchapter%
9120   }
9121 \else
9122   \newcommand\beforeeledchapter{%
9123     \if@openright%
9124       \cleardoublepage%
9125     \else%
9126       \clearpage%
9127     \fi%
9128   }
9129 \fi
9130 %
```

XXXIII.4 Auxiliary commands

`\print@leftmargin@eledsection` and `\print@rightmargin@eledsection` are added by `reledmac` inside the code of sectioning command, in order to affix line numbers. They include tests for RTL languages.

```

9131 \def\print@rightmargin@eledsection{%
9132   \if@eled@sectioning%
9133     \begingroup%
9134     \if@RTL%
9135       \let\llap\rlap%
9136       \let\leftlinenum\rightlinenum%
9137       \let\leftlinenumR\rightlinenumR%
9138       \let\l@drd@ta\l@dld@ta%
9139       \let\l@drsn@te\l@dlsn@te%
9140     \fi%
9141     \hfill\l@drd@ta \csuse{LR}{\l@drsn@te}%
9142   \endgroup%
9143 \fi%
9144 }%
9145
9146 \def\print@leftmargin@eledsection{%
9147   \if@eled@sectioning%
9148     \leavevmode%
9149     \begingroup%
9150     \if@RTL%
9151       \let\rlap\llap%
9152       \let\rightlinenum\leftlinenum%
9153       \let\rightlinenumR\leftlinenumR%
9154       \let\l@dld@ta\l@drd@ta%
9155       \let\l@dlsn@te\l@drsn@te%
9156     \fi%
9157     \l@dld@ta\csuse{LR}{\l@dlsn@te}%
9158   \endgroup%
9159 \fi%
9160 }%
9161
9162 %

```

XXXIII.5 Patching standard commands

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.

```

9163 \AtBeginDocument{%
9164
9165
9166 \pretocmd{\M@sect}
9167   {\let\old@edtext=\edtext%
9168   \let\edtext=\dummy@edtext@showlemma%
9169   }
9170 {}
9171 {}
9172
9173 \apptocmd{\M@sect}
9174   {\let\edtext=\old@edtext}
9175 {}
9176 {}
9177
9178 \patchcmd{\M@sect}
9179   { #9}
9180   { #9%
9181   \print@rightmargin@eledsection%
9182   }
9183 {}
9184 {}
9185
9186 \patchcmd{\M@sect}
9187   {\hskip #3\relax}
9188   {\hskip #3\relax%
9189   \print@leftmargin@eledsection%
9190   }
9191 {}
9192 {}
9193
9194 \patchcmd{\@mem@old@ssect}
9195   {#5}
9196   {#5%
9197   \print@leftmargin@eledsection%
9198   }
9199 {}
9200 {}
9201
9202 \patchcmd{\@mem@old@ssect}
9203   {\hskip #1}
9204   {\hskip #1%
9205   \print@rightmargin@eledsection%
9206   }
9207 {}
9208 {}
9209
9210
9211
9212 \patchcmd{\scr@startchapter}{\if@openright\cleardoublepage\else\clearpage\

```



```

fi}{%
9213   \if@eled@sectioning\else%
9214   \ifl@dprintingpages\else%
9215   \if@openright\cleardoublepage\else\clearpage\fi%No clearpage inside a
\Pages: will keep critical notes from printing on the title page. Here for
scrbook.
9216   \fi%
9217   \fi%
9218   }
9219   {}
9220   {}
9221
9222 \patchcmd{\@makechapterhead}
9223   {#1}
9224   {\print@leftmargin@eledsection%
9225    #1%
9226   \print@rightmargin@eledsection%
9227   }
9228   {}
9229   {}
9230
9231 \patchcmd{\@makechapterhead}% For BIDI
9232   {\if@RTL\raggedleft\else\raggedright\fi}%
9233   {\if@eled@sectioning\else%
9234    \if@RTL\raggedleft\else\raggedright\fi%
9235   \fi%
9236   }%
9237   {}%
9238   {}%
9239
9240 \patchcmd{\@makeschapterhead}
9241   {#1}
9242   {\print@leftmargin@eledsection%
9243    #1%
9244   \print@rightmargin@eledsection%
9245   }
9246   {}
9247   {}
9248
9249 \pretocmd{\@sect}
9250   {\let\old@edtext=\edtext
9251   \let\edtext=\dummy@edtext@showlemma%
9252   }
9253   {}
9254   {}
9255
9256 \apptocmd{\@sect}
9257   {\let\edtext=\old@edtext}
9258   {}
9259   {}

```

```

9260
9261 \pretocmd{\@ssect}
9262   {\let\old@edtext=\edtext%
9263   \let\edtext=\dummy@edtext@showlemma%
9264   }
9265   {}
9266   {}
9267
9268 \apptocmd{\@ssect}
9269   {\let\edtext=\old@edtext}
9270   {}
9271   {}
9272
9273 %

```

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.

```

9274 \@ifpackageloaded{nameref}{
9275
9276   \patchcmd{\NR@sect}
9277     {#8}
9278     {#8%
9279     \print@rightmargin@eledsection%
9280     }
9281     {}
9282     {}
9283
9284   \patchcmd{\NR@sect}
9285     {\hskip #3\relax}
9286     {\hskip #3\relax%
9287     \print@leftmargin@eledsection%
9288     }
9289     {}
9290     {}
9291
9292   \patchcmd{\NR@ssect}
9293     {#5}
9294     {#5%
9295     \print@rightmargin@eledsection%
9296     }
9297     {}
9298     {}
9299
9300   \patchcmd{\NR@sect}
9301     {\hskip #1}
9302     {\hskip #1%
9303     \print@leftmargin@eledsection%
9304     }
9305     {}
9306     {}

```

```

9307 }%
9308 {
9309 \patchcmd{\@sect}
9310   {#8}
9311   {#8%
9312   \print@rightmargin@eledsection%
9313   }
9314   {}
9315   {}
9316
9317 \patchcmd{\@sect}
9318   {\hskip #3\relax}
9319   {\hskip #3\relax%
9320   \print@leftmargin@eledsection%
9321   }
9322   {}
9323   {}
9324
9325 \patchcmd{\@ssect}
9326   {#5}
9327   {#5%
9328   \print@rightmargin@eledsection%
9329   }
9330   {}
9331   {}
9332
9333 \patchcmd{\@ssect}
9334   {\hskip #1}
9335   {\hskip #1%
9336   \print@leftmargin@eledsection%
9337   }
9338   {}
9339   {}
9340 }%
9341 }%
9342 %

```

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 #.

```

9343 {}}%
9344 \protect\catcode`\#=6 %Space NEEDS by \catcode
9345 %

```

`\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.

```

9346 \AtBeginDocument{%
9347 \patchcmd{\chapter}{\clearforchapter}{%
9348   \if@eled@sectioning\else%

```

```

9349 \ifl@dprintingpages\else%
9350 \clearforchapter%
9351 \fi%
9352 \fi%
9353 }%
9354 {}%
9355 {}%
9356
9357 \patchcmd{\chapter}{\if@openright\cleardoublepage\else\clearpage\fi}{%
9358 \if@eled@sectioning\else%
9359 \ifl@dprintingpages%
9360 \endgraf%
9361 \else%
9362 \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
9363 \fi%
9364 \fi%
9365 }%
9366 {}%
9367 {}%
9368 }%
9369 %

```

`\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.

```

9370 \newif\if@eled@sectioning%
9371 %

```

We reopen a new `\notbool{@noeled@sec}` statement, as we will define the `\elesection` commands.

```

9372 \notbool{@noeled@sec}{%
9373 %

```

XXXIII.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.

```

9374 \newwrite\eled@sectioning@out
9375 %

```

`\eledchapter` And now, the user sectioning commands, which write to the file, and also add content as a “normal” line.

`\eledsection`

`\eledsubsection`

`\eledsubsubsection`

`\eledchapter*`

`\eledsection*`

`\eledsubsection*`

`\eledsubsubsection*`

```

9376 \newcommand{\eledchapter}[2] [] {%
9377   \disable@familiarnotes%
9378   #2%
9379   \restore@familiarnotes%
9380   \ifledRcol%
9381     \immediate\write\eled@sectioningR@out{%
9382       \string\eled@chapter{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR}{-}{R}
9383     }%
9384   \else%
9385     \immediate\write\eled@sectioning@out{%
9386       \string\eled@chapter{#1}{\unexpanded{#2}}{\the\l@dnumpstartsL}{-}{-}
9387     }%
9388   \fi%
9389 }
9390
9391 \newcommand{\eledsection}[2] [] {%
9392   \disable@familiarnotes%
9393   #2%
9394   \restore@familiarnotes%
9395   \ifledRcol%
9396     \immediate\write\eled@sectioningR@out{%
9397       \string\eled@section{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR}{-}{R}
9398     }%
9399   \else%
9400     \immediate\write\eled@sectioning@out{%
9401       \string\eled@section{#1}{\unexpanded{#2}}{\the\l@dnumpstartsL}{-}{-}
9402     }%
9403   \fi%
9404 }
9405
9406 \newcommand{\eledsubsection}[2] [] {%
9407   \disable@familiarnotes%
9408   #2%
9409   \restore@familiarnotes%
9410   \ifledRcol%
9411     \immediate\write\eled@sectioningR@out{%
9412       \string\eled@subsection{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR}{-}{R}
9413     }%
9414   \else%
9415     \immediate\write\eled@sectioning@out{%
9416       \string\eled@subsection{#1}{\unexpanded{#2}}{\the\l@dnumpstartsL}{-}{-}
9417     }%
9418   \fi%
9419 }
9420 \newcommand{\eledsubsubsection}[2] [] {%
9421   \disable@familiarnotes%
9422   #2%
9423   \restore@familiarnotes%
9424   \ifledRcol%

```

```

9425     \immediate\write\eled@sectioningR@out{%
9426         \string\eled@subsubsection{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR
9427     }-}{R}
9428     }%
9429     \else%
9430         \immediate\write\eled@sectioning@out{%
9431             \string\eled@subsubsection{#1}{\unexpanded{#2}}{\the\l@dnumpstartsL
9432         }-}{L}
9433     }%
9434     \fi%
9435 }
9436
9437 \WithSuffix\newcommand\eledchapter*[2] [] {%
9438     \disable@familiarnotes%
9439     #2%
9440     \restore@familiarnotes%
9441     \ifledRcol%
9442         \immediate\write\eled@sectioningR@out{%
9443             \string\eled@chapter{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR}-*}{R}
9444         }%
9445     \else%
9446         \immediate\write\eled@sectioning@out{%
9447             \string\eled@chapter{#1}{\unexpanded{#2}}{\the\l@dnumpstartsL}-*}{L}
9448         }%
9449     \fi%
9450 }
9451
9452 \WithSuffix\newcommand\eledsection*[2] [] {%
9453     \disable@familiarnotes%
9454     #2%
9455     \restore@familiarnotes%
9456     \ifledRcol%
9457         \immediate\write\eled@sectioningR@out{%
9458             \string\eled@section{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR}-*}{R}
9459         }%
9460     \else%
9461         \immediate\write\eled@sectioning@out{%
9462             \string\eled@section{#1}{\unexpanded{#2}}{\the\l@dnumpstartsL}-*}{L}
9463         }%
9464     \fi%
9465 }
9466
9467 \WithSuffix\newcommand\eledsubsection*[2] [] {%
9468     \disable@familiarnotes%
9469     #2%
9470     \restore@familiarnotes%
9471     \ifledRcol%
9472         \immediate\write\eled@sectioningR@out{%
9473             \string\eled@subsection{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR}-*}{R}

```

```

R}
9473   }%
9474   \else%
9475     \immediate\write\eled@sectioning@out{%
9476     \string\eled@subsection{#1}{\unexpanded{#2}}{\the\l@dumpstartsL
}
9477   }%
9478   \fi%
9479 }
9480
9481 \WithSuffix\newcommand\eledsubsubsection*[2] [] {%
9482   \disable@familiarnotes%
9483   #2%
9484   \restore@familiarnotes%
9485   \ifledRcol%
9486     \immediate\write\eled@sectioningR@out{%
9487     \string\eled@subsubsection{#1}{\unexpanded{#2}}{\the\l@dumpstartsR
}
9488   }%
9489   \else%
9490     \immediate\write\eled@sectioning@out{%
9491     \string\eled@subsubsection{#1}{\unexpanded{#2}}{\the\l@dumpstartsL
}
9492   }%
9493   \fi%
9494 }
9495 %

```

XXXIII.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 L^AT_EX sectioning command.
2. Mandatory arguments of L^AT_EX sectioning command.
3. Pstart number.
4. Side: R if right, nothing if left.
5. Starred or not.

```

9496 \def\eled@chapter#1#2#3#4#5{%
9497   \ifstrempy{#4}%
9498   {%
9499     \ifstrempy{#1}%
9500     {%
9501     \csgdef\eled@sectioning@#3#5}{\let\edtext=\dummy@edtext@showlemma\
chapter{#2}}%

```

```

9502 \csgdef{eled@sectmark@#3#5}{\let\edtext=\dummy@edtext{}\chaptermark
{#2}}%
9503 }%Need for \pairs, because of using parbox.
9504 {%
9505 \csgdef{eled@sectioning@#3#5}{\let\edtext=\dummy@edtext@showlemma\
chapter[#1]{#2}}%
9506 \csgdef{eled@sectmark@#3#5}{\let\edtext=\dummy@edtext{}\chaptermark
{#2}}%Need for \pairs, because of using parbox.
9507 }%
9508 }%
9509 {%
9510 \ifstrempy{#1}%
9511 {\csgdef{eled@sectioning@#3#5}{\let\edtext=\dummy@edtext@showlemma\
chapter*{#2}}}%
9512 {\csgdef{eled@sectioning@#3#5}{\let\edtext=\dummy@edtext@showlemma\
chapter*{#1}{#2}}}%Bug in LaTeX!
9513 }%
9514 \listcsgadd{eled@sections#5@#}{#3}%
9515 }
9516 \def\eled@section#1#2#3#4#5{%
9517 \ifstrempy{#4}%
9518 {\ifstrempy{#1}%
9519 {%
9520 \csgdef{eled@sectioning@#3#5}{\section{#2}}%
9521 \csgdef{eled@sectmark@#3#5}{\let\edtext=\dummy@edtext{}\sectionmark
{#2}}%Need for \pairs, because of using parbox.
9522 }%
9523 {%
9524 \csgdef{eled@sectioning@#3#5}{\section[#1]{#2}}%
9525 \csgdef{eled@sectmark@#3#5}{\let\edtext=\dummy@edtext{}\sectionmark
{#1}}%Need for \pairs, because of using parbox.
9526 }%
9527 }%
9528 {\ifstrempy{#1}%
9529 {\csgdef{eled@sectioning@#3#5}{\section*{#2}}}%
9530 {\csgdef{eled@sectioning@#3#5}{\section*{#1}{#2}}}%Bug in LaTeX!
9531 }
9532 \listcsgadd{eled@sections#5@#}{#3}%
9533 }
9534 \def\eled@subsection#1#2#3#4#5{%
9535 \ifstrempy{#4}%
9536 {\ifstrempy{#1}%
9537 {%
9538 \csgdef{eled@sectioning@#3#5}{\subsection{#2}}%
9539 \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)
9540 }%
9541 {%
9542 \csgdef{eled@sectioning@#3#5}{\subsection[#1]{#2}}%

```



```

9543 \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)
9544 }%
9545 }%
9546 {\ifstrempy{#1}%
9547 {\csgdef{eled@sectioning@#3#5}{\subsection*{#2}}}%
9548 {\csgdef{eled@sectioning@#3#5}{\subsection*{#1}{#2}}}%Bug in LaTeX!
9549 }
9550 \listcsgadd{eled@sections#5@0}{#3}%
9551 }
9552 \def\eled@subsubsection#1#2#3#4#5{%
9553 \ifstrempy{#4}%
9554 {\ifstrempy{#1}%
9555 {\csgdef{eled@sectioning@#3#5}{\subsubsection{#2}}}%
9556 {\csgdef{eled@sectioning@#3#5}{\subsubsection[#1]{#2}}}%
9557 }%
9558 {\ifstrempy{#1}%
9559 {\csgdef{eled@sectioning@#3#5}{\subsubsection*{#2}}}%
9560 {\csgdef{eled@sectioning@#3#5}{\subsubsection*{#1}{#2}}}%Bug in
LaTeX!
9561 }
9562 \listcsgadd{eled@sections#5@0}{#3}%
9563 }
9564
9565 %

```

End of the conditional test about noeledsec option.

```

9566 }{}
9567 %

```

XXXIV Page breaking or no page breaking depending of specific lines

By default, page breaks are automatic. However, the user can define lines which will force page breaks, or prevent page breaks around one specific line. On the first run, the line-list file records the line number of where the page break is being changed (either forced, or prevented). On the next run, page breaks occur either before or after this line, depending on how the user sets the command. The default setting is after the line.

`\normal@page@break` `\normal@page@break` is an etoolbox list which contains the absolute line number of the last line, for each page.

```

9568 \def\normal@page@break{}
9569 %

```

`\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.

```
9570 \def\l@prev@pb{}
9571 \def\l@prev@nopb{}
9572 %
```

`\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.

```
9573 \newcommand{\ledpb}{\write\linenum@out{\string\led@pb}}
9574 \newcommand{\ledpbnum}[1]{\write\linenum@out{\string\led@pbnum{#1}}}
9575 \newcommand{\lednopb}{\write\linenum@out{\string\led@nopb}}
9576 \newcommand{\lednopbnum}[1]{\write\linenum@out{\string\led@nopbnum{#1}}}
9577 %
```

`\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.

```
9578 \newcommand{\led@pb}{\listxadd{\l@prev@pb}{\the\absline@num}}
9579 \newcommand{\led@pbnum}[1]{\listxadd{\l@prev@pb}{#1}}
9580 \newcommand{\led@nopb}{\listxadd{\l@prev@nopb}{\the\absline@num}}
9581 \newcommand{\led@nopbnum}[1]{\listxadd{\l@prev@nopb}{#1}}
9582 %
```

`\ledpbsetting` The `\ledpbsetting` macro only changes the value of `\led@pb@macro`, for which the default value is before.

```
9583 \def\led@pb@setting{before}
9584 \newcommand{\ledpbsetting}[1]{\gdef\led@pb@setting{#1}}
9585 %
```

`\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`.

```
9586 \newcommand{\led@check@pb}{\xifinlist{\the\absline@num}{\l@prev@pb}{\pagebreak[4]}}
9587 \newcommand{\led@check@nopb}{%
9588   \IfStrEq{\led@pb@setting}{before}{%
9589     \xifinlist{\the\absline@num}{\l@prev@nopb}%
9590     {\numdef{\abs@prevline}{\the\absline@num-1}%
9591     \xifinlist{\abs@prevline}{\normal@page@break}%
9592     {\nopagebreak[4]\enlargethispage{\baselineskip}}%
9593     {}}%
9594   {}}%
```

```

9595     {}%
9596   {}%
9597   \IfStrEq{\led@pb@setting}{after}{%
9598     \xifinlist{\the\absline@num}{\l@prev@nopb}{%
9599       \xifinlist{\the\absline@num}{\normal@page@break}%
9600         {\nopagebreak[4]\enlargethispage{\baselineskip}}%
9601         {}%
9602   }%
9603     {}%
9604   {}%
9605   {}%
9606 }
9607 %

```

XXXV 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.

```

9608 \newcommand{\check@pb@in@verse}{%
9609   \ifinstanza\iflednopbinverse\ifinserthangingsymbol% Using stanzas and
9610   enabling page breaks in verse control, while on a hanging verse.
9611   \ifnum\page@num=\last@page@num\else%If we have change page
9612     \IfStrEq{\led@pb@setting}{before}{%
9613       \numgdef{\abs@line@verse}{\the\absline@num-1}%
9614       \ledpbnum{\abs@line@verse}%
9615     }{}%
9616     \IfStrEq{\led@pb@setting}{after}{%
9617       \numgdef{\abs@line@verse}{\the\absline@num-1}%
9618       \lednopbnum{\abs@line@verse}%
9619     }{}%
9620   \fi%
9621 }
9622 %

```

XXXVI 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

```

9623 \def\Hy@raisedlink@left#1{%
9624   \ifvmode
9625     #1%
9626   \else
9627     \Hy@SaveSpaceFactor
9628     \llap{\smash{%
9629       \begingroup
9630         \let\HyperRaiseLinkLength\@tempdima
9631         \setlength\HyperRaiseLinkLength\HyperRaiseLinkDefault
9632         \HyperRaiseLinkHook
9633       \expandafter\endgroup
9634       \expandafter\raise\the\HyperRaiseLinkLength\hbox{%
9635         \Hy@RestoreSpaceFactor
9636         #1%
9637         \Hy@SaveSpaceFactor
9638       }%
9639     }}%
9640     \Hy@RestoreSpaceFactor
9641     \penalty\@M\hskip\z@\relax
9642   \fi
9643 }
9644 %

```

XXXVII Compatibility with eledmac

Here, we define some commands for the eledmac-compat option.

```

9645 \ifeledmaccompat%
9646
9647   \newcommand{\footnormalX}[1]{\arrangementX[#1]{normal}}%
9648   \newcommand{\footparagraphX}[1]{\arrangementX[#1]{paragraph}}%
9649   \newcommand{\foottwocolX}[1]{\arrangementX[#1]{twocol}}%
9650   \newcommand{\footthreecolX}[1]{\XarrangementX[#1]{threecol}}%
9651
9652   \unless\ifnocritical@
9653     \newcommand{\footnormal}[1]{\Xarrangement[#1]{normal}}%
9654     \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>.

```

9655 \newcommand{\foottwocol}[1]{\Xarrangement[#1]{twocol}}%
9656 \newcommand{\footthreecol}[1]{\Xarrangement[#1]{threecol}}%
9657 \let\hsizetwocol\Xhsizetwocol
9658 \let\hsizethreecol\Xhsizethreecol
9659 \let\hookXnote\Xhooknote
9660 \let\boxsymlinenum\Xboxsymlinenum
9661 \let\symlinenum\Xsymlinenum
9662 \let\beforenumberinfootnote\Xbeforenumber
9663 \let\afternumberinfootnote\Xafternumber
9664 \let\beforeXsymlinenum\XbeforeXsymlinenum
9665 \let\afterXsymlinenum\XafterXsymlinenum
9666 \let\inplaceofnumber\Xinplaceofnumber
9667 \let\Xlemmaseparator\lemmaseparator
9668 \let\afterlemmaseparator\Xafterlemmaseparator
9669 \let\beforelemmaseparator\Xbeforelemmaseparator
9670 \let\inplaceoflemmaseparator\Xinplaceoflemmaseparator
9671 \let\txbeforeXnotes\Xtxbeforenotes
9672 \let\afterXrule\Xafterrule
9673 \let\numberonlyfirstinline\Xnumberonlyfirstinline
9674 \let\numberonlyfirstintwoline\Xnumberonlyfirstintwoline
9675 \let\nonumberinfootnote\Xnonumberinfootnote
9676 \let\pstartinfootnote\Xpstart
9677 \let\pstartinfootnoteeverytime\Xpstarteverytime
9678 \let\onlyXpstart\Xonlypstart
9679 \let\Xnonumberinfootnote\Xnonumber
9680 \let\nonbreakableafternumber\Xnonbreakableafternumber
9681 \let\maxhXnotes\Xmaxhnotes
9682 \let\beforeXnotes\Xbeforenotes
9683 \let\boxlinenum\Xboxlinenum
9684 \let\boxlinenumalign\Xboxlinenumalign
9685 \let\boxstartlinenum\Xboxstartlinenum
9686 \let\boxendlinenum\Xboxendlinenum
9687 \let\twoline\Xtwoline
9688 \let\morethantwoline\Xmorethantwoline
9689 \let\twolinebutnotmore\Xtwolinebutnotmore
9690 \let\twolineonlyinsamepage\Xtwolineonlyinsamepage
9691 \fi
9692
9693 \unless\ifnofamiliar@
9694 \let\notesXwidthliketwocolumns\noteswidthliketwocolumnsX
9695 \fi
9696 \newcommandx{\parafootsep}[2][1,usedefault]{%
9697 \Xparafootsep[#1]{#2}%
9698 \parafootsepX[#1]{#2}
9699 }%
9700
9701 \newcommandx{\afternote}[2][1,usedefault]{%
9702 \Xafternote[#1]{#2}%
9703 \afternoteX[#1]{#2}%
9704 }%

```

```
9705 \unless\ifnoend@
9706   \let\XendXtwolines\Xendtwolines
9707   \let\XendXmorethantwolines\Xendmorethantwolines
9708   \let\hookXendnote\Xendhooknote
9709   \let\boxXendlinenum\Xendboxlinenum%
9710   \let\boxXendlinenumalign\Xendboxlinenumalign%
9711   \let\boxXendstartlinenum\Xendboxstartlinenum%
9712   \let\boxXendendlinenum\Xendboxendlinenum%
9713   \let\XendXlemmaseparator\Xendlemmaseparator
9714   \let\XendXbeforelemmaseparator\Xendbeforelemmaseparator
9715   \let\XendXafterlemmaseparator\Xendafterlemmaseparator
9716   \let\XendXinplaceoflemmaseparator\Xendinplaceoflemmaseparator
9717 \fi
9718
9719 \AtBeginDocument{%
9720   \ifdef\lineref{}\let\lineref\edlineref}%
9721 }%
9722
9723
9724 \fi%
9725 %
9726
```

</code>

Appendix A Things to do when changing versions

Appendix 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:

<code>I saw my friend \critext{Smith}</code>	1 I saw my friend
<code>\Afootnote{Jones C, D.}/</code>	2 Smith on Tuesday.
on Tuesday.	<u>2 Smith]</u> Jones C, D.

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

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

<code>\critext{I saw my friend</code>	1 I saw my friend
<code>\critext{Smith}{\Afootnote{Jones</code>	2 Smith on Tuesday.
<code>C, D.}/ on Tuesday.}</code>	<u>2 Smith]</u> Jones C, D.
<code>\Bfootnote{The date was</code>	
<code>July 16, 1954.}</code>	1-2 I saw my friend
<code>/</code>	Smith on Tuesday.] The
	date was July 16, 1954.

However, `\critext` cannot handle overlapping but unnnested 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. 130 to find out why.) Aliases of the second kind may be nested without any problem.

If you really have to use `\critext` in any of the tabular or array environments, then `\edtext` must not be used in the same environment. If you use `\critext` in one of these environments then you have to issue the declaration `\usingcritext` beforehand. The declaration `\usingedtext` must be issued to revert to the default assumption that `\edtext` will be used.

Appendix 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. 36), 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. 37) or `\Xfootnote` options (6.2.2 p. 25). 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.

Appendix A.3 Migration to eledmac 1.5.1

The version 1.5.1 corrects a bug in `stanzaindentsrepetition` (cf. 9.3 p. 52). 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}
```

Appendix A.4 Migration to eledmac 1.12.0

The migration to eledmac 1.12.0 is easy:

- You must first delete all the auxiliary files, then compile your document three times as usual.
- If you have modified `\l@reg`, which is not advisable, you must rename it to `\@nl@reg`.

There is an additional problem. If you have put text into brackets just after `\pstart` or `\pend`, this text will be considered to be an optional argument of `\pstart` or `\pend` (see 5.2.3 p. 18). If so, add a `\relax` between `\pstart`/`\pend` and the first bracket.

The version 1.12.0 also introduces a better way to handle sectional divisions inside numbered text. Please read 16.2 p. 71.

Appendix A.5 Migration to eledmac 17.1

This version changes the default setting of `\Xpstart`. Henceforth, `pstart` numbers will be printed in footnotes within the section of text where you have called `\numberpstarttrue`.

We do not see any reason to print them in the other sections. However, if you want to print the `pstart` numbers in all of the footnotes, whatever the section, without having to use `\numberpstarttrue`, you can use `\Xpstarteverytime`.

Appendix A.6 Migration to eledmac 1.21.0

Appendix 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.

Appendix A.6.2 Endnotes

In any case, delete the `.end` file before the next run.

The previous version of Eledmac had a bug: there were two spaces between the starting page number and the starting line number, but only one space between the ending page number and the ending line number.

As a matter of fact, a spurious space was added after the first `\printnpnum`. This spurious space has been deleted. However, if you want to keep the previous spurious space, you may load the package with the `oldprintnpnumspace` option.

If you have redefined `\endprint`, you must:

- Contact us and ask for the feature that required your hack, in order to avoid such a hack in the future.
- Use the new fifth argument.
- Add `\xdef\@currentseries{#4}` at the beginning of your own command.

Appendix 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`.

Appendix A.8 Migration to eledmac 1.23.0

You must delete the numbered auxiliary files before compiling with the new version of eledmac.

Appendix A.9 Migration from eledmac to reledmac

There are many changes in reledmac which require the user to make modifications.

Appendix A.9.1 Risk of ‘no room for a new’

The risk to obtain a ‘no room for a new something’ error is greater in reledmac than it is in eledmac. See 19.1.3 p. 74 in order to know how to limit it.

Appendix 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.

Appendix A.9.3 Deprecated commands and options

The table of deprecated commands and their alternatives follows. Note that the way some commands must be used may have changed. Please read the handbook.

<i>Deprecated command</i>	<i>Replaced with</i>
<code>\addfootins</code>	<code>\newseries</code>
<code>\addfootinsX</code>	<code>\newseries</code>
<code>\critext</code>	<code>\edtext</code>
<code>\falseverse</code>	<code>\newverse</code>
<code>\interparanoteglu</code>	<code>\Xafternote</code> and <code>\afternoteX</code>
<code>\ledchapter</code>	<code>\eledchapter</code>
<code>\ledsection</code>	<code>\eledsection</code>
<code>\ledsetnormalparstuff</code>	<code>\Xledsetnormalparstuff</code> and <code>\ledsetnormalparstuffX</code>
<code>\ledsubsection</code>	<code>\eledsubsection</code>
<code>\ledsubsubsection</code>	<code>\eledsubsubsection</code>
<code>\noeledsec</code>	Package option <code>noeledsec</code>
<code>\noendnotes</code>	Package option <code>noendnotes</code>
<code>\pageparbreak</code>	<code>\ledpb</code>

The `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.

Appendix 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>\symplinum</code>	<code>\Xsymplinum</code>

Appendix 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>\afterXsymplinum</code>	<code>\Xaftersymplinum</code>
<code>\beforelemmaseparator</code>	<code>\Xbeforelemmaseparator</code>
<code>\beforenumberinfootnote</code>	<code>\Xbeforenumber</code>
<code>\beforeXnotes</code>	<code>\Xbeforenotes</code>
<code>\beforeXsymplinum</code>	<code>\Xbeforesymplinum</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>

Appendix A.9.6 Endnotes

With `reledmac`, there is now one auxiliary file for every endnotes set (`.Aend`, `.Bend`, `.Cend` etc.). If you have overridden `\doendnotes` (which you would not have done) you must adapt your code.

Appendix 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.

Appendix 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.

Appendix A.10 Migration to `reledmac 2.1.0`

`Reledmac 2.1.0` fix some bugs when using `\Xbhooknote` and `\bhooknoteX` not in order to execute code at the beginning of each notes, but to insert content of at the beginning of each notes.

People who use these commands to do it, which is not the original idea, must change the following:

1. Horizontal space is no longer automatically added after the content of the `\Xbhooknote/\bhooknoteX` argument. You must include it manually. So instead of `\Xbhooknote{content}`, use `\Xbhooknote{content }`.
2. Indent is no longer automatically added before the content of the `\Xbhooknote/\bhooknoteX` argument. If you want to keep it, add `\indent` in the argument of `\Xbhooknote/\bhooknoteX`.

Appendix 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`.

Appendix A.12 Migration to `reledmac 2.3.0`

Before `reledmac 2.3.0`, for typesetting verse, any empty line was considered a paragraph inside verses. Counting empty lines this created breaking verse, hanging verses, and also added spurious vertical spaces. Version 2.3.0 disables paragraph in stanza. If you want vertical space, use optional argument of `\stanza` or `\endverse`.

Appendix A.13 Migration to reledmac 2.4.0

It is not mandatory, but strongly recommended, to change any `\renewcommand{\endashchar}{...}` to the use of `\Xlinerangeseparator` or `/` and `\Xendlinerangeseparator` (7.2.4 p. 40).

Appendix A.14 Migration to reledmac 2.5.0

It is strongly recommended to stop redefining `\printnpnum` and to use the hooks documented in 7.2.3 p. 40.

`\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.

Appendix 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.

Appendix 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.

Appendix 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 `\Xparinden` and `\parindentX`.

Appendix 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.

Appendix 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.

Appendix 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.}}
```

Appendix A.21 Migration to `reledmac` 2.24.0

When using `\labelpstarttrue`, a spurious space^a was added after the `pstart` number (only on normal typesetting, not on parallel typesetting). The new version of the package delete this spurious space. If you consider that it was NOT a spurious space, you should add it manually in your definition of `\thepstart`.

Appendix A.22 Migration to `reledmac` 2.26.0

You must delete yours `.aux` file after having upgraded to this new version of `reledmac`.

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>	51
<code>\@EDROWFILL@</code>	1
<code>\@adv</code>	1
<code>\@advancestanzanumber</code>	1
<code>\@annot</code>	1
<code>\@annot@end@print</code>	1
<code>\@annot@start@print</code>	1
<code>\@beforeinsertofthisedtext</code>	1
<code>\@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>\@changingsymbol</code>	1
<code>\@iiiminipage</code>	1
<code>\@insertstanza</code>	1
<code>\@k</code>	1
<code>\@l@dtempcnta</code>	1
<code>\@l@dtempcntb</code>	1
<code>\@lab</code>	1
<code>\@led@testifnofoot</code>	1
<code>\@lemma</code>	1
<code>\@line@num</code>	1
<code>\@lock</code>	1
<code>\@lopL</code>	1
<code>\@lopR</code>	1
<code>\@makechapterhead</code>	1
<code>\@makeschapterhead</code>	1
<code>\@mem@extranofeet</code>	1
<code>\@mem@old@ssect</code>	1
<code>\@mpfnpos</code>	1
<code>\@msd</code>	1
<code>\@msd@c</code>	1
<code>\@msd@options@iffullpage</code>	1
<code>\@msdata@list</code>	1
<code>\@nl</code>	1
<code>\@nl@reg</code>	1
<code>\@opXfeet</code>	1
<code>\@pend</code>	1
<code>\@pendR</code>	1
<code>\@ref</code>	1
<code>\@ref@reg</code>	1
<code>\@ref@reg@parse</code>	1
<code>\@sect</code>	1
<code>\@series</code>	1
<code>\@set</code>	1
<code>\@sidenotesep</code>	1
<code>\@ssect</code>	1
<code>\@startstanza</code>	1
<code>\@stopmsd</code>	1
<code>\@stopstanza</code>	1

<code>\@sw</code>	1
<code>\@tag</code>	1
<code>\@wredindex</code>	1
<code>\@xloop</code>	1
<code>\@xympar</code>	1
<code>CLASSarticle</code>	74
<code>CLASSbook</code>	74, 359
<code>CLASSmemoir</code>	211, 212, 274, 275, 309, 359, 379, 440, 444
<code>CLASSscrbook</code>	444
<code>COMMAND*footnote</code>	75
<code>COMMAND\...@footnotemark...</code>	215
<code>COMMAND\...d@ta</code>	157
<code>COMMAND\<hook</code>	
<code>@<series</code>	262
<code>COMMAND\<hookname</code>	
<code><pseudoserries</code>	264
<code>COMMAND\<type</code>	
<code>footfmt</code>	200
<code>COMMAND\@@line</code>	191
<code>COMMAND\@MM</code>	178, 441
<code>COMMAND\@Rlineflag</code>	312, 441
<code>COMMAND\@SErefprefix</code>	288
<code>COMMAND\@SErefprefixmore</code>	288
<code>COMMAND\@add@</code>	349
<code>COMMAND\@adv</code>	114
<code>COMMAND\@annot</code>	163
<code>COMMAND\@annot@end@print</code>	203
<code>COMMAND\@annot@start@print</code>	203
<code>COMMAND\@apprefprefixmore</code>	288
<code>COMMAND\@apprefprefixsingle</code>	288
<code>COMMAND\@beforeinsertofthisedtext</code>	137
<code>COMMAND\@bsphack</code>	277
<code>COMMAND\@doclearpage</code>	275, 434, 444
<code>COMMAND\@doreinfeetX</code>	444
<code>COMMAND\@dprintingcolumns</code>	441
<code>COMMAND\@edindex@hyperref</code>	312, 313
<code>COMMAND\@edtext@level</code>	134
<code>COMMAND\@esphack</code>	277
<code>COMMAND\@firstofone</code>	33
<code>COMMAND\@fnpos</code>	233, 271
<code>COMMAND\@footnotemark</code>	212, 434, 444
<code>COMMAND\@footnotetext</code>	212, 213, 434
<code>COMMAND\@gobble</code>	32, 58, 132, 133, 163, 256
<code>COMMAND\@gobblefive</code>	443
<code>COMMAND\@gobblefour</code>	440
<code>COMMAND\@gobbleseven</code>	258
<code>COMMAND\@gobblethree</code>	433
<code>COMMAND\@h</code>	195
<code>COMMAND\@hangingsymbol</code>	317
<code>COMMAND\@iiminipage</code>	301, 303, 433, 444

COMMAND\@iiiminpage	301
COMMAND\@l	439
COMMAND\@l@tempcnta	160, 164, 172
COMMAND\@l@tempcntb	164
COMMAND\@l@reg	439
COMMAND\@lab	110, 276, 280, 284, 433
COMMAND\@ldunboxmpfoot	303
COMMAND\@led@extranofeet	380
COMMAND\@ledinnote@command	307, 308
COMMAND\@lemma	137, 139
COMMAND\@lock	104, 317
COMMAND\@lopL	434
COMMAND\@lopR	434
COMMAND\@makecol	270, 271, 274, 444
COMMAND\@mpfnpos	233
COMMAND\@msd	326, 327
COMMAND\@msd@c	326
COMMAND\@msd@options@iffullpage	333
COMMAND\@msdata@list	327, 328
COMMAND\@nl	110–112, 114, 115, 125, 163, 280, 433, 434
COMMAND\@nl@reg	111, 377, 434, 439
COMMAND\@opXfeet	434
COMMAND\@opfeetX	444
COMMAND\@opxtrafeeti	444
COMMAND\@page	112, 280
COMMAND\@pend	434
COMMAND\@pendR	434
COMMAND\@ref	110, 120–122, 126, 132
COMMAND\@ref@later	122, 127
COMMAND\@ref@reg	121, 163, 434
COMMAND\@ref@reg@parsearg	122
COMMAND\@reinserts	270, 272, 274, 444
COMMAND\@secondoftwo	76
COMMAND\@sect	362
COMMAND\@series	260
COMMAND\@set	115
COMMAND\@sidenoteseq	299
COMMAND\@stopmsd	327
COMMAND\@sw	122, 140, 144, 145
COMMAND\@tag	133, 135, 138
COMMAND\@tempcnta	89
COMMAND\@tempcntb	89
COMMAND\@toksa	96
COMMAND\@toksb	96
COMMAND\@xloop	173
COMMAND\@xympar	293, 444
COMMAND\Aendnote	16, 25
COMMAND\Afootfmt	177
COMMAND\Afootgroup	177
COMMAND\Afootnote	8, 16, 24, 25, 28, 135, 186, 211, 234, 251, 443

COMMAND\Afootstart	177
COMMAND\AtBeginDocument	274
COMMAND\AtEndEveryPend	19, 450
COMMAND\AtEveryPend	18, 19, 54, 151, 441, 442, 444, 450
COMMAND\AtEveryPend*	19
COMMAND\AtEveryPstart	18, 19, 54, 441, 442, 444, 447, 450
COMMAND\AtEveryPstart*	19
COMMAND\AtEveryStanza	54, 448, 450
COMMAND\AtEveryStopStanza	54, 448, 450
COMMAND\AtStartEveryPstart	19, 450
COMMAND\AtStartEveryStanza	54, 450
COMMAND\BeforeEveryStopStanza	450
COMMAND\Bendnote	16, 24
COMMAND\Bfootnote	8, 16, 211, 234, 251
COMMAND\Centering	47
COMMAND\Cfootnote	211
COMMAND\Columns	89, 183
COMMAND\Dfootnote	211
COMMAND\Efootnote	211
COMMAND\Gls	67
COMMAND\Hy@raisedlink	372
COMMAND\Hy@raisedlink@left	372
COMMAND\LTR	47
COMMAND\NR	362
COMMAND\Pages	89, 271, 273
COMMAND\ProcessOptionsX	79
COMMAND\RL	46
COMMAND\RaggedLeft	47
COMMAND\RaggedRight	47
COMMAND\RenewExpandableDocumentCommand	33, 143
COMMAND\SEonlypage	286, 446
COMMAND\SEref	61–63, 286, 289, 447, 449
COMMAND\SErefonlypage	61–63, 383, 446
COMMAND\SErefwithpage	61, 63, 286, 289, 383, 446, 448
COMMAND\Stanza	439
COMMAND\Waklam	350
COMMAND\X@doreinfect	272, 444
COMMAND\XR@prefix	292
COMMAND\XR@test	292
COMMAND\XR@test@mac	292
COMMAND\XR@test@mac@test	292
COMMAND\XXXXXXfmt	376
COMMAND\XXXXXfmt	376
COMMAND\Xafterenumber	42
COMMAND\Xafterlemmaseparator	44, 380
COMMAND\Xafternote	48, 379, 380
COMMAND\Xafternumber	42, 380
COMMAND\Xafterrule	49, 234, 380, 439, 442
COMMAND\Xaftersymlinenum	42, 380
COMMAND\Xarrangement	38, 48, 75, 178, 179, 262, 380

COMMAND\Xarrangement@footparagraph	184
COMMAND\Xarrangement@normal	179
COMMAND\Xarrangement@paragraph	184
COMMAND\Xbeforeinserting	47
COMMAND\Xbeforelemmaseparator	44, 380
COMMAND\Xbeforenotes	49, 234, 380, 439, 442
COMMAND\Xbeforenumber	40, 42, 380
COMMAND\Xbeforesymlinenum	42, 380
COMMAND\Xbhookendnote	381
COMMAND\Xbhookgroup	49, 446, 447
COMMAND\Xbhooknote	46, 381, 382, 445
COMMAND\Xboxendlinenum	43, 381, 443
COMMAND\Xboxlinenum	43, 381
COMMAND\Xboxlinenumalign	43, 381, 443
COMMAND\Xboxstartlinenum	43, 381, 443
COMMAND\Xboxsymlinenum	43, 381
COMMAND\Xcolalign	47, 442
COMMAND\Xdo@feet	444, 449
COMMAND\Xend	258
COMMAND\XendXafterlemmaseparator	381
COMMAND\XendXbeforelemmaseparator	381
COMMAND\XendXinplaceoflemmaseparator	381
COMMAND\XendXlemmaseparator	381
COMMAND\XendXmorethantwolines	381
COMMAND\XendXtwolines	381
COMMAND\Xendafterenumber	42, 445
COMMAND\Xendafterlemmaseparator	44, 381
COMMAND\Xendafternote	50, 447
COMMAND\Xendafternumber	44
COMMAND\Xendafterpagenumbe	448
COMMAND\Xendafterpagenumber	40, 63
COMMAND\XendafterpagenumberSErefonlypage	383
COMMAND\Xendaftersymlinenum	42, 44, 445
COMMAND\Xendahookinplaceofnumber	44, 445
COMMAND\Xendahooklinenum	44, 445
COMMAND\Xendbeforelemmaseparator	44, 381
COMMAND\Xendbeforelinenum	44
COMMAND\Xendbeforenumber	42, 445
COMMAND\Xendbeforepagenumber	40, 62, 63
COMMAND\XendbeforepagenumberSErefonlypage	62
COMMAND\Xendbeforesymlinenum	42, 44, 445
COMMAND\Xendbhookinplaceofnumber	44, 445
COMMAND\Xendbhooklinenum	44, 445
COMMAND\Xendbhooknote	46
COMMAND\Xendboxendlinenum	43, 443
COMMAND\Xendboxlinenum	43, 381, 441
COMMAND\Xendboxlinenumalign	43, 381, 443
COMMAND\Xendboxstartlinenum	43, 443
COMMAND\Xendboxsymlinenum	43, 445
COMMAND\Xendhangindent	46, 445, 447

COMMAND\Xendinplaceoflemmaseparator	26, 45, 381
COMMAND\Xendinplaceofnumber	42, 444
COMMAND\Xendinplaceofpagenumber	40, 449
COMMAND\Xendinsertsep@	241
COMMAND\Xendlemmadisablefontselection	45
COMMAND\Xendlemmafont	45, 447
COMMAND\Xendlemmaseparator	26, 44, 381
COMMAND\Xendletboxendlinenum	381
COMMAND\Xendlineflag	63
COMMAND\Xendlinenumannotationposition	57, 63
COMMAND\Xendlineprefixmore	40, 63
COMMAND\Xendlineprefixsingle	40, 63
COMMAND\Xendlinerangeseparator	40, 63, 198, 383, 446
COMMAND\Xendmorethantwolines	26, 41, 63, 381, 442, 443
COMMAND\Xendnonumber	41, 444
COMMAND\Xendnote	237, 257, 258, 442
COMMAND\Xendnotefontsize	45, 380
COMMAND\Xendnotenumfont	44, 45, 380
COMMAND\Xendnotes	240
COMMAND\Xendnumberonlyfirstinline	39, 445
COMMAND\Xendnumberonlyfirstintwolines	39, 445
COMMAND\Xendpagenumberonlyfirst	39, 449
COMMAND\Xendpagenumberonlyfirstifsingle	39, 449
COMMAND\Xendpagenumberonlyfirstintwo	39, 449
COMMAND\Xendparagraph	50, 439
COMMAND\Xendsep	50
COMMAND\Xendsublinesep	42, 63, 198
COMMAND\Xendsymlinenum	39, 445
COMMAND\Xendsympagenum	40, 449
COMMAND\Xendtwolines	26, 41, 63, 381, 442, 443
COMMAND\Xendtwolinesbutnotmore	41, 63, 442, 443
COMMAND\Xendtwolinesonlyinsamepage	41, 63, 442, 443
COMMAND\Xendwrapcontent	46, 448
COMMAND\Xendwraplemma	46, 448
COMMAND\Xendwraplinenumannotation	57, 63
COMMAND\Xfootfmt	382
COMMAND\Xfootgroup	183
COMMAND\Xfootins	182
COMMAND\Xfootnote	59, 66, 133, 376, 436, 440, 442, 446, 448
COMMAND\Xfootstarts	183
COMMAND\Xgroupbyline	48, 170, 210
COMMAND\Xgroupbylines	449
COMMAND\Xgroupbylineseparetwolines	48
COMMAND\Xhangindent	46, 445
COMMAND\Xhsize	383, 446, 447
COMMAND\Xhsizethreecol	47, 50, 381
COMMAND\Xhsizetwocol	47, 50, 264, 381
COMMAND\Xinplaceoflemmaseparator	25, 44, 381
COMMAND\Xinplaceofnumber	42, 381, 442, 443
COMMAND\Xinsertparafootsep	189, 190

COMMAND\Xledsetnormalparstuff	378, 379, 442
COMMAND\Xlemmadisablefontselection	45
COMMAND\Xlemmafnt	45, 447
COMMAND\Xlemmaseparator	44, 198, 265, 267, 270, 381
COMMAND\Xlineflag	62, 383, 384, 448
COMMAND\Xlinenumannotationposition	57, 62, 162
COMMAND\Xlinenumannotationpositionside	57
COMMAND\Xlinerangeseparator	40, 62, 198, 383, 446
COMMAND\Xmaxhnotes	49, 74, 75, 380, 381, 439, 441
COMMAND\Xmorethantwolines	25, 40, 41, 62, 381, 441
COMMAND\Xnoindent	445
COMMAND\Xnolemmaseparator	44, 270, 381
COMMAND\Xnonbreakableafternumber	42, 381, 437
COMMAND\Xnonumber	41, 381
COMMAND\Xnonumberinfootnote	381
COMMAND\Xnotefontsize	45, 380
COMMAND\Xnotefontsize@⟨s⟩	189, 193, 194
COMMAND\Xnotenumfont	45, 380
COMMAND\Xnoteswidthliketwocolumns	50, 440
COMMAND\Xnumberonlyfirstinline	39, 40, 48, 107, 200, 264, 265, 267, 381, 436, 441, 450
COMMAND\Xnumberonlyfirstintwolines	39, 48, 381, 436, 450
COMMAND\Xonlypstart	41, 381, 436, 441
COMMAND\Xpagelinesep	42, 449
COMMAND\Xparafootsep	48, 107, 380, 381, 448, 450
COMMAND\Xparafootsep@series	189
COMMAND\Xparinden	383
COMMAND\Xparindent	46, 442, 445, 447
COMMAND\Xprenotes	49, 234, 448
COMMAND\Xprenotes@	182, 234, 436
COMMAND\Xpstart	41, 378, 381, 436, 441
COMMAND\Xpstarteverytime	41, 378, 381, 441
COMMAND\Xragged	48
COMMAND\Xrigidbalance	191, 383, 446
COMMAND\Xstanza	41, 42, 55
COMMAND\Xstanzaseparator	42
COMMAND\Xstorelineinfo	200
COMMAND\Xsublinesep	22, 42, 62, 198, 383
COMMAND\Xsublinesepside	22, 42
COMMAND\Xsymlinenum	39, 48, 380, 381, 443, 450
COMMAND\Xtextbeforenotes	173
COMMAND\Xtoendnotes	27, 258
COMMAND\Xtwolines	25, 40, 41, 62, 206, 207, 264, 381, 441
COMMAND\Xtwolinesappref	264
COMMAND\Xtwolinesbutnotmore	41, 62, 381, 442
COMMAND\Xtwolinesbutnotmoreappref	264
COMMAND\Xtwolinesonlyinsamepage	41, 62, 381, 442
COMMAND\Xtxtbeforenotes	48, 49, 381, 448, 449, 451
COMMAND\Xtxtbeforenotesonlyonce	48, 451
COMMAND\Xunvxh	187, 382
COMMAND\Xwidth	50, 383, 447

COMMAND\Xwrapcontent	46, 448
COMMAND\Xwraplemma	46, 47, 448
COMMAND\Xwraplinenumannotation	57, 62, 163
COMMAND\Xwraplinenumannotation@ref	163
COMMAND\Xwraplinenumannotation@side	163
COMMAND\Xwraplinenumannotationref	57
COMMAND\Xwraplinenumannotation@side	57
COMMAND\&	380
COMMAND\	57
COMMAND\absline@num	104, 159
COMMAND\accent	132
COMMAND\actionlines@list	105, 161
COMMAND\actions@list	105
COMMAND\add@Xgroupbyline	170
COMMAND\add@inserts	105, 170
COMMAND\add@inserts@next	170
COMMAND\add@msd@	326
COMMAND\add@msdata	327, 328
COMMAND\add@msdata@firstlineofpage	330
COMMAND\add@msddata	326
COMMAND\add@penalties	159, 171
COMMAND\addcontentsline	357
COMMAND\addfootins	376, 379
COMMAND\addfootinsX	376, 379
COMMAND\addtoendnotes	258
COMMAND\advancelabel@refs	278
COMMAND\advanceline	23, 106, 114, 128, 444
COMMAND\affixlin@num	299
COMMAND\affixline@num	164, 167, 168, 434
COMMAND\affixpstart@num	168
COMMAND\afterXrule	380
COMMAND\afterXsymlinenum	380
COMMAND\aftergroup	131, 136
COMMAND\afterlemmaseparator	380
COMMAND\afternote	380
COMMAND\afternoteX	48, 379, 380
COMMAND\afternumberinfootnote	380
COMMAND\afterruleX	49, 439, 442
COMMAND\annot@list	106, 163
COMMAND\annotationref	57
COMMAND\annotref	59
COMMAND\applabel	61, 280–282, 442, 448
COMMAND\appref	59, 61–63, 286, 289, 446, 447
COMMAND\apprefprefixmore	62, 380
COMMAND\apprefprefixsingle	62, 380
COMMAND\apprefwithpage	61, 63, 286, 289, 443, 446
COMMAND\arrangementX	38, 75, 215, 262, 380
COMMAND\arrangementX@normal	221
COMMAND\article	15
COMMAND\at@every@pend	151

COMMAND\autopar	18, 148, 152, 153, 231, 435, 437, 438, 442
COMMAND\ballast	75
COMMAND\ballast@count	159, 172
COMMAND\baselineskip	38, 185, 189
COMMAND\beforeXnotes	380
COMMAND\beforeXsymlinenum	380
COMMAND\beforeelectedchapter	10, 72, 358
COMMAND\beforeinsertingX	47
COMMAND\beforelemmaseparator	380
COMMAND\beforenotesX	49, 382, 438, 439, 442
COMMAND\beforenumberinfootnote	380
COMMAND\begin	334
COMMAND\beginnumbering	17, 19, 20, 90, 91, 94, 95, 103, 108, 124, 152, 237, 326, 436, 439, 443, 444, 449, 450
COMMAND\beginnumberingR	90
COMMAND\bf	436
COMMAND\bfseries	45, 436
COMMAND\bhookXnote	381
COMMAND\bhookgroupX	49, 446
COMMAND\bhooknoteX	46, 382, 445
COMMAND\body	318
COMMAND\bodyfootmarkA	35
COMMAND\book	15
COMMAND\boxXendlinenum	381
COMMAND\boxXendlinenumalign	381
COMMAND\boxXendstartlinenum	381
COMMAND\boxendlinenum	381
COMMAND\boxlinefootnote	202
COMMAND\boxlinenum	381
COMMAND\boxlinenumalign	381
COMMAND\boxstartlinenum	381
COMMAND\boxsymlinenum	381
COMMAND\break	38, 187
COMMAND\brokenpenalty	172
COMMAND\centering	47
COMMAND\ch@ck@l@ck	434
COMMAND\ch@cksub@l@ck	167, 434
COMMAND\chapter	71, 359, 363, 439, 442, 444, 448
COMMAND\chaptermark	357
COMMAND\check@pb@in@verse	371
COMMAND\colalignX	47, 442
COMMAND\collect@body	335
COMMAND\color	448
COMMAND\colorbox	76
COMMAND\columns	50
COMMAND\columnwidth	185, 440
COMMAND\command names	264
COMMAND\copyright	132
COMMAND\correct@Xfootins@box	441
COMMAND\correct@footinsX@box	441

COMMAND\count	192, 193
COMMAND\critex	435
COMMAND\critext	139, 375, 376, 379
COMMAND\csname	80, 142
COMMAND\csquotes	255
COMMAND\ctab	351, 356
COMMAND\ctabtext	356
COMMAND\current@annot	163
COMMAND\dcoll	344
COMMAND\def	77, 93
COMMAND\detokenize	142
COMMAND\dimen	193
COMMAND\dimexpr	50
COMMAND\discretionary	186
COMMAND\displaywidowpenalty	171
COMMAND\do@Xfeet	271, 434, 444, 449
COMMAND\do@actions	159, 161, 434
COMMAND\do@actions@fixedcode	434
COMMAND\do@actions@next	159, 161
COMMAND\do@ballast	159, 172
COMMAND\do@feet@custom@order	271
COMMAND\do@insidelinehook	437
COMMAND\do@line	105, 131, 150, 153, 157, 158, 170, 171, 317, 434, 435, 437, 439
COMMAND\do@linehook	434
COMMAND\do@lockoff	106
COMMAND\do@lockon	106
COMMAND\do@reinxtrafeet	433
COMMAND\doendnotes	26, 241, 382, 443, 450
COMMAND\doendnotesbysection	26, 241, 259, 443, 450
COMMAND\doenotes	450
COMMAND\doinsidelinehook	24, 440
COMMAND\dolinehook	24, 440
COMMAND\doreinxtrafeeti	444
COMMAND\doreinxtrafeetii	444
COMMAND\doxtrafeet	270, 433
COMMAND\doxtrafeeti	444
COMMAND\doxtrafeetii	444
COMMAND\dummy@ref	132
COMMAND\edaftertab	70, 350–352
COMMAND\edannotationref	283
COMMAND\edatleft	70, 348
COMMAND\edatright	70, 349
COMMAND\edbeforetab	70, 350, 351
COMMAND\edfilldimen	349
COMMAND\edfont@info	138
COMMAND\edgls	67, 307
COMMAND\edglsadd	450
COMMAND\edgls...	447
COMMAND\edindex	33, 65–67, 306, 307, 311, 313, 339, 437, 440, 441, 444, 445, 449–451
COMMAND\edindexlab	67

- COMMAND\edlabel 59,
 61, 63, 132, 276, 278, 280, 282, 284, 292, 306, 339, 372, 384, 433, 436–438, 441, 446, 451
 COMMAND\edlabelE 61, 282
 COMMAND\edlabelS 61, 282
 COMMAND\edlabelSE 61
 COMMAND\edlineref 59, 276, 381, 441, 443, 446, 450
 COMMAND\edmakelabel 61, 292
 COMMAND\edpageref 59, 276, 282, 292
 COMMAND\edrowfill 350
 COMMAND\edsublineref 59
 COMMAND\edtabcolsep 344
 COMMAND\edtext 7, 24, 25, 27, 28,
 30–32, 51, 56, 59–61, 65, 68, 75, 104–106, 120, 122, 125–127, 130–139, 141, 142, 144, 146,
 162, 163, 280, 281, 285, 339, 341, 359, 372, 375, 376, 379, 433, 435, 437, 439–443, 449–451
 COMMAND\edtext@level 443
 COMMAND\edtextlater 123
 COMMAND\edvertdots 71, 349
 COMMAND\edvertline 70, 71, 349
 COMMAND\elechapter 72
 COMMAND\eled@sectioning@out 364
 COMMAND\eledchapter 71, 379, 440, 444
 COMMAND\eledchapter* 71
 COMMAND\eledmac@error 433
 COMMAND\eledsection 7, 16, 71, 132, 156, 358, 379, 442, 450
 COMMAND\eledsection* 71
 COMMAND\eledsubsection 71, 379
 COMMAND\eledsubsection* 71
 COMMAND\eledsubsubsection 71, 379
 COMMAND\eledsubsubsection* 71
 COMMAND\eledxxx 10, 72, 364, 439
 COMMAND\eledxxxx 357
 COMMAND\elesection 364
 COMMAND\else 305, 358
 COMMAND\emph 33
 COMMAND\empty 89, 165, 277
 COMMAND\end 334
 COMMAND\end@lemmas 131
 COMMAND\endashchar 198
 COMMAND\endgraf 150, 189, 231
 COMMAND\endlock 22, 106, 129, 322
 COMMAND\endminipage 301, 303, 433, 444
 COMMAND\endmsdata 34
 COMMAND\endnotes 442, 446
 COMMAND\endnumbering 17, 20, 90, 91, 93, 94, 124, 434, 443, 449, 450
 COMMAND\endnumberingR 90
 COMMAND\endprint 237, 239, 258, 378
 COMMAND\endstanzaextra 380
 COMMAND\endsub 22, 106, 128
 COMMAND\endverse 382
 COMMAND\everypar 152

COMMAND\extensionchars	73, 90
COMMAND\externaldocument	63, 292
COMMAND\f@x@l@cks	434
COMMAND>falseverse	379, 437, 439
COMMAND\fi	358
COMMAND\firstlinenum	21, 164, 435
COMMAND\firstsublinenum	21, 435
COMMAND\fix@page	111, 112, 434
COMMAND\flag@end	126, 127, 137, 439
COMMAND\flag@end@RTL	127
COMMAND\flag@end@later	127
COMMAND\flag@start	126, 127, 137, 439, 440
COMMAND\flag@start@RTL	127
COMMAND\flag@start@later	127
COMMAND\flagstanza	58
COMMAND\floatingpenalty	178, 441
COMMAND\flush@notes	172, 173
COMMAND\fnpos	37, 233, 438, 449
COMMAND\footfmt	177, 180
COMMAND\footfmt...	216
COMMAND\footfootmarkA	35
COMMAND\footfudgefactor	187
COMMAND\footfudgefiddle	75, 185, 433
COMMAND\footgroup	177
COMMAND\footins	182
COMMAND\footnormal	263, 380, 433
COMMAND\footnormalX	380
COMMAND\footnote	35, 74, 211–213, 377, 434
COMMAND\footnote@lang	198
COMMAND\footnoteA	16, 35
COMMAND\footnoteB	16
COMMAND\footnoteC	24
COMMAND\footnoteE	35
COMMAND\footnoteX	8, 36, 255, 256, 449
COMMAND\footnoteX@reading	256
COMMAND\footnoteXmark	256, 450
COMMAND\footnoteXmk	270
COMMAND\footnoteXtext	256, 450
COMMAND\footnote⟨X⟩	133
COMMAND\footnote⟨X⟩mark	36
COMMAND\footnote⟨X⟩mk	36
COMMAND\footnote⟨X⟩nomk	36
COMMAND\footnote⟨X⟩text	36
COMMAND\footnotelang@lua	176
COMMAND\footnotelang@poly	176
COMMAND\footnotemark	36, 256
COMMAND\footnoteoption@	175, 445
COMMAND\footnoterule	192
COMMAND\footnotesize	45
COMMAND\footnotetext	36, 256

COMMAND\footparagraph	185, 263, 380, 439
COMMAND\footparagraphX	226, 380, 439
COMMAND\footsplitskips	434, 441
COMMAND\footstart	177, 182, 192
COMMAND\footstrut	189
COMMAND\foothreecol	380
COMMAND\foothreecolX	380, 443
COMMAND\foottwocol	380
COMMAND\foottwocolX	380, 443
COMMAND\foreignlanguage	46
COMMAND\fullstop	383
COMMAND\get@edindex@hyperref	312
COMMAND\get@edindex@ledinnote@command	308
COMMAND\get@fnmark	213
COMMAND\get@index@command	438
COMMAND\get@linelistfile	434
COMMAND\get@thisfootnote	220
COMMAND\getline@num	158, 160
COMMAND\gl@p	96
COMMAND\global	110
COMMAND\globaldefs	110
COMMAND\gls	67, 315
COMMAND\hangindentX	46, 442, 445
COMMAND\hangingsymbol	380, 435
COMMAND\hbox	186
COMMAND\hfill	438
COMMAND\hidenumbering	23, 119, 442
COMMAND\hidenumberingonleftpage	23, 119, 448
COMMAND\hidenumberingonrightpage	23, 119, 448
COMMAND\hline	68
COMMAND\hrulefill	350
COMMAND\hsize	39, 182, 185, 187, 193, 196, 232, 434, 440
COMMAND\hsizeX	383, 446, 447
COMMAND\hsizethreecol	381
COMMAND\hsizethreecolX	47, 50
COMMAND\hsizetwocol	381
COMMAND\hsizetwocolX	47, 50
COMMAND\hyperlinkR	311
COMMAND\hyperlinkformat	311
COMMAND\hyperlinkformatR	312
COMMAND\if@RTL	81, 135
COMMAND\if@edtext@	440, 443
COMMAND\if@eled@sectioning	364
COMMAND\if@firstlineofpage	81
COMMAND\if@firstlineofpageR	81
COMMAND\if@msd@options@fullpage	333
COMMAND\if@msdata@insertedfrompreviouspage	330
COMMAND\if@nobreak	151
COMMAND\if@noneed@Footnote	125
COMMAND\ifXnote@	90

COMMAND\ifafterendnumberingR	90
COMMAND\ifbypage@	97
COMMAND\ifbypage@R	97
COMMAND\ifbypstart@	97
COMMAND\ifbypstart@R	97
COMMAND\iffirst@linenum@out@	123, 124
COMMAND\ifindtl@innote	90
COMMAND\ifindtl@notenumber	90
COMMAND\ifinserthangingsymbol	317
COMMAND\ifinstanza	317
COMMAND\ifistwofollowinglines	207
COMMAND\ifl@d@Xmorethantwolines	203, 441
COMMAND\ifl@d@Xtwolines	203
COMMAND\ifl@d@dash	203
COMMAND\ifl@d@elin	203
COMMAND\ifl@d@esl	203
COMMAND\ifl@d@pnum	203
COMMAND\ifl@d@ssub	203
COMMAND\ifl@dend@X	257
COMMAND\ifl@dmemoir	433
COMMAND\ifl@dpaging	440
COMMAND\ifl@dpairing	89, 435
COMMAND\ifl@dprintingpages	441
COMMAND\ifl@dskipnumber	164
COMMAND\ifl@dstartendok	350
COMMAND\ifl@imakeidx	81
COMMAND\ifledRcol	89, 435
COMMAND\ifledRcol@	89, 439
COMMAND\iflemmacommand@	440
COMMAND\ifnoend@	241
COMMAND\ifnoledgroup@	305
COMMAND\ifnoteschanged@	107
COMMAND\ifnumberedpar@	147
COMMAND\ifnumbering	91, 94
COMMAND\ifnumberingR	89, 435
COMMAND\ifnumberline	137, 164
COMMAND\ifpst@rted	435
COMMAND\ifpst@rtedL	91
COMMAND\ifresumenummering@start	94
COMMAND\ifseriesbefore	261
COMMAND\ifstopmsdata@inserted@	326
COMMAND\ifsublines@	104, 116
COMMAND\iftrue	443
COMMAND\ifvmode	278
COMMAND\ifxxx	358
COMMAND\ignorespaces	135
COMMAND\imki@wrindexentry	81
COMMAND\immediate	123, 124, 236
COMMAND\indent	18, 152, 382
COMMAND\index	314, 449

COMMAND\indtl@wrindexentry	81
COMMAND\initnumbering@quote	356, 444
COMMAND\initnumbering@reg	434
COMMAND\initnumbering@sectcmd	444
COMMAND\inplaceoflemmaseparator	381
COMMAND\inplaceofnumber	381
COMMAND\insert	137, 169, 177, 180, 194, 210, 216
COMMAND\insert@Xtxtbeforenotes	174, 194
COMMAND\insert@count	120, 126, 135
COMMAND\insert@countR	135
COMMAND\insert@msdata	327, 333
COMMAND\insert@txtbeforenotesX	174
COMMAND\inserthangingsymbol	438
COMMAND\insertlines@list	105, 120, 121
COMMAND\insertparafootsepX	230
COMMAND\inserts@list	131, 148, 169, 170, 186
COMMAND\interAfootnotelinepenalty	435
COMMAND\interfootnotelinepenalty	435
COMMAND\interlinepenalty	177
COMMAND\interparanoteglue	379
COMMAND\justifying	47
COMMAND\l@advance@parledegroupp@beforenormalnotes	444
COMMAND\l@d@@wrindexhyp	440
COMMAND\l@d@add	140
COMMAND\l@d@end	237, 257
COMMAND\l@d@nums	135, 137, 139, 140, 203, 204
COMMAND\l@d@section	237
COMMAND\l@d@set	115, 129
COMMAND\l@dampcount	341
COMMAND\l@dbfnote	213, 434
COMMAND\l@dcheckstartend	350
COMMAND\l@dchset@num	115
COMMAND\l@dcolcount	341, 342
COMMAND\l@dcollect@@body	334
COMMAND\l@dcollect@body	334
COMMAND\l@dcsnote	439
COMMAND\l@dcsnotetext	157, 297
COMMAND\l@dcsnotetext@l	157, 297
COMMAND\l@dcsnotetext@r	157, 297
COMMAND\l@ddodoreinxtrafeet	272, 433
COMMAND\l@ddoxtrafeet	272, 433
COMMAND\l@emptyd@ta	435
COMMAND\l@dend@close	236
COMMAND\l@dend@open	236
COMMAND\l@dend@stuff	237
COMMAND\l@denvbody	334
COMMAND\l@dfeetbeginmini	433
COMMAND\l@dfeetendmini	433
COMMAND\l@dgetline@margin	435
COMMAND\l@dgetlock@disp	435

COMMAND\@dgetref@num	284, 285
COMMAND\@dgetsidenote@margin	293, 435
COMMAND\@dgobbeloptarg	440
COMMAND\@dgobblearg	440
COMMAND\@dgobbleoptarg	339
COMMAND\@dlabel@parse	284, 285
COMMAND\@dld@ta	164, 166
COMMAND\@dlp@rbox	298
COMMAND\@dlsn@te	435
COMMAND\@dlsnote	439
COMMAND\@dmake@labels	279, 292
COMMAND\@dmake@labelsR	292
COMMAND\@dnumpstartsL	91, 435
COMMAND\@dp@rsefootspec	204
COMMAND\@dparsefootspec	204
COMMAND\@dpush@begins	334
COMMAND\@drd@ta	164, 166
COMMAND\@dref@undefined	284
COMMAND\@drsn@te	435
COMMAND\@drsnote	439
COMMAND\@dtabaddcols	349
COMMAND\@dtabnoexpands	433
COMMAND\@dumboxmpfoot	444
COMMAND\@dunboxmpfoot	435
COMMAND\@dzeropenalties	435, 440
COMMAND\@pb	370
COMMAND\@prev@nopb	370
COMMAND\@prev@pb	370
COMMAND\@reg	377
COMMAND\@wrap@ifnotempty	88
COMMAND\label	19, 61, 63, 67, 277, 285
COMMAND\label@refs	276
COMMAND\labelstarttrue	19, 384, 436, 451
COMMAND\labelref@list	276, 280
COMMAND\language	186
COMMAND\last@page@num	434
COMMAND\lastbox	152
COMMAND\lastskip	128
COMMAND\latex@makecol	274
COMMAND\leavevmode	18, 152
COMMAND\led@check@nopb	370
COMMAND\led@check@pb	370
COMMAND\led@nopb	370, 371
COMMAND\led@nopbnum	370
COMMAND\led@pb	370, 371
COMMAND\led@pb@macro	370
COMMAND\led@pbnum	370
COMMAND\led@reinit@index@fornote	314
COMMAND\led@set@index@fornote	314
COMMAND\ledRflag	311

COMMAND\ledchapter	379, 437
COMMAND\ledfootinsdim	380
COMMAND\ledinnernote	64, 295, 439, 448
COMMAND\ledinnote	309, 378, 443
COMMAND\ledinnotemark	66, 378, 442
COMMAND\ledleftnote	64, 295
COMMAND\ledlinenum	102, 435
COMMAND\ledllfill	158
COMMAND\ledlsnotefontsetup	448
COMMAND\ledlsnotesep	64
COMMAND\ledlsnotewidth	64
COMMAND\lednopb	73, 370
COMMAND\lednopbinverse	371
COMMAND\lednopbinversetrue	54, 73
COMMAND\lednopbnum	370
COMMAND\ledouternote	64, 295, 439, 448
COMMAND\ledpb	73, 370, 379
COMMAND\ledpbnum	370
COMMAND\ledpbsetting	73, 370, 445
COMMAND\ledrightnote	64, 295
COMMAND\ledrsnotefontsetup	448
COMMAND\ledrsnotesep	64
COMMAND\ledrsnotewidth	64
COMMAND\ledsection	379
COMMAND\ledsectnomark	357
COMMAND\ledsectnotoc	357
COMMAND\ledsetnormalparstuff	378, 379, 442
COMMAND\ledsetnormalparstuff@common	231
COMMAND\ledsetnormalparstuffX	378, 379, 442
COMMAND\ledsidenote	64, 295, 297
COMMAND\ledsubsection	379
COMMAND\ledsubsubsection	379
COMMAND\ledxxx	439
COMMAND\left	70
COMMAND\leftctab	351
COMMAND\leftheadline	102
COMMAND\leftlinenum	22, 102, 433, 435
COMMAND\leftltab	350
COMMAND\leftnoteupfalse	64
COMMAND\leftpstartnum	168
COMMAND\lefttab	351
COMMAND\leftsidenote	297
COMMAND\leftskip	182, 185, 187
COMMAND\lemma	3, 25, 27, 28, 30–32, 130, 134, 135, 138, 139, 141, 375, 435, 436, 443, 444, 446
COMMAND\lemmaseparator	381
COMMAND\let	28, 51, 133, 137, 256, 274, 322, 433
COMMAND\letboxXendendlinum	381
COMMAND\line	191, 195
COMMAND\line@list	104, 106, 121, 137
COMMAND\line@list@stuff	91, 108, 124, 433, 435

COMMAND\line@list@version	110
COMMAND\line@margin	98, 166, 293
COMMAND\line@num	103, 104, 106, 164, 433
COMMAND\line@set	139, 140
COMMAND\lineannot	60, 140
COMMAND\lineannot@set	140
COMMAND\lineation	21, 97, 451
COMMAND\linebreak	38
COMMAND\lineneation	447
COMMAND\linenum	25, 27, 28, 56, 60, 61, 130, 139, 140, 282, 285, 292, 375
COMMAND\linenum@out	123, 276, 280
COMMAND\linenumannotation	56, 162
COMMAND\linenumannotation@side	162
COMMAND\linenumberlist	21, 89, 165, 433
COMMAND\linenumberstyle	23, 101, 433, 450
COMMAND\linenumincrement	21, 435
COMMAND\linenummargin	21, 98, 293
COMMAND\linenumr@p	101, 433, 435
COMMAND\linenumrep	101, 435
COMMAND\linenumsep	22, 64, 102, 294
COMMAND\linerangesep@	270
COMMAND\lineref	276, 282, 292, 381, 441
COMMAND\list@clear	96
COMMAND\list@clearing@reg	435
COMMAND\list@create	95, 96
COMMAND\lock@disp	100
COMMAND\lock@off	118
COMMAND\lock@on	117
COMMAND\lockdisp	22, 100
COMMAND\loop	173, 318
COMMAND\ltab	350, 351, 356
COMMAND\ltabtext	356
COMMAND\m@mmf@prepare	212
COMMAND\makeatletter	157
COMMAND\makehboxoffhboxes	188, 189
COMMAND\makeindex	65, 311
COMMAND\makelabel	292
COMMAND\managestanza@modulo	319
COMMAND\marginpar	64, 74, 293, 434
COMMAND\marginparwidth	64, 294
COMMAND\markboth	157
COMMAND\mathchardef	318
COMMAND\maxhXnotes	381
COMMAND\maxhnotesX	49, 75, 380, 438, 439, 441–443
COMMAND\maxlinesinpar@list	108
COMMAND\measurebody	352
COMMAND\measuretbody	353
COMMAND\memorybreak	20
COMMAND\metaXXXvfootnote	210
COMMAND\morenoexpands	75, 76, 131, 132

COMMAND\morethantwolines	381
COMMAND\mpfnpos	37, 233, 438, 449
COMMAND\mpnormalfootgroup	433
COMMAND\mpnormalvfootnote	434
COMMAND\msdata	34, 324, 326, 327, 450
COMMAND\msdataposition	35
COMMAND\multfootsep	35, 212
COMMAND\multiplefootnotemarker	212
COMMAND\musixtex	439
COMMAND\n@num	435, 442
COMMAND\n@num@ref	442
COMMAND\new@line	125, 434
COMMAND\new@series	133
COMMAND\newcommand	28, 77, 93, 212, 279
COMMAND\newcommandx	28, 29
COMMAND\newhookarg@specific	269
COMMAND\newhookcommand@series	264, 265, 442
COMMAND\newhookcommand@series@reload	265
COMMAND\newhookcommand@toggle@reload	264, 440
COMMAND\newhooktoggle@series	264, 442
COMMAND\newhooktoggle@specific	269
COMMAND\newif	442
COMMAND\newline	38
COMMAND\newlinechar	257
COMMAND\newseries	36, 376, 379, 380
COMMAND\newseries@	248, 249, 262
COMMAND\newverse	54, 58, 379, 439, 450
COMMAND\next	318
COMMAND\next@action	109
COMMAND\next@actionline	109
COMMAND\next@insert	170
COMMAND\next@line@list@stuff	124, 451
COMMAND\next@page@num	94
COMMAND\nl@regR	111
COMMAND\no@expands	75, 133, 138, 256, 433
COMMAND\noXlemmaseparator	381
COMMAND\nobreak	203
COMMAND\nocritical	249
COMMAND\noledsec	72, 379
COMMAND\noendnotes	379
COMMAND\noexpand	377
COMMAND\nofamiliar	267
COMMAND\noindent	18, 19, 54, 152, 445
COMMAND\noindentX	445
COMMAND\nomk@	270
COMMAND\nonbreakableafternumber	381
COMMAND\nonnumberinfootnote	381
COMMAND\norelax	51
COMMAND\normal@footnotemarkX	216
COMMAND\normal@page@break	369

COMMAND\normal@pars	231
COMMAND\normalbfnoteX	435
COMMAND\normalbodyfootmarkX	216
COMMAND\normalfont	449
COMMAND\normalfootfmt	51, 182, 189, 198, 237, 448
COMMAND\normalfootfmtX	217
COMMAND\normalfootfootmarkX	217
COMMAND\normalfootgroup	183
COMMAND\normalfootgroupX	218
COMMAND\normalfootnoterule	178
COMMAND\normalfootstart	182, 185
COMMAND\normalfootstartX	217
COMMAND\normalvfootnote	180, 181
COMMAND\normalvfootnote@inserted	180, 181
COMMAND\normalvfootnoteX	216
COMMAND\notbool	358
COMMAND\notfontsetup	380
COMMAND\notfontsizeX	45, 380
COMMAND\notenumfont	380
COMMAND\notenumfontX	45, 380
COMMAND\notesXwidthliketwocolumns	381
COMMAND\noteswidthliketwocolumnsX	50, 381, 440, 442
COMMAND\num@lines	147, 172
COMMAND\numberlinefalse	20
COMMAND\numberlinetrue	20
COMMAND\numberonlyfirstinline	262, 381
COMMAND\numberonlyfirstintwolines	381
COMMAND\numberpstartfalse	19
COMMAND\numberpstarttrue	19, 41, 378, 436, 444, 450
COMMAND\numberstanza	41
COMMAND\numberstanzafalse	55
COMMAND\numberstanzatrue	55
COMMAND\numlabfont	22, 51, 102
COMMAND\none@line	147
COMMAND\nonehalfspacing	445
COMMAND\nonlyXpstart	381
COMMAND\nonlysideX	255
COMMAND\npage@action	105, 116
COMMAND\npage@start	105, 435
COMMAND\npagecontents	105
COMMAND\npageparbreak	379
COMMAND\npageref	61, 282
COMMAND\npar	27, 38, 152, 153, 231
COMMAND\npar@line	147, 172
COMMAND\npara@footgroup	185
COMMAND\npara@footgroupX	229
COMMAND\npara@footsetup	185, 433
COMMAND\npara@footsetupX	226, 433, 440
COMMAND\npara@vfootnoteX	227
COMMAND\nparafootfmt	188, 189, 448

COMMAND\parafootfmtX	228
COMMAND\parafootftm	190
COMMAND\parafootftmX	230
COMMAND\parafootftmsep	380
COMMAND\parafootsep	381, 438, 443
COMMAND\parafootsepX	48, 107, 380, 381, 448
COMMAND\parafootstart	185
COMMAND\parafootstartX	227
COMMAND\paravfootnote	186, 189
COMMAND\parfillskip	188
COMMAND\parindent	445
COMMAND\parindentX	46, 383, 445, 447
COMMAND\parse@annot	163
COMMAND\parshape	75
COMMAND\parskip	19, 54, 152
COMMAND\pause@numbering	94
COMMAND\pausenumbering	20, 94, 108, 110, 153, 438, 440, 447
COMMAND\pausenumbering@page@num	94
COMMAND\penalty	188
COMMAND\pend	2, 7, 17–21, 23, 72, 129, 131, 134, 140, 147, 148, 150–153, 168, 169, 377, 438, 439, 448–450
COMMAND\preXnotes	442, 448
COMMAND\prenotesX	49, 235, 442
COMMAND\prepare@Xgroupbyline	210
COMMAND\prepare@Xprenotes	234
COMMAND\prev@nopb	370
COMMAND\prev@pb	370
COMMAND\prevlineX	107
COMMAND\prevpageX@num	107
COMMAND\print@Xfootnoterule	442
COMMAND\print@Xnotes	271, 273
COMMAND\print@Xnotes@forpages	441
COMMAND\print@eledsection	156
COMMAND\print@footnoteXrule	442
COMMAND\print@leftmargin@eledsection	359
COMMAND\print@lemma	199
COMMAND\print@line	154
COMMAND\print@notesX@forpages	441
COMMAND\print@rightmargin@eledsection	359
COMMAND\printendlines	242, 248, 289, 383, 433, 435
COMMAND\printlinefootnote	200, 202, 441
COMMAND\printlinefootnotearea	202, 203, 441
COMMAND\printlinefootnotenumbers	200
COMMAND\printlines	106, 182, 198, 203, 204, 242, 289, 383, 433, 435, 441, 446
COMMAND\printnpnum	378, 383
COMMAND\printpstart	199
COMMAND\protect	132, 377
COMMAND\providecommand	211, 433
COMMAND\pstart	2, 7, 17–21, 23, 71, 72, 115, 129, 134, 140, 147, 148, 150–153, 156, 169, 363, 377, 435–437, 439, 440, 442–444, 447–451

COMMAND\pstartinfootnote	381
COMMAND\pstartinfootnoteeverytime	381
COMMAND\pstartnum	168
COMMAND\pstartref	59, 276, 284, 438, 450
COMMAND\pstarts	436
COMMAND\raggedX	48
COMMAND\raggedleft	47
COMMAND\raggedright	47
COMMAND\raw@text	147, 148
COMMAND\rbracket	44
COMMAND\read@linelist	108–110
COMMAND\ref	61, 63, 67
COMMAND\reformatted@	289
COMMAND\reformattedwithpage	289
COMMAND\relax	19, 115, 137, 159, 170, 322, 339, 377
COMMAND\renewcommand	75, 380, 383
COMMAND\reset@current@annot	163
COMMAND\reset@msd@options@iffullpage	333
COMMAND\resetprevline@	107
COMMAND\resetprevpage@	107
COMMAND\resumenumbering	20, 90, 94, 95, 108, 110, 153, 435, 439, 440, 447
COMMAND\right	70
COMMAND\rightctab	351
COMMAND\rightlinenum	22, 102, 433, 435
COMMAND\rightltab	351
COMMAND\rightnoteupfalse	64
COMMAND\rightrtab	352
COMMAND\rightsidenote	297
COMMAND\rightskip	182, 185, 187, 188
COMMAND\rightstartnum	168
COMMAND\rigidbalance	191, 193, 195, 383, 446
COMMAND\rigidbalanceX	191, 383, 446
COMMAND\robustify	39
COMMAND\roman	344, 446
COMMAND\rtab	351, 352, 356
COMMAND\rtabtext	353, 356
COMMAND\sameword	16, 29–32, 123, 140–144, 146, 441, 443, 445, 448, 451
COMMAND\sameword@inedtext	142
COMMAND\saweword	141
COMMAND\scriptsize	102
COMMAND\section	71, 435, 449
COMMAND\section@num	90
COMMAND\sectionmark	357
COMMAND\select@lemmafont	51, 175
COMMAND\series	248, 249
COMMAND\series@	248
COMMAND\seriesatbegin	36, 261, 442
COMMAND\seriesatend	36, 261, 442
COMMAND\set@Xtxtbeforenotes	173
COMMAND\set@continuousnumberingforl	95

COMMAND\set@line	137
COMMAND\set@line@action	106, 116
COMMAND\set@txtbeforenotesX	173
COMMAND\setSErefonlypageprefixmore	62, 289, 447, 450
COMMAND\setSErefonlypageprefixsingle	62, 289, 447, 450
COMMAND\setSErefprefixmore	62
COMMAND\setSErefprefixsingle	62
COMMAND\setapprefprefixmore	62, 380
COMMAND\setapprefprefixsingle	62, 380, 446
COMMAND\setcommand@series	263
COMMAND\sethangingsymbol	53, 317, 380, 445
COMMAND\sethangingsymbol	52
COMMAND\setistwofollowinglines	207
COMMAND\setl@dlprbox	298
COMMAND\setline	23, 106, 111, 115, 129, 132, 150, 444
COMMAND\setlinenum	23, 111, 115, 129, 433, 451
COMMAND\setmsdatalabel	35
COMMAND\setmsdataposition	450
COMMAND\setmsdataseries	35
COMMAND\setprintendlines	242, 244, 435
COMMAND\setprintlines	204, 207, 242, 435
COMMAND\setsidenotesep	65
COMMAND\setsidenotsep	380
COMMAND\setstanzaindent	319
COMMAND\setstanzaindents	53, 319, 377
COMMAND\setstanzapenalties	319
COMMAND\setstanzavalues	319
COMMAND\settoggle@series	262, 436, 440
COMMAND\showlemma	132, 433, 434
COMMAND\showwordrank	32, 142
COMMAND\sidenote@margin	434
COMMAND\sidenotemargin	64, 434, 439
COMMAND\sidenotesep	380
COMMAND\sidepstartnumtrue	19
COMMAND\skip	182
COMMAND\skipnumbering	23, 118, 119, 130, 435, 442, 443
COMMAND\skipnumbering@reg	442
COMMAND\small	45
COMMAND\somemacro	33
COMMAND\special	13
COMMAND\splitmaxdepth	178, 193
COMMAND\splitoff	191
COMMAND\splittopskip	177, 193, 195
COMMAND\stanza	22, 23, 54, 55, 58, 322, 380, 382, 445, 450
COMMAND\stanza@hang	321
COMMAND\stanza@line	321
COMMAND\stanzaindent	53, 319, 441
COMMAND\stanzaindent*	53
COMMAND\stanzaindentbase	318
COMMAND\stanzanumwrapper	55

COMMAND\startlock	22, 106, 129, 322
COMMAND\startstanzahook	380
COMMAND\startsub	22, 106, 128
COMMAND\stopmsd	327
COMMAND\stopmsdata	34, 325, 326
COMMAND\store@annot@to@absline	163
COMMAND\strip@pt	185
COMMAND\strutbox	193
COMMAND\sub@action	106, 116
COMMAND\sub@lock	104
COMMAND\sub@off	114, 280
COMMAND\sub@on	114, 280
COMMAND\subline@num	103, 104, 106
COMMAND\sublinenum@rep	433
COMMAND\sublinenumberstyle	23, 101, 433
COMMAND\sublinenumincrement	21
COMMAND\sublinenumr@p	101, 433, 435
COMMAND\sublinenumrep	101, 435
COMMAND\sublineref	59, 276, 283
COMMAND\subsectionmark	357
COMMAND\sw@inthisedtext	134
COMMAND\sw@list@inedtext	138, 146
COMMAND\sw@txt	143
COMMAND\swnoexpands	32, 451
COMMAND\symlinenum	381
COMMAND\symplinenum	380
COMMAND\sza@penalty	321
COMMAND>tag	441
COMMAND\text	375
COMMAND\text<language>	46
COMMAND\textbeforenotesX	173
COMMAND\textcolor	76
COMMAND\textheight	75
COMMAND\textsuperscript	57, 88, 163
COMMAND\the	433
COMMAND\thefootnoteA	35
COMMAND\thefootnoteX	437
COMMAND\thelabidx	313
COMMAND\thepage	111
COMMAND\thepstart	19, 384
COMMAND\thepstartL	436
COMMAND\thepstartR	436
COMMAND\thestanza	55
COMMAND\this@line@list@version	123
COMMAND\thisfootnote	220
COMMAND\threecol@begin@insert	194
COMMAND\threecolfootfmt	194, 448
COMMAND\threecolfootfmtX	225
COMMAND\threecolfootgroup	193
COMMAND\threecolfootgroupX	225

COMMAND\threecolfootsetup	192
COMMAND\threecolfootsetupX	224
COMMAND\threecolvfootnote	193
COMMAND\threecolvfootnote@inserted	193
COMMAND\threecolvfootnoteX	224
COMMAND\toendnotes	26, 240, 448
COMMAND\twocolfootfmt	448
COMMAND\twocolfootfmtX	222
COMMAND\twocolfootgroupX	223
COMMAND\twocolfootsetupX	222
COMMAND\twocolvfootnoteX	222
COMMAND\twolines	262, 381
COMMAND\twolines@A	262
COMMAND\twolines@B	262
COMMAND\twolines@C	262
COMMAND\twolinesbutnotmore	381
COMMAND\twolinesonlyinsamepage	381
COMMAND\txtbeforeXnotes	381
COMMAND\txtbeforenotesX	48, 451
COMMAND\txtbeforenotesonlyonceX	48, 451
COMMAND\uline	36
COMMAND\unhbox	186
COMMAND\unpenalty	188, 189
COMMAND\unskip	188
COMMAND\unvxh	188, 382
COMMAND\unvxhX	382
COMMAND\upbracefill	350
COMMAND\usingcritext	376, 379
COMMAND\usingdtext	376, 379
COMMAND\vAfootnote	177
COMMAND\vadjust	126
COMMAND\variant	28
COMMAND\vbox	150, 152, 186, 187, 191, 233
COMMAND\vfootnote	177, 182, 186, 193
COMMAND\vl@dbfnote	213, 434
COMMAND\vl@disnote	297
COMMAND\vl@dlsnote	297
COMMAND\vl@dosnote	297
COMMAND\vl@drsnote	297
COMMAND\vnumfootnoteX	435
COMMAND\vsizer	49, 75
COMMAND\vsplit	171
COMMAND\waklam	350
COMMAND\waklamec	350
COMMAND\wapunktel	350
COMMAND\wastricht	350
COMMAND\widthX	50, 383, 447
COMMAND\wrap@edcrossref	282, 440
COMMAND\wrapcontentX	46, 448
COMMAND\wrapped@bodyfootmarkX	230

COMMAND\wrapped@footfootmarkX	230
COMMAND\x...	60
COMMAND\xabslineref	283
COMMAND\xannotationref	60
COMMAND\xdef	96, 322
COMMAND\xflagref	60, 284, 383, 446
COMMAND\xleft@appenditem	96, 131
COMMAND\xlineref	60, 383, 446
COMMAND\xpageref	60
COMMAND\xparse	33
COMMAND\xpstartref	60, 438, 451
COMMAND\xr	63
COMMAND\xright@appenditem	96
COMMAND\xsublineref	60
COMMAND\xxref	60, 285, 292, 438, 441, 448
ENVIRONMENTastanza	446
ENVIRONMENTedarrayc	356
ENVIRONMENTedarrayl	356
ENVIRONMENTedarrayr	356
ENVIRONMENTedtabularc	356
ENVIRONMENTedtabularl	356
ENVIRONMENTedtabularr	356
ENVIRONMENTledgroup	81, 304, 383, 446
ENVIRONMENTledgroupsize	304
PACKAGE(r)(e)ledmac	37
PACKAGEEledmac	77, 105, 309, 378, 379, 441, 442
PACKAGEEledpar	442
PACKAGEEtoolbox	80
PACKAGEParallel	385
PACKAGEReledmac	382, 383
PACKAGEamsgen	333
PACKAGEamsmath	333, 334
PACKAGEbabel	46, 76, 344, 446
PACKAGEbiblatex	74
PACKAGEbidi	46, 47, 81, 445
PACKAGEcaption	89
PACKAGEcolor	76
PACKAGEcsquotes	450
PACKAGEedmac	1, 6, 10, 12–14, 77, 203, 211, 277, 319, 375, 385, 433
PACKAGEedstanza	1, 14, 317
PACKAGEeledmac	1, 10, 11, 14, 16, 66, 211, 306, 309, 337, 359, 372, 376, 378–380, 437, 439, 441
PACKAGEeledpar	178, 358, 385, 435, 439–441
PACKAGEetex	445
PACKAGEetoolbox	95, 140, 248, 262, 270, 297, 358, 369, 370
PACKAGEfancyhdr	274, 449
PACKAGEfloatrow	74, 383
PACKAGEfootmisc	35, 76, 81, 211, 385, 449
PACKAGEgeometry	15
PACKAGEglossaries	67, 315, 446
PACKAGEhandout	440

PACKAGEhyperlink	256
PACKAGEhyperref	60, 134, 230, 277, 312, 362, 372, 438–440, 447, 450
PACKAGEifluatex	80
PACKAGEifixetex	80
PACKAGEimakeidx	65, 74, 81, 306, 309, 310, 379, 437–439, 441
PACKAGEindextols	314
PACKAGEindextool	379
PACKAGEindextools	65, 74, 81, 90, 306, 309, 310, 314, 379, 441, 446, 449
PACKAGEinputenc	142
PACKAGEledarab	76
PACKAGEledmac	1, 10, 14, 76, 95, 309, 375, 376, 379, 382
PACKAGEledpar	76
PACKAGEMemoir	81, 310, 379, 385, 440
PACKAGEMorewrites	74
PACKAGEMusixtex	439
PACKAGEperpage	446
PACKAGEpolyglossia	44, 76, 176, 198, 446
PACKAGEragged2e	47, 80
PACKAGEreledmac	1, 2, 11, 12, 14–17, 20, 21, 24, 25, 28–30, 33, 35, 36, 39, 40, 43, 46, 47, 49, 51, 53, 55, 58, 60, 61, 63, 65–68, 72, 74–78, 97, 99, 105, 106, 109, 110, 113, 123, 124, 132, 135, 137, 140, 169, 179, 182, 187, 198, 211, 237, 249, 253, 254, 262, 270, 282, 283, 286, 292, 309, 330, 337, 358, 359, 371, 372, 379, 380, 382–384, 444, 447, 449
PACKAGEreledpar	1, 4, 6, 8, 16, 19, 36, 50, 60, 62, 63, 72–74, 76, 78, 89, 97, 108, 113, 124, 135, 137, 179, 183, 232, 233, 249, 255, 270, 271, 273, 306, 317, 445, 446, 448, 449, 451
PACKAGESuffix	80
PACKAGETabmac	1, 14, 385
PACKAGEulem	36
PACKAGEuninormalize	29
PACKAGEXargs	29, 80
PACKAGEXkeyval	78, 270
PACKAGEXparse	80
PACKAGEXI	5, 63, 292, 447
PACKAGEXref	292
PACKAGEXstring	80, 312

A

\absline@num	1
Abu Kamil Shuja' b. Aslam	13
\actionlines@list	1
\actions@list	1
\add@inserts	1
\add@inserts@next	1
\add@msd@	1
\add@msdata	1
\add@msdata@firstlineofpage	1
\add@penalties	1
\add@Xgroupbyline	1
\addtol@denvbody	1
Adelard II	13
\advancelabel@refs	1

<code>\advanceline</code>	1, 22
<code>\Aendnote</code>	25
<code>\affixline@num</code>	1
<code>\affixpstart@num</code>	1
<code>\affixside@note</code>	1
<code>\Afootnote</code>	25
<code>\afternoteX</code>	48
<code>\afterruleX</code>	49
<code>\ampersand</code>	1, 58
<code>\annot@list</code>	1
<code>\annotationref</code>	1
<code>\annotref</code>	59
<code>\append@notesX</code>	1
<code>\append@Xnotes</code>	1
<code>\applabel</code>	1, 61
<code>\appref</code>	1, 61
<code>\apprefwithpage</code>	1, 61
<code>\arrangementX</code>	1, 38
<code>\arrangementX@normal</code>	1
<code>\arrangementX@threecol</code>	1
<code>\arrangementX@twocol</code>	1
<code>\at@every@pend</code>	1
<code>\AtEndEveryPend</code>	1, 19
<code>\AtEveryPend</code>	1, 19
<code>\AtEveryPstart</code>	1, 19
<code>\AtEveryStanza</code>	1, 54
<code>\AtEveryStopStanza</code>	1, 54
<code>\AtStartEveryPstart</code>	1, 19
<code>\AtStartEveryStanza</code>	1, 54
<code>\autopar</code>	1, 18

B

<code>\ballast</code>	75
<code>\ballast@count</code>	1
Beeton, Barbara Ann Neuhaus Friend	18
<code>\beforeeledchapter</code>	1
<code>\BeforeEveryStopStanza</code>	1
<code>\beforeinsertingX</code>	47
<code>\beforeinsertion@X</code>	1
<code>\beforenotesX</code>	49
<code>\beginnumbering</code>	1, 17
<code>\Bendnote</code>	25
<code>\Bfootnote</code>	25
<code>\bhookgroupX</code>	49
<code>\bhooknoteX</code>	46
<code>\bodyfootmarkA</code>	35
<code>\boxfootnotenumbers</code>	1
Bredon, Simon	13
Breger, Herbert	13, 14, 337
Brey, Gerhard	13

Busard, Hubert L. L.	13
\bypage@false	<u>1</u>
\bypage@true	<u>1</u>
\bypstart@false	<u>1</u>
\bypstart@true	<u>1</u>

C

\c@addcolcount	<u>1</u>
\c@ballast	<u>1</u>
\c@firstlinenum	<u>1</u>
\c@firstsublinenum	<u>1</u>
\c@labidx	<u>1</u>
\c@linenumincrement	<u>1</u>
\c@sublinenumincrement	<u>1</u>
\Cendnote	25
\Cfootnote	25
\ch@ck@l@ck	<u>1</u>
\ch@cksub@l@ck	<u>1</u>
\chapter	<u>1</u>
\check@pb@in@verse	<u>1</u>
Chester, Robert of	13
Claassens, Geert H. M.	13
\colalignX	47
Copernicus, Nicolaus	13
\critext	375
\ctab	<u>1</u>
\ctabtext	<u>1</u>
\current@annot	<u>1</u>

D

Dekker, Dirk-Jan	76
\Dendnote	25
\Dfootnote	25
\disable@endnotes	<u>1</u>
\disable@familiarnotes	<u>1</u>
\disable@notes	<u>1</u>
\disable@sidenotes	<u>1</u>
\disablel@dtabfeet	<u>1</u>
\do@actions	<u>1</u>
\do@actions@fixedcode	<u>1</u>
\do@actions@next	<u>1</u>
\do@ballast	<u>1</u>
\do@feet@custom@order	<u>1</u>
\do@feetX	<u>1</u>
\do@insidelinehook	<u>1</u>
\do@line	<u>1</u>
\do@linehook	<u>1</u>
\do@lockoff	<u>1</u>
\do@lockoffL	<u>1</u>
\do@lockon	<u>1</u>

<code>\do@lockonL</code>	1
<code>\do@Xfeet</code>	1
<code>\doedindexlabel</code>	1
<code>\doendnotes</code>	1, 26
<code>\doendnotesbysection</code>	1, 26
<code>\doinsidelinehook</code>	1, 24
<code>\dolinehook</code>	1, 24
<code>\dosplits</code>	1
Downes, Michael	75, 186, 188
<code>\doxtrafeet</code>	1
<code>\dummy@edindex</code>	1
<code>\dummy@edtext</code>	1
<code>\dummy@edtext@showlemma</code>	1
<code>\dummy@ref</code>	1

E

<code>\edaftertab</code>	1, 70, 350
<code>edarrayc</code> (environment)	68
<code>edarrayl</code> (environment)	68
<code>edarrayr</code> (environment)	68
<code>\edatleft</code>	1, 70
<code>\edatright</code>	1, 70
<code>\edbeforetab</code>	1, 70, 350
<code>\edfilldimen</code>	1
<code>\edfont@info</code>	1
<code>\edindex</code>	1, 65
<code>\edindexlab</code>	1, 67
<code>\EDLABEL</code>	1
<code>\edlabel</code>	1, 59
<code>\edlabelE</code>	1, 61
<code>\edlabelS</code>	1, 61
<code>\edlabelSE</code>	1, 61
<code>\edlineref</code>	1, 59
<code>\edmakelabel</code>	1, 61
<code>\edpageref</code>	1, 59
<code>\edrowfill</code>	1, 69
<code>\EDTAB</code>	1
<code>\edtabcolsep</code>	1, 68
<code>\EDTABINDENT</code>	1
<code>\edtabindent</code>	1
<code>\EDTABtext</code>	1
<code>edtabularc</code> (environment)	68
<code>edtabularl</code> (environment)	68
<code>edtabularr</code> (environment)	68
<code>\EDTEXT</code>	1
<code>\edtext</code>	1, 24
<code>\edvertdots</code>	1, 70
<code>\edvertline</code>	1, 70
<code>\Eendnote</code>	25
<code>\Efootnote</code>	25

<code>\eled@chapter</code>	1
<code>\eled@section</code>	1
<code>\eled@sectioning@out</code>	1
<code>\eled@subsection</code>	1
<code>\eled@subsubsection</code>	1
<code>\eledchapter</code>	1
<code>\eledchapter*</code>	1
<code>\eledsection</code>	1
<code>\eledsection*</code>	1
<code>\eledsubsection</code>	1
<code>\eledsubsection*</code>	1
<code>\eledsubsubsection</code>	1
<code>\eledsubsubsection*</code>	1
<code>\enablel@dtabfeet</code>	1
<code>\end@lemmas</code>	1
<code>\endashchar</code>	1
<code>\endline@num</code>	1
<code>\endlock</code>	1, 22
<code>\endminipage</code>	1
<code>\endnumbering</code>	1, 17
<code>\endpage@num</code>	1
<code>\endprint</code>	1
<code>\endquotation</code>	1
<code>\endquote</code>	1
<code>\endsub</code>	1, 22
<code>\endsubline@num</code>	1
environments:	
<code>edarrayc</code>	68
<code>edarrayl</code>	68
<code>edarrayr</code>	68
<code>edtabularc</code>	68
<code>edtabularl</code>	68
<code>edtabularr</code>	68
<code>ledgroup</code>	58
<code>ledgroupsize</code>	58
<code>minipage</code>	58
Euclid	13
<code>\extensionchars</code>	1, 73

F

<code>\f@x@l@cks</code>	1
Fairbairns, Robin	35
<code>\first@linenum@out@false</code>	1
<code>\first@linenum@out@true</code>	1
<code>\firstlinenum</code>	1, 21
<code>\firstseriesX@</code>	1
<code>\firstsublinenum</code>	1, 21
<code>\firstXseries@</code>	1
<code>\fix@page</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>\flagstanz</code>	1, 58
<code>\flush@notes</code>	1
<code>\fnpos</code>	1, 37
Folkerts, Menso	13
<code>\footfootmarkA</code>	35
<code>\footfudgefiddle</code>	1, 75
<code>\footnote</code>	1
<code>\footnoteA</code>	35
<code>\footnoteB</code>	35
<code>\footnoteC</code>	35
<code>\footnoteD</code>	35
<code>\footnoteE</code>	35
<code>\footnotelang@lua</code>	1
<code>\footnotelang@poly</code>	1
<code>\footnoteoptions@</code>	1
<code>\footnoteXmark</code>	36
<code>\footnoteXtext</code>	36
<code>\footplitskips</code>	1
<code>\fullstop</code>	1

G

Gädeke, Nora	13
<code>\get@edindex@hyperref</code>	1
<code>\get@edindex@ledinnote@command</code>	1
<code>\get@fnmark</code>	1
<code>\get@fnmarkX</code>	1
<code>\get@index@command</code>	1
<code>\get@linelistfile</code>	1
<code>\get@sw@txt</code>	1
<code>\get@thisfootnote</code>	1
<code>\get@thisfootnoteX</code>	1
<code>\getline@num</code>	1
<code>\gl@p</code>	1

H

<code>\h@num</code>	1
<code>\hangindentX</code>	46
<code>\hidenumbering</code>	1, 23
<code>\hidenumberingonleftpage</code>	1, 23
<code>\hidenumberingonrightpage</code>	1
<code>\Hilfsbox</code>	1
<code>\hilfsbox</code>	1
<code>\hilfscount</code>	1
<code>\HILFSskip</code>	1
<code>\Hilfsskip</code>	1

<code>\hilfsskip</code>	1
<code>\hsizethreecolX</code>	47
<code>\hsizetwocolX</code>	47
<code>\Hy@raisedlink@left</code>	1
<code>\hyperlinkformat</code>	1
<code>\hyperlinkformatR</code>	1
<code>\hyperlinkR</code>	1

I

<code>\if@addsw</code>	1
<code>\if@edtext@secondarg@</code>	1
<code>\if@eled@sectioning</code>	1
<code>\if@firstlineofpage</code>	1
<code>\if@led@nofoot</code>	1
<code>\if@ledgroup</code>	1
<code>\if@lemmacommand@</code>	1
<code>\if@msdata@insertedfrompreviouspage</code>	1
<code>\if@noeled@sec</code>	1
<code>\if@noneed@Footnote</code>	1
<code>\if@RTL</code>	1
<code>\ifafterendnumberingR</code>	1
<code>\ifat@every@pend@star@</code>	1
<code>\ifat@every@pstart@star@</code>	1
<code>\ifautopar@pause</code>	1
<code>\ifbypage@</code>	1
<code>\ifbypage@R</code>	1
<code>\ifbypstart@</code>	1
<code>\ifbypstart@R</code>	1
<code>\ifeledmacompat@</code>	1
<code>\iffirst@linenum@out@</code>	1
<code>\ifindtl@innote</code>	1
<code>\ifindtl@notenumber</code>	1
<code>\ifinserthangingsymbol</code>	1
<code>\ifinstanza</code>	1
<code>\ifl@d@dash</code>	1
<code>\ifl@d@elin</code>	1
<code>\ifl@d@esl</code>	1
<code>\ifl@d@pnum</code>	1
<code>\ifl@d@ssub</code>	1
<code>\ifl@d@Xmorethantwolines</code>	1
<code>\ifl@d@Xtwolines</code>	1
<code>\ifl@dend@X</code>	1
<code>\ifl@dhiddenumber</code>	1
<code>\ifl@dmemoir</code>	1
<code>\ifl@dpaging</code>	1
<code>\ifl@dpairing</code>	1
<code>\ifl@dprintingcolumns</code>	1
<code>\ifl@dprintingpages</code>	1
<code>\ifl@dskipnumber</code>	1
<code>\ifl@dskipversenumber</code>	1

<code>\ifl@dstartendok</code>	1
<code>\ifl@footmisc</code>	1
<code>\ifl@imakeidx</code>	1
<code>\ifl@indextools</code>	1
<code>\ifledfinal</code>	1, 73
<code>\ifledgroupnotesL@</code>	1
<code>\ifledgroupnotesR@</code>	1
<code>\iflednopbinverse</code>	1
<code>\ifledRcol</code>	1
<code>\ifledRcol@</code>	1
<code>\ifnocritical@</code>	1
<code>\ifnoend@</code>	1
<code>\ifnofamiliar@</code>	1
<code>\ifnoledgroup@</code>	1
<code>\ifnoquotation@</code>	1
<code>\ifnoteschanged@</code>	1
<code>\ifnumberedpar@</code>	1
<code>\ifnumbering</code>	1
<code>\ifnumberingR</code>	1
<code>\ifnumberline</code>	1
<code>\ifnumberstanza</code>	1
<code>\ifparapparatus@</code>	1
<code>\ifparledgroup</code>	1
<code>\ifpst@rtedL</code>	1
<code>\ifresumenummering@start</code>	1
<code>\ifseriesbefore</code>	1
<code>\ifsidepstartnum</code>	1
<code>\ifstopmsdata@inserted@</code>	1
<code>\ifsublines@</code>	1
<code>\ifwidthliketwocolumns</code>	1
<code>\ifXendinsertsep@</code>	1
<code>\ifxindy@</code>	1
<code>\ifxindyhyperref@</code>	1
<code>\initnumbering@quote</code>	1
<code>\initnumbering@reg</code>	1
<code>\insert@count</code>	0, 1
<code>\insert@msdata</code>	1
<code>\insert@txtbeforenotesX</code>	1
<code>\insert@Xtxtbeforenotes</code>	1
<code>\inserthangingymbol</code>	1
<code>\insertlines@list</code>	1
<code>\insertparafootsepX</code>	1
<code>\inserts@list</code>	1

J

Jayaditya	13
-----------------	----

K

Kabelschacht, Alois	173
---------------------------	-----

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@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@dobblearg</code>	1
<code>\l@disnote</code>	1
<code>\l@dlabel@parse</code>	1
<code>\l@dld@ta</code>	1
<code>\l@dldp@rbox</code>	1
<code>\l@dlsn@te</code>	1
<code>\l@dlsnote</code>	1
<code>\l@dmake@labels</code>	1
<code>\l@dmodforedtext</code>	1
<code>\l@dnullfills</code>	1
<code>\l@dnumpstartsL</code>	1

<code>\@doldold@footnotetext</code>	1
<code>\@dp@rsefootspec</code>	1
<code>\@dparsedendline</code>	1
<code>\@dparsedendpage</code>	1
<code>\@dparsedendsub</code>	1
<code>\@dparsedstartline</code>	1
<code>\@dparsedstartpage</code>	1
<code>\@dparsedstartsub</code>	1
<code>\@dparsefootspec</code>	1
<code>\@dpush@begins</code>	1
<code>\@drd@ta</code>	1
<code>\@dref@undefined</code>	1
<code>\@drestorefills</code>	1
<code>\@drestoreforedtext</code>	1
<code>\@drp@rbox</code>	1
<code>\@drsn@te</code>	1
<code>\@drsnote</code>	1
<code>\@dsetmaxcolwidth</code>	1
<code>\@dskipnumberfalse</code>	1
<code>\@dskipnumbertrue</code>	1
<code>\@dtabaddcols</code>	1
<code>\@dtabnoexpands</code>	1
<code>\@dunboxmpfoot</code>	1
<code>\@dunhbox@line</code>	1
<code>\@dzeropenalties</code>	1
<code>\@wrap@ifnotempty</code>	1
<code>\@wrapcs@ifnotempty</code>	1
<code>\label</code>	61
<code>\labelpstartfalse</code>	1
<code>\labelpstarttrue</code>	1, 19
<code>\labelref@list</code>	1
<code>\labelrefsparseline</code>	1
<code>\labelrefsparsesubline</code>	1
<code>\last@page@num</code>	1
Lavagnino, John	12
<code>\led@check@nopb</code>	1
<code>\led@check@pb</code>	1
<code>\led@err@AutoparNotNumbered</code>	1
<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@PendNoPstart</code>	1
<code>\led@err@PendNotNumbered</code>	1
<code>\led@err@PstartInPstart</code>	1
<code>\led@err@PstartNotNumbered</code>	1
<code>\led@err@ReverseColumns</code>	1
<code>\led@err@toendnotes@outsidenumbering</code>	1
<code>\led@err@TooManyColumns</code>	1
<code>\led@err@UnequalColumns</code>	1
<code>\led@error@fail@patch@docclearpage</code>	1
<code>\led@error@fail@patch@iiiminipage</code>	1
<code>\led@error@fail@patch@makeecol</code>	1
<code>\led@error@fail@patch@reinserts</code>	1
<code>\led@error@fail@patch@endminipage</code>	1
<code>\led@error@PackageAfterEledmac</code>	1
<code>\led@mess@NotesChanged</code>	1
<code>\led@mess@SectionContinued</code>	1
<code>\led@nopb</code>	1
<code>\led@nopbnum</code>	1
<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@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

ledgroup (environment)	58
ledgroupsize (environment)	58
\ledinnernote	1, 64
\ledinnote	1
\ledinnotehyperpage	1
\ledinnotemark	1
\ledleftnote	1, 64
\ledlinenum	1
\ledllfill	1
\ledlsnotefontsetup	1, 64
\ledlsnotesep	1, 64
\ledlsnotewidth	1, 64
\lednopb	1, 72
\lednopbinversetrue	73
\lednopbnum	1
\ledouternote	1, 64
\ledpb	1, 72
\ledpbnum	1
\ledpbsetting	1, 73
\ledrightnote	1, 64
\ledrlfill	1
\ledrsnotefontsetup	1, 64
\ledrsnotesep	1, 64
\ledrsnotewidth	1, 64
\ledsectnomark	1
\ledsectnotoc	1
\ledsetnormalparstuff@common	1
\ledsetnormalparstuffX	1
\ledsidenote	1, 64
\leftctab	1
\leftlinenum	1, 22
\leftltab	1
\leftnoteupfalse	64
\leftpstartnum	1
\leftrtab	1
Leibniz	13
\lemma	1, 27
\letsforverteilen	1
\line@list	1
\line@list@stuff	1
\line@list@version	1
\line@margin	1
\line@num	1
\line@set	1
\lineannot	1, 56
\lineannot@set	1
\lineation	1, 21
\linenum	1, 27
\linenum@out	1
\linenumannotation	1, 55

<code>\linenumannotation@side</code>	1
<code>\linenumberlist</code>	1, 21
<code>\linenumberstyle</code>	1, 23
<code>\linenumincrement</code>	1, 21
<code>\linenummargin</code>	1, 21
<code>\linenumr@p</code>	1
<code>\linenumrep</code>	1
<code>\linenumsep</code>	1, 22
<code>\linerangesep@</code>	1
<code>\list@clear</code>	1
<code>\list@clearing@reg</code>	1
<code>\list@create</code>	1
<code>\lock@disp</code>	1
<code>\lock@off</code>	1
<code>\lock@on</code>	1
<code>\lockdisp</code>	1, 22
Lorch, Richard	13
<code>\ltab</code>	1
<code>\ltabtext</code>	1
Luecking, Dan	79

M

<code>\m@mmf@check</code>	1
<code>\m@mmf@prepare</code>	1
<code>\M@sect</code>	1
<code>\makehboxofhboxes</code>	1
<code>\managestanza@modulo</code>	1
<code>\maxhnotesX</code>	49
Mayer, Gyula	13
<code>\measurebody</code>	1
<code>\measurecell</code>	1
<code>\measuremrow</code>	1
<code>\measuretbody</code>	1
<code>\measuretcell</code>	1
<code>\measuretrow</code>	1
Middleton, Thomas	13, 104
minipage (environment)	58
Mittelbach, Frank	12, 13
<code>\morenoexpands</code>	1, 75
<code>\mp@append@notesX</code>	1
<code>\mp@append@Xnotes</code>	1
<code>\mpfnpos</code>	1, 37
<code>\mpnormalfootgroup</code>	1
<code>\mpnormalfootgroupX</code>	1
<code>\mpnormalvfootnote</code>	1
<code>\mpnormalvfootnote@inserted</code>	1
<code>\mpnormalvfootnoteX</code>	1
<code>\mppara@footgroupX</code>	1
<code>\mppara@vfootnoteX</code>	1
<code>\mpparafootgroup</code>	1

<code>\mpparavfootnote</code>	1
<code>\mpthreecolfootgroup</code>	1
<code>\mpthreecolfootgroupX</code>	1
<code>\mpthreecolfootsetup</code>	1
<code>\mpthreecolfootsetupX</code>	1
<code>\mptwocolfootgroup</code>	1
<code>\mptwocolfootgroupX</code>	1
<code>\mptwocolfootsetup</code>	1
<code>\mptwocolfootsetupX</code>	1
<code>\msdata</code>	1, 34
<code>\msdata@c</code>	1
<code>\msdata@cR</code>	1
<code>\multfootsep</code>	1, 35
<code>\multiplefootnotemarker</code>	1

N

<code>\n@num</code>	1
<code>\n@num@stanza</code>	1
<code>\new@line</code>	1
<code>\newhookarg@specific</code>	1
<code>\newhookcommand@series</code>	1
<code>\newhookcommand@series@reload</code>	1
<code>\newhooktoggle@series</code>	1
<code>\newhooktoggle@series@reload</code>	1
<code>\newhooktoggle@specific</code>	1
<code>\newseries@</code>	1
<code>\newverse</code>	1
<code>\NEXT</code>	1
<code>\next@line@list@stuff</code>	1
<code>\no@expands</code>	1
<code>\noeledsec</code>	72
<code>\nomk@</code>	1
<code>\normal@footnotemarkX</code>	1
<code>\normal@page@break</code>	1
<code>\normal@pars</code>	1
<code>\normalbfnoteX</code>	1
<code>\normalbodyfootmarkX</code>	1
<code>\normalfootfmt</code>	1
<code>\normalfootfmtX</code>	1
<code>\normalfootfootmarkX</code>	1
<code>\normalfootgroup</code>	1
<code>\normalfootgroupX</code>	1
<code>\normalfootnoterule</code>	1
<code>\normalfootnoteruleX</code>	1
<code>\normalfootstart</code>	1
<code>\normalfootstartX</code>	1
<code>\normalvfootnote</code>	1
<code>\normalvfootnote@inserted</code>	1
<code>\normalvfootnoteX</code>	1
<code>\notefontsizeX</code>	45

<code>\notenumfontX</code>	45
<code>\noteschanged@false</code>	<u>1</u>
<code>\noteschanged@true</code>	<u>1</u>
<code>\noteswidthliketwocolumnsX</code>	50
<code>\nulledindex</code>	<u>1</u>
<code>\nullsetzen</code>	<u>1</u>
<code>\num@lines</code>	<u>1</u>
<code>\numberedpar@false</code>	<u>1</u>
<code>\numberedpar@true</code>	<u>1</u>
<code>\numberingfalse</code>	<u>1</u>
<code>\numberingtrue</code>	<u>1</u>
<code>\numberlinefalse</code>	20
<code>\numberlinetrue</code>	20
<code>\numberpstartfalse</code>	<u>1</u> , 19
<code>\numberpstarttrue</code>	<u>1</u> , 19
<code>\numberstanzafalse</code>	55
<code>\numberstanzatrue</code>	55
<code>\numlabfont</code>	<u>1</u> , 51

O

<code>\old@hsize</code>	<u>1</u>
<code>\one@line</code>	<u>1</u>
<code>optionafter</code>	57
<code>optionauxdir</code>	15, 448
<code>optionbefore</code>	57
<code>optioncontinuousnumberingwithcolumns</code>	90, 155, 447
<code>optioninnote</code>	446
<code>optioninnote</code>	446
<code>optionlinangesep</code>	270
<code>optionnocritical</code>	446
<code>optionnoeledsec</code>	363, 448
<code>optionnoend</code>	446
<code>optionnopenalties</code>	75
<code>optionnotenumber</code>	446
<code>optionswcaseinsensitive</code>	29, 451

P

<code>\page@action</code>	<u>1</u>
<code>\page@num</code>	<u>1</u>
<code>\pagelinesep</code>	<u>1</u> , 66
<code>\pageref</code>	61
<code>\par@line</code>	<u>1</u>
<code>\para@footgroupX</code>	<u>1</u>
<code>\para@footsetup</code>	<u>1</u>
<code>\para@footsetupX</code>	<u>1</u>
<code>\para@vfootnoteX</code>	<u>1</u>
<code>\parafootfmt</code>	<u>1</u>
<code>\parafootfmtX</code>	<u>1</u>
<code>\parafootgroup</code>	<u>1</u>
<code>\parafootsepX</code>	48

<code>\parafootstart</code>	1
<code>\parafootstartX</code>	1
<code>\paravfootnote</code>	1
<code>\parindentX</code>	46
<code>\parse@annot</code>	1
<code>\pausenumbering</code>	1, 20
<code>\pausenumbering@page@num</code>	1
<code>\pend</code>	1, 17
Plato of Tivoli	13
<code>\postbodyfootmark</code>	1
<code>\prebodyfootmark</code>	1
<code>\prenotesX</code>	49
<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, 17
<code>\pstarteref</code>	1
<code>\pstartnum</code>	1
<code>\pstartref</code>	59
Q	
<code>\quotation</code>	1
<code>\quote</code>	1

R

<code>\raggedX</code>	48
<code>\raw@text</code>	1
<code>\rbracket</code>	1
<code>\read@linelist</code>	1
<code>\ref</code>	61
<code>\ref@reg@later</code>	1
<code>\Relax</code>	1
<code>\reledmac@error</code>	1
<code>\reledmac@warning</code>	1
<code>\removehboxes</code>	1
<code>\reset@current@annot</code>	1
<code>\reset@msd@options@iffullpage</code>	1
<code>\resetprevline@</code>	1, 107
<code>\resetprevpage@</code>	1
<code>\resetprevpage@num</code>	107
<code>\restore@endnotes</code>	1
<code>\restore@familiarnotes</code>	1
<code>\restore@notes</code>	1
<code>\restore@sidenotes</code>	1
<code>\resumenumbering</code>	1, 20
<code>\rightctab</code>	1
<code>\rightlinenum</code>	1, 22
<code>\rightltab</code>	1
<code>\rightnoteupfalse</code>	64
<code>\rightrtab</code>	1
<code>\rightstartnum</code>	1
<code>\rigidbalance</code>	1
<code>\rigidbalanceX</code>	1
<code>\rtab</code>	1
<code>\rtabtext</code>	1

S

Sacrobosco	13
<code>\sameword</code>	1, 28
<code>\sameword@inedtext</code>	1
Schöpf, Rainer	13
<code>\section@num</code>	1
<code>\select@@lemmafont</code>	1
<code>\select@lemmafont</code>	1, 51
<code>\SEref</code>	1, 61
<code>\SErefonlypage</code>	61
<code>\SErefwithpage</code>	1, 61
<code>\series</code>	1
<code>\seriesatbegin</code>	1, 36
<code>\seriesatend</code>	1, 36
<code>\set@continuousnumberingforL</code>	1
<code>\set@line</code>	1
<code>\set@line@action</code>	1
<code>\set@txtbeforenotesX</code>	1

<code>\set@Xtxtbeforenotes</code>	1
<code>\setapprefprefixmore</code>	62
<code>\setapprefprefixsingle</code>	62
<code>\setcommand@series</code>	1
<code>\sethangingsymbol</code>	1, 53
<code>\setistwofollowinglines</code>	1
<code>\setl@dlp@rbox</code>	1
<code>\setl@drpr@rbox</code>	1
<code>\setline</code>	1, 22
<code>\setlinenum</code>	1, 23
<code>\setmcellcenter</code>	1
<code>\setmcellleft</code>	1
<code>\setmcellright</code>	1
<code>\setmrowcenter</code>	1
<code>\setmrowleft</code>	1
<code>\setmrowright</code>	1
<code>\setmsdatalabel</code>	1, 35
<code>\setmsdataposition</code>	1, 35
<code>\setmsdataseries</code>	1, 35
<code>\setnoteswidthliketwocolumnsX@</code>	1
<code>\setnotesXpositionliketwocolumns@</code>	1
<code>\setprintendlines</code>	1
<code>\setprintlines</code>	1
<code>\setSErefonlypageprefixmore</code>	62
<code>\setSErefonlypageprefixsingle</code>	62
<code>\setSErefprefixmore</code>	62
<code>\setSErefprefixsingle</code>	62
<code>\setsidenotesep</code>	65
<code>\setstanzaindents</code>	1, 51
<code>\setstanzapenalties</code>	1, 53
<code>\setstanzavalues</code>	1
<code>\settcellcenter</code>	1
<code>\settcellleft</code>	1
<code>\settcellright</code>	1
<code>\settoggle@series</code>	1
<code>\setthrowcenter</code>	1
<code>\setthrowleft</code>	1
<code>\setthrowright</code>	1
<code>\setXnotespositionliketwocolumns@</code>	1
<code>\setXnoteswidthliketwocolumns@</code>	1
<code>\showlemma</code>	1, 73
<code>\showwordrank</code>	1, 32
<code>\sidenote@margin</code>	1
<code>\sidenotemargin</code>	1, 64
<code>\sidepstartnumtrue</code>	19
<code>\skip@lockoff</code>	1
<code>\skipnumbering</code>	1, 23
<code>\splitoff</code>	1
<code>\spreadmath</code>	1, 69
<code>\spreadtext</code>	1, 69

<code>\stanza</code>	1, 51
<code>\stanza@count</code>	1
<code>\stanza@hang</code>	1
<code>\stanza@line</code>	1
<code>\stanzaindent</code>	1, 53
<code>\stanzaindent*</code>	1, 53
<code>\stanzaindentbase</code>	1, 51
<code>\stanzanumwrapper</code>	1, 55
<code>\startlock</code>	1, 22
<code>\startsub</code>	1, 22
<code>\stepl@dcolcount</code>	1
<code>\stopmsdata</code>	1, 34
<code>\store@cannot@to@absline</code>	1
<code>\strip@szacnt</code>	1
<code>\sub@action</code>	1
<code>\sub@lock</code>	1
<code>\sub@off</code>	1
<code>\sub@on</code>	1
<code>\subline@num</code>	1
<code>\sublinenumberstyle</code>	1, 23
<code>\sublinenumincrement</code>	1, 21
<code>\sublinenumr@p</code>	1
<code>\sublinenumrep</code>	1
<code>\sublineref</code>	1, 59
<code>\sublines@false</code>	1
<code>\sublines@true</code>	1
<code>\sublock@disp</code>	1
<code>\sublockdisp</code>	1
Sullivan, Wayne	13, 14, 75, 89, 94, 186, 187, 277, 317
<code>\sw@noexpand</code>	1
<code>\swnoexpands</code>	32
<code>\sza@penalty</code>	1

T

<code>\tabHilfbox</code>	1
<code>\tabhilfbox</code>	1
<code>\theadcolcount</code>	1
<code>\theadtext</code>	1
<code>\theendpageline</code>	1
<code>\tfootnoteA</code>	35
Theodosius	13
<code>\thepageline</code>	1
<code>\thepstart</code>	1, 19
<code>\thestanza</code>	1, 55
<code>\thestartpageline</code>	1
<code>\this@line@list@version</code>	1
<code>\threecol@begin@insert</code>	1
<code>\threecolfootfmt</code>	1
<code>\threecolfootfmtX</code>	1
<code>\threecolfootgroup</code>	1

<code>\threecolfootgroupX</code>	1
<code>\threecolfootsetup</code>	1
<code>\threecolfootsetupX</code>	1
<code>\threecolvfootnote</code>	1
<code>\threecolvfootnote@inserted</code>	1
<code>\threecolvfootnoteX</code>	1
<code>\toendnotes</code>	1, 26
<code>\toendnotes*</code>	1
<code>\twocolfootfmt</code>	1
<code>\twocolfootfmtX</code>	1
<code>\twocolfootgroup</code>	1
<code>\twocolfootgroupX</code>	1
<code>\twocolfootsetup</code>	1
<code>\twocolfootsetupX</code>	1
<code>\twocolvfootnote</code>	1
<code>\twocolvfootnote@inserted</code>	1
<code>\twocolvfootnoteX</code>	1
<code>\txtbeforenotesonlyonceX</code>	48
<code>\txtbeforenotesX</code>	48

U

<code>\unvxhX</code>	1
----------------------------	---

V

Vamana	13
<code>\variab</code>	1
<code>\vbfnoteX</code>	1
<code>\vl@dbfnote</code>	1
<code>\vl@dcsnote</code>	1
<code>\vl@disnote</code>	1
<code>\vl@dlsnote</code>	1
<code>\vl@dosnote</code>	1
<code>\vl@drsnote</code>	1
<code>\vnumfootnoteX</code>	1

W

Whitney, Ron	13
<code>\widthX</code>	50
<code>\wrap@edcrossref</code>	1
<code>\wrapcontentX</code>	46
<code>\wrapped@bodyfootmarkX</code>	1
<code>\wrapped@footfootmarkX</code>	1
Wujastyk, Dominik	12

X

<code>\X@atbegininsertion</code>	1
<code>\X@beforeinsertion</code>	1
<code>\X@doreinfeet</code>	1
<code>\xabslineref</code>	1
<code>\Xafterlemmaseparator</code>	44

<code>\Xafternote</code>	48
<code>\Xafternumber</code>	42
<code>\Xafterrule</code>	49
<code>\Xaftersymlinenum</code>	42
<code>\xannotationref</code>	1
<code>\xannotref</code>	60
<code>\Xarrangement</code>	1, 38
<code>\Xarrangement@normal</code>	1
<code>\Xarrangement@paragraph</code>	1
<code>\Xarrangement@threecol</code>	1
<code>\Xarrangement@twocol</code>	1
<code>\Xbeforeinserting</code>	47
<code>\Xbeforelemmaseparator</code>	44
<code>\Xbeforenotes</code>	49
<code>\Xbeforenumber</code>	40, 42
<code>\Xbeforesymlinenum</code>	42
<code>\Xbhookgroup</code>	49
<code>\Xbhooknote</code>	46
<code>\Xboxlinenum</code>	43
<code>\Xboxlinenumalign</code>	43
<code>\Xboxsymlinenum</code>	43
<code>\Xcolalign</code>	47
<code>\xedindex</code>	1
<code>\xedlabel</code>	1
<code>\xedtext</code>	1
<code>\Xendafterlemmaseparator</code>	44
<code>\Xendafternote</code>	50
<code>\Xendafterpagenumber</code>	40
<code>\Xendaftersymlinenum</code>	42
<code>\Xendaftrenumber</code>	42
<code>\Xendahookinplaceofnumber</code>	44
<code>\Xendahooklinenumber</code>	44
<code>\Xendbeforelemmaseparator</code>	44
<code>\Xendbeforenumber</code>	42
<code>\Xendbeforepagenumber</code>	40
<code>\Xendbeforesymlinenum</code>	42
<code>\Xendbhookinplaceofnumber</code>	44
<code>\Xendbhooklinenumber</code>	44
<code>\Xendbhooknote</code>	46
<code>\Xendboxendlinenumalign</code>	43
<code>\Xendboxlinenum</code>	43
<code>\Xendboxlinenumalign</code>	43
<code>\Xendboxstartlinenumalign</code>	43
<code>\Xendboxsymlinenum</code>	43
<code>\Xendhangindent</code>	46
<code>\Xendinplaceoflemmaseparator</code>	45
<code>\Xendinplaceofnumber</code>	42
<code>\Xendinplaceofpagenumber</code>	40
<code>\Xendlemmadisablefontselection</code>	45
<code>\Xendlemmafont</code>	45

<code>\Xendlemmaseparator</code>	44
<code>\Xendlinenumannotationposition</code>	57
<code>\Xendlineprefixmore</code>	40
<code>\Xendlineprefixsingle</code>	40
<code>\Xendlinerangeseparator</code>	40
<code>\Xendmorethantwolines</code>	41
<code>\Xendnonumber</code>	41
<code>\Xendnotefontsize</code>	45
<code>\Xendnotenumfont</code>	45
<code>\Xendnumberonlyfirstinline</code>	39
<code>\Xendnumberonlyfirstintwolines</code>	39
<code>\Xendpagenumberonlyfirst</code>	39
<code>\Xendparagraph</code>	50
<code>\Xendsep</code>	50
<code>\Xendsublinesep</code>	42
<code>\Xendsymlinenum</code>	39
<code>\Xendsympagenum</code>	40
<code>\Xendtwolines</code>	41
<code>\Xendtwolinesbutnotmore</code>	41
<code>\Xendwrapcontent</code>	46
<code>\xflagref</code>	<u>1</u>
<code>\Xgroupbyline</code>	48
<code>\Xgroupbylineseparetwolines</code>	48
<code>\Xhangindent</code>	46
<code>\Xhsizethreecol</code>	47
<code>\Xhsizetwocol</code>	47
<code>\Xinplaceoflemmaseparator</code>	44
<code>\Xinplaceofnumber</code>	42
<code>\Xinsertparafootsep</code>	<u>1</u>
<code>\Xledsetnormalparstuff</code>	<u>1</u>
<code>\xleft@appenditem</code>	<u>1</u>
<code>\Xlemmadisablefontselection</code>	45
<code>\Xlemmafont</code>	45
<code>\Xlemmaseparator</code>	44
<code>\Xlinenumannotationposition</code>	57
<code>\Xlinerangeseparator</code>	40
<code>\xlineref</code>	<u>1</u> , 60
<code>\Xmaxhnotes</code>	49
<code>\Xmorethantwolines</code>	40
<code>\Xnolemmaseparator</code>	<u>1</u> , 44
<code>\Xnonbreakableafternumber</code>	42
<code>\Xnonumber</code>	41
<code>\Xnotefontsize</code>	45
<code>\Xnotenumfont</code>	45
<code>\Xnoteswidthliketwocolumns</code>	50
<code>\Xnumberonlyfirstinline</code>	39
<code>\Xnumberonlyfirstintwolines</code>	39
<code>\Xonlypstart</code>	41
<code>\Xpagelinesep</code>	42
<code>\xpageref</code>	<u>1</u> , 60

<code>\Xparafootsep</code>	48
<code>\Xparindent</code>	46
<code>\Xprenotes</code>	<u>1</u> , 49
<code>\Xprenotes@</code>	<u>1</u>
<code>\Xpstart</code>	41
<code>\Xpstarteverytime</code>	41
<code>\xpstartref</code>	<u>1</u> , 60
<code>\XR@test</code>	<u>1</u>
<code>\XR@test@mac</code>	<u>1</u>
<code>\XR@test@mac@test</code>	<u>1</u>
<code>\Xragged</code>	48
<code>\xright@appenditem</code>	<u>1</u>
<code>\Xrigidbalance</code>	<u>1</u>
<code>\Xstanza</code>	41
<code>\Xstanzaseparator</code>	41
<code>\Xstorelineinfo</code>	<u>1</u>
<code>\xsublineref</code>	<u>1</u> , 60
<code>\Xsublinesep</code>	22, 42
<code>\Xsublinesepside</code>	22
<code>\Xsymlinenum</code>	39
<code>\Xtoendnotes</code>	26
<code>\Xtwolines</code>	40
<code>\Xtwolinesonlyinsamepage</code>	41
<code>\Xtxtbeforenotes</code>	48
<code>\Xtxtbeforenotesonlyonce</code>	48
<code>\Xunvxh</code>	<u>1</u>
<code>\Xwidth</code>	50
<code>\Xwrapcontent</code>	46
<code>\Xwrapendlemma</code>	46
<code>\Xwraplemma</code>	46
<code>\Xwraplinenumannotation@ref</code>	<u>1</u>
<code>\Xwraplinenumannotation@side</code>	<u>1</u>
<code>\xxref</code>	<u>1</u> , 60

Z

<code>\zz@@@</code>	<u>1</u>
<code>\zz@@@</code>	<u>1</u>

Change History

v0.1.0.		
General: First public release	1
v0.2.0.		
\ifl@dmemoir: Added \ifl@dmemoir for memoir class having been used	81
\morenoexpands: Added \l@dtabnoexpands to \no@expands	133
\reledmac@error: Added \eledmac@error and replaced error messages	82
General: Added tabmac code, and extended indexing	1
v0.2.1.		
\@lab: Removed page setting from \@lab	280
\doxtrafeet: Renamed \doxtrafeet to \l@ddoxtrafeet	270
\edlabel: Tweaked \edlabel to get correct page numbers	277
\l@ddodoreinextrafeet: Renamed \dodoreinextrafeet to \l@ddodoreinextrafeet	272
\morenoexpands: Removed some \lets from \no@expands. These were in edmac but Peter Wilson feels that they should not have been as they disabled page/line refs in a footnotes	133
General: Added text about normal labeling	61
Bug fixes and match with mempatch v1.8	1
Major changes to insert code when memoir is loaded	274
v0.2.2.		
\footfudgefiddle: Added \footfudgefiddle	185
\next@line@list@stuff: Added initial write of page number in \line@list@stuff	124
\para@footsetup: Added \footfudgefiddle to \para@footsetup	185
\para@footsetupX: Added \footfudgefiddle to \para@footsetupX	226
General: Improved paragraph footnotes	1
New Dekker example	1
Used \providecommand for \@gobblethree to avoid clash with the amsfonts package	88
v0.3.0.		
\@lab: Replaced \the\line@num by \linenumr@p\line@num in \@lab, and similar for sub-lines	280
\@nl@reg: Added a bunch of code to \@nl for handling \setlinenum	111
\ledlinenum: Added \linenumr@p and \sublinenum@rep to \leftlinenum and \rightlinenum	102
\linenumberlist: Added \linenumberlist mechanism	89
\printendlines: Added \linenumr@p and \sublinenumr@p to \printendlines	244
\printlines: Added \linenumr@p and \sublinenumr@p to \printlines	209
\sublinenumr@p: Added \linenumberstyle and \sublinenumberstyle	101
General: Includes edstanza and more	1
v0.3.1.		
General: Not released. Added remarks about the parallel package	1
v0.4.0.		
\@iiiminipage: Modified kernel \@iiiminipage and \endminipage to cater for critical footnotes	303
\Xarrangement@normal: Added minpage footnote setup to \footnormal	179
\edtext: Added \showlemma to \edtext	134
\l@dfeetendmini: Added \l@dfeetbeginmini, \l@dfeetendmini and all their supporting code	301
\mpnormalfootgroup: Added \mpnormalfootgroup	183

<code>\mpnormalvfootnote</code> : Added <code>\mpnormalvfootnote</code>	181
<code>\showlemma</code> : Added <code>\showlemma</code>	88
General: Added final/draft options	78
Added <code>ledgroup</code> environment	304
Added <code>ledgroupsize</code> environment	305
Added <code>minipage</code> , etc., support	1
v0.4.1.	
<code>\doXfeet</code> : Changed <code>\doXfeet</code> code for easier extensions	271
<code>\edindex</code> : Let <code>eledmac</code> take advantage of memoir's indexing	310
<code>\printXnotes</code> : Added <code>\opXfeet</code>	271
General: Added code for changing <code>\doclearpage</code>	275
Not released. Minor editorial improvements and code tweaks	1
Only change <code>\@footnotetext</code> and <code>\@footnotemark</code> if memoir not used	212
v0.5.0.	
<code>\@footnotetext</code> : Enabled regular <code>\footnote</code> in numbered text	213
<code>\xympar</code> : Eliminated <code>\marginpar</code> disturbance	293
General: Added left and right side notes	293
Added sidenotes, familiar footnotes in numbered text	1
v0.5.1.	
<code>\affixline@num</code> : Changed <code>\affixline@num</code> to cater for sidenotes	164
<code>\l@getsidenote@margin</code> : Added <code>\sidenotemargin</code> and <code>\sidenote@margin</code> ...	293
General: Added moveable side note	293
Fixed right line numbers killed in v0.5	1
Only change <code>\hsize</code> in <code>ledgroupsize</code> environment otherwise page number can be in wrong place	305
v0.6.0.	
<code>\@lopR</code> : Added <code>\@pend</code> , <code>\@pendR</code> , <code>\@lopL</code> and <code>\@lopR</code> in anticipation of parallel processing	113
<code>\@nl@reg</code> : Added <code>\fix@page</code> to <code>\@nl</code>	111
Extended <code>\@nl</code> to include the page number	111
<code>\fix@page</code> : Added <code>\last@page@num</code> and <code>\fix@page</code>	112
<code>\get@thisfootnote</code> : Changed <code>\l@dbfnote</code> and <code>\vl@dbfnote</code> as originals could give incorrect markers in the footnotes	213
<code>\new@line</code> : Extended <code>\new@line</code> to output page numbers	125
General: Fixed long paragraphs looping	1
Fixed minor typos	1
Prepared for <code>eledpar</code> package	1
v0.7.0.	
<code>\@nl@reg</code> : Added <code>\@nl@reg</code>	111
<code>\@ref@reg</code> : Added <code>\@ref@reg</code>	121
<code>\affixline@num</code> : Added <code>skipnumering</code> to <code>\affixline@num</code>	164
<code>\do@actions@fixedcode</code> : Added <code>\do@actions@fixedcode</code>	161
<code>\do@actions@next</code> : Added number skipping to <code>\do@actions</code>	159
<code>\do@insidelinehook</code> : Added <code>\do@linehook</code> for use in <code>\do@line</code>	157
<code>\endnumbering</code> : Changed <code>\endnumbering</code> for <code>eledpar</code>	93
<code>\f@x@l@cks</code> : Added <code>\ch@cksub@l@ck</code> , <code>\ch@ck@l@ck</code> and <code>\f@x@l@cks</code>	167
<code>\footsplitskips</code> : Added <code>\footsplitskips</code> for use in many footnote styles	178
<code>\get@linelistfile</code> : Added <code>\get@linelistfile</code>	109
<code>\initnumbering@reg</code> : Added <code>\initnumbering@reg</code>	91

\l@advance@parledgroup@beforenormalnotes: Added \l@dunboxmpfoot containing some common code	303
\l@dcsnotetext@r: Added \l@emptyd@ta	157
\l@dgetline@margin: Added \l@dgetline@margin	98
\l@dgetlock@disp: Added \l@dgetlock@disp	100
\l@dgetsidenote@margin: Added \l@dgetsidenote@margin	293
\l@dnumpstartsL: Added \l@dnumpstartsL, \ifl@dpairing and \ifpst@rted for/from eledpar	89
\l@drsn@te: Added \l@dlsn@te and \l@drsn@te for use in \do@line	158
\l@dzeropenalties: Added \l@dzeropenalties	152
\ledlinenum: Added \ledlinenum for use by \leftlinenum and \rightlinenum	102
\list@clearing@reg: Added \list@clearing@reg	109
\n@num: Added \n@num	118
\next@line@list@stuff: Deleted \page@start from \line@list@stuff	124
\normalbfnoteX: Removed extraneous space from \normalbfnoteX	219
\pausenumbering@page@num: Changed \resumenumbering for eledpar	94
\setprintendlines: Added \setprintendlines for use by \printendlines	242
\setprintlines: Added \setprintlines for use by \printlines	204
\skipnumbering: Added \skipnumbering and supports	130
\sublinenumincrement: Added \firstlinenum, \linenumincrement, \firstsublinenum and \linenumincrement	99
\sublinenumr@p: Using \linenumrep instead of \linenumr@p	101
Using \sublinenumrep instead of \sublinenumr@p	101
\vnumfootnoteX: Removed extraneous space from \vnumfootnoteX	220
General: eledmac having been available for 2 years, deleted the commented out original edmac texts	1
Maïeul Rouquette new maintainer	1
Made macros of all messages	82
Replaced all \interAfootnotelinepenalty, etc., by just \interfootnotelinepenalty	1
Tidying up for eledpar and ledarab packages	1
v0.8.0.	
General: Bug on endnotes fixed: in a // text, all endnotes will print and be placed at the ends of columns ()	1
v0.8.1.	
General: Bug on \edtext ; \critex ; \lemma fixed: we can now us non-switching commands	1
v0.9.0.	
General: No more ledpatch. All patches are now in the main file.	1
v0.9.1.	
General: Fix some bugs linked to integrating ledpatch on the main file.	1
v0.10.0.	
General: Corrections to \section and other titles in numbered sections	1
v0.11.0.	
General: Makes it possible to add a symbol on each verse's hanging, as in French typography. Redefines the command \hangingsymbol to define the character.	1
v0.12.0.	
\l@dnumpstartsL: Added \ifledRcol and \ifnumberingR for/from eledpar	89
General: For compatibility with eledpar, possibility to use \autopar on the right side.	1
Possibility to number \pstart.	19

Possibility to number the pstart with the commands <code>\numberpstarttrue</code>	1
v0.12.1.	
General: Don't number <code>\pstarts</code> of stanza.	1
The numbering of <code>\pstarts</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.	319
General: New <code>stanzaindentsrepetition</code> counter to repeat stanza indents every n verses.	52
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 memoir 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.	277
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.	164
<code>\ifinserthangingsymbol</code> : New management of <code>hangingsymbol</code> insertion, preventing undesirable insertions.	317
<code>\printlines</code> : Line numbering can be reset at each <code>pstart</code>	208
General: Line numbering can be reset at each <code>pstart</code>	96
Possibility to print <code>\pstart</code> number inside.	19
v0.17.0.	
<code>\ifinserthangingsymbol</code> : New new management of <code>hangingsymbol</code> insertion, preventing undesirable insertions.	317
v1.0.0.	
<code>\morenoexpands</code> : Change to be compatible with new features	133
General: <code>\lemma</code> can contain commands.	27
Debug in lineation command	21
New generic commands to customize footnote display.	37, 261
Options <code>nonum</code> and <code>nosep</code> in <code>\Xfootnote</code>	25
Options of <code>\Xfootnotes</code>	175
Possibility to have commands in sidenotes.	64
Some compatibility break with <code>eledmac</code> . Change of name: <code>eledmac</code>	1
v1.0.1.	
General: Correction on <code>\Xnumberonlyfirstinline</code> with lineation by <code>pstart</code> or by page. 39	
v1.1.0.	
<code>\Xprenotes</code> : New skip <code>\Xprenotes@</code>	234
<code>\settoggle@series</code> : <code>\settoggle@series</code> switch the global value of the toggle, not only the local value.	262
General: Add <code>\labelpstarttrue</code>	19
Add <code>\Xnumberonlyfirstintwolines</code>	39
Add <code>\Xpstart</code> and <code>\Xonlypstart</code>	41
New hook to add arbitrary code at the beginning of the notes	46
New options for block of notes.	48
New package option: <code>parapparatus</code>	1
New tools to change order of series	261
Sectioning commands.	71

v1.2.0.		
<code>\Xprenotes</code> :	Debug in familiar footnotes (bug introduced by v1.1).	234
<code>\endquote</code> :	Compatibility of <code>\ledchapter</code> with the <i>memoir</i> class.	356
v1.3.0.		
<code>\endquote</code> :	<i>Quotation</i> and quote environment inside numbered sections.	356
v1.4.0.		
<code>\edtext</code> :	Compatibility of <code>\edtext</code> with the right-to-left direction (with Polyglossia).	134
<code>\ledsetnormalparstuffX</code> :	Direction of footnotes with polyglossia.	231
<code>\newseries@</code> :	Remembers the language of the lemma, in order to create a correct direction for the footnote separator.	251
<code>\rbracket</code> :	Switch the right bracket to a left bracket when the lemma is RTL (needs polyglossia or LuaTeX).	198
General:	Compatibility with LuaTeX of RTL notes.	1
v1.4.1.		
<code>\affixside@note</code> :	Remove spurious spaces.	299
<code>\endquote</code> :	New option <i>noquotation</i> .	356
<code>\get@thisfootnote</code> :	Compatibility of standard footnotes with <code>eledmac</code> when these footnotes contain any commands.	213
<code>\labelrefsparsesubline</code> :	Fix a bug with <code>\edlabel</code> .	278
v1.4.2.		
General:	Debug with some special classes.	1
v1.4.3.		
General:	Add <code>\Xnonbreakableafternumber</code> .	42
Spurious space after familiar footnotes.		1
v1.4.4.		
General:	Label inside familiar footnotes.	1
v1.4.5.		
General:	Bug with <code>komascript</code> + <code>eledpar</code> + <code>chapter</code> .	1
v1.4.6.		
General:	Bug with <i>memoir</i> class introduced by 1.4.5.	1
v1.4.7.		
<code>\endquote</code> :	Compatibility of sectioning commands with <code>\autopar</code> .	356
v1.4.8.		
General:	Corrects a bug with parallel texts introduced by 1.1.	1
v1.4.9.		
<code>\normalbfnoteX</code> :	Allow to redefine <code>\thefootnoteX</code> with <code>alph</code> when some packages are loaded.	219
v1.5.0.		
<code>\do@insidelinehook</code> :	Added <code>\do@insidelinehook</code> for use in <code>\do@line</code> .	157
<code>\edindex</code> :	Compatibility with <code>imakeidx</code> package, and possibility to use multiple index with <code>\edindex</code> .	310
General:	Correct indexing when the call is made in critical notes.	306
v1.5.1.		
<code>\managestanza@modulo</code> :	Correct <code>stanzaindentsrepetition</code> counter.	319
<code>\normalvfootnoteX</code> :	Fix a bug with normal familiar footnotes when mixing RTL and LTR text.	216
v1.6.0.		
<code>\newverse</code> :	Add <code>\falseverse</code> macro.	322
v1.6.1.		
<code>\AtStartEveryPstart</code> :	Spurious space in <code>\pstart</code> .	148

\ifinserthangingsymbol: Hang verse is now not automatically flush right.	317
\l@dunhbox@line: Move the call to \inserthangingsymbol to allow use \hfill inside.	153
\pend: Spurious space in \pend.	150
General: Corrects a false hanging verse when a verse is exactly the length of a line.	1
v1.7.0.	
General: New features for managing page breaks.	72
v1.8.0.	
\endquote: Correction of sectioning commands in parallel texts.	356
\get@index@command: Debug \get@index@command and compatibility with hyperref package.	308
\newhookcommand@series@reload: Debug \beforenotesX and \maxnotesX which did not work.	265
\prevpage@num: Correct \parafootsep when using with ledgroup.	190
General: Compatibility with parledgroup option of eledpar package.	1
If imakeidx and hyperref are loaded, adds hyperref in the index.	306
v1.8.1.	
General: Debug endnotes when more than one series is used (change the position where tools for endnotes are defined).	236
v1.8.2.	
General: Debug compatibility problem with hebrew option of babel package.	1
v1.8.3.	
General: Fixes spurious spaces added by v1.7.0.	1
v1.8.5.	
General: Debug indexing in right column, with eledpar.	306
v1.9.0.	
\doxtrafeet: Add \fnpos to choice the order of footnotes.	270
\l@dfheetendmini: Add \mpfnpos to choice the order of footnotes in minipage / ledgroup.	301
v1.10.0.	
\endquote: Correction of sectioning commands in parallel texts.	356
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.	45
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 <code>\musixtex</code> , <code>\@l</code> becomes <code>\@1</code> . Consequently, \@l@reg becomes <code>\@nl@reg</code>	111
\AtStartEveryPstart: New optional argument for <code>\pstart</code> , to execute code before it.	148
\edindex: Use correctly default index when <code>imakeidx</code> is loaded.	310
\endquote: <code>\ledxxx</code> sectioning commands are deprecated and replaced by <code>\eledxxx</code> commands.	356
\initnumbering@reg: <code>\beginnumbering</code> is defined only on <code>eledmac</code> , not on <code>eledpar</code> .	91
\l@dgetsidenote@margin: <code>\sidenotemargin</code> is now directly defined in <code>eledmac</code> to be able to manage <code>eledpar</code>	293
\l@disnote: <code>\l@dlsnote</code> , <code>\l@drsnote</code> and <code>\l@dcsnote</code> defined only one time, in <code>eledmac</code> , including needs for <code>eledpar</code> case.	295
\l@dnumpstartsL: Add <code>\ifledRcol@</code> for <code>eledpar</code>	89
\l@dunhbox@line: <code>\do@line</code> is split in more little commands.	154
\newhookcommand@series@reload: Debug <code>\beforenotesX</code> and <code>\maxhnotesX</code> which did not work when called after <code>\footparagraphX</code>	265
Debug <code>\Xbeforenotes</code> and <code>\Xmaxhnotes</code> which did not work when called after <code>\footparagraph</code>	265
\pend: New optional argument for <code>\pend</code> , to execute code after it.	150
\stanza: <code>&can</code> have an optional argument: content to be printed after.	322
\Stanza can have an optional argument: content to be printed before.	322
Add <code>\newverse</code> macro, <code>\falseverse</code> deprecated.	322
General: Add <code>\ledinnernote</code> and <code>\ledouternote</code> commands.	64
Add <code>\Xendparagraph</code> and related settings.	50
Add hyperlink to <code>crossref</code> (needs <code>hyperref</code> package).	59
Compatibility with <code>musixtex</code>	1
Debug <code>eledmac</code> sectioning command after using <code>\resumenumbering</code>	1
Ensure that <code>imakeidx</code> is loaded <i>before</i> <code>eledmac</code>	306
New hooks: <code>\Xafterrule</code> and <code>\afterruleX</code>	49
New options for ragged-paragraph notes	48
New sectioning commands.	71
Optional arguments for <code>\pstart</code> and <code>\pend</code>	18
v1.12.1.	
\wrap@edcrossref: Fix spurious spaces.	282
v1.12.2.	
\l@dunhbox@line: Fix a bug with critical notes at the tops of pages (added by v12.0.0)	153
v1.12.3.	
\flag@end: <code>\flag@start</code> and <code>\flag@end</code> are now defined only one time for <code>eledmac</code> and <code>eledpar</code>	126
\flag@start send a error message when a <code>\edtext</code> is done without insert (note)	126
\reledmac@error: Replaced error messages	82
General: Add macros for new messages since v0.7	82
Correct bug with side and familiar notes in tabular environments.	1
Debug <code>\eledxxx</code> with some paper size	1
Debug <code>\ledinnernote</code> and <code>\ledouternote</code> commands in the top of pages.	64
Debug left and right notes (bugs added by 1.12.0)	1
Underline lemma in <code>\eledxxx</code> when using draft mode.	1
v1.12.4.	
General: Debug spurious page breaks before <code>\chapter</code> (bug added in 1.12.0)	1

v1.12.5.	
\@edindex@hyperref: Debug \edindex when hyperref is not loaded	312
\@ssect: Debug \eledchapter in parallel with memoir	359
\doinsidelinehook: Added \dolinehook and \doinsidelinehook	157
\endnumbering: Allow to mix parallel columns and normal text when using \pausenumbering	93
\l@dgobblearg: \l@dgobblearg becomes \l@gobbelloptarg	339
\l@drestoreforedtext: Debug optional arguments of \Xfootnote in tabular context	340
\pausenumbering@page@num: Debug \resumenumbering	94
v1.12.7.	
\wrap@edcrossref: \wrap@edcrossref is now robust	282
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	126
v1.13.0.	
\newhooktoggle@series@reload: Add \newhookcommand@toggle@reload	264
\para@footsetupX: In \para@footsetupX, use \columnwidth instead of \hsize	226
\settoggle@series: \settoggle@series can take an optional arguments to reload series setup.	262
General: Add \Xnoteswidthliketwocolumns and \noteswidthliketwocolumnsX	50
Added widthliketwocolumns option	78
v1.13.1.	
\ifat@every@pstart@star@: Add \l@dzeropenalties in \pstart	148
General: Coming back of page and line breaking penalties's management, deleted by error in v0.17.	1
Debug quotation environment inside of a \pstart preceded by a sectioning command.	1
v1.13.2.	
\l@dnumpstartsL: Add \ifl@dypaging for eledpar	89
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	310
General: Debug spurious characters before endnotes.	236
Delete previous override of \l@d@wrindexhyp at the beginning of a document when hyperref is not loaded.	314
Move gobbling command	88
Provide \@gobblefour	88
v1.14.1.	
\@ssect: Debug sectioning commands when using both handout and hyperref package.	362
v1.14.2.	
\@ssect: Debug \edtext after starred sectioning commands when using memoir class.	359
v1.15.0.	
\@edtext@level: New boolean \if@edtext@.	133
\arrangementX@threecol: Correct bug with paragraphed familiar footnotes setting.	226
\endsub: Restore subtitle feature (disabled by mistake in v1.8.0).	128
\if@lemmacommand@: New boolean \iflemmacommand@.	139

General: Fix a bug with footnotes layout when using some options of the geometry package (bug add by v1.13.0).	1
New commands <code>\AtEveryPstart</code> and <code>\AtEveryPend</code>	19
New tools to prevent ambiguous references in lemma	28
v1.15.1.	
<code>\next@line@list@stuff</code> : Revert modification of 1.5.2 which makes bug with numbering. Leave vertical mode to solve spurious space before minipage.	124
v1.16.0.	
General: <code>\edtext</code> is now defined only in <code>eledmac</code> , not in <code>eledpar</code> . Debug wrong numbering when using <code>\sameword + eledpar + \tag</code> command.	133
Compatibility of standard footnotes with some biblatex styles.	1
New <code>\stanzaindent</code> command.	1
v1.16.1.	
<code>\xlineref</code> : <code>\lineref</code> is not defined if defined by some other package, like <code>lineno</code> . Eledmac provides <code>\edlineref</code> instead.	282
v1.17.0.	
<code>\edtext</code> : Error message when calling <code>\edtext</code> outside of a numbered paragraph.	134
v1.18.0.	
<code>@edindex@hyperref</code> : Fix spurious space with <code>\edindex</code> when using <code>imakeidx/indextools + hyperref</code>	312
<code>\edlabel</code> : <code>\edlabel</code> is now defined only one time for both <code>eledmac</code> and <code>eledpar</code>	277
<code>\l@d@section</code> : Option <code>parapparatus</code> works for endnotes.	237
<code>\l@dnumprintstartL</code> : Add <code>\ifl@dprintingpages</code> and <code>\dprintingcolumns</code> for <code>eledpar</code>	89
<code>\print@line</code> : Compatibility with Lua \TeX RTL languages.	154
<code>\printlinefootnote</code> : Code refactoring in <code>\printlinefootnote</code> : the printing of the numbers are factorized in <code>\printlinefootnotearea</code>	200
<code>\printpstart</code> : Debug <code>\Xpstart</code> with parallel pages and columns (<code>eledpar</code>)	199
General: Add <code>\Xpstarteverytime</code>	41
Compatibility with Lua \TeX RTL languages.	1
Debug <code>\Xonlypstart</code> when using <code>\Xnumberonlyfirstinline</code> and the current line number differs from the previous.	41
v1.19.0.	
<code>\footsplitskips</code> : <code>\footsplitskips</code> doesn't set <code>\floatingpenalty</code> to <code>\@MM</code> when processing parallel pages.	178
<code>\xxref</code> : <code>\xxref</code> works also with right side numbers, when <code>\@Rlineflag</code> is not empty.	285
General: <code>\Xmaxhnotes</code> and <code>\maxhnotesX</code> work now for both two-columns and three-columns setting.	1
Compatibility with <code>eledpar v1.13.0</code>	1
v1.19.1.	
General: Call <code>\correct@footinsX@box</code> and <code>\correct@Xfootins@box</code> directly in <code>\print@notesX@forpages</code> and <code>\print@Xnotes@forpages</code> , that is in <code>eledpar</code>	1
v1.20.0.	
<code>\printlines</code> : Added <code>\ifl@d@Xmorethantwolines</code> and <code>\ifl@d@Xmorethantwolines</code> to <code>\printlines</code>	209
<code>\stanza</code> : <code>&</code> and <code>\&</code> can be preceded by spaces.	322
<code>\xxref</code> : Debug <code>\xxref</code> when not loading <code>eledpar</code> (Fix a bug added in 1.19.0).	285
General: Add <code>\Xendboxlinenum</code>	43
Add <code>\Xtwolines</code> and <code>\Xmorethantwolines</code> hooks	40
Add series option.	1

Correct <code>\Xinplaceofnumber</code> hook.	1
Explicit error message when calling <code>\Xfootnote</code> outside of <code>\edtext</code>	1
Fix a bug with line number typesetting direction when using <code>\eledsection</code> and similar commands for RTL texts with <code>Lua\TeX</code>	1
Fix issues with RTL text in notes when using <code>Lua\TeX</code>	1
Options <code>fulllines</code> in <code>\Xfootnote</code>	25
The <code>\newifs</code> are not followed by boolean values set to <code>false</code> , because it is the <code>\TeX</code> default setting.	1
v1.21.0.	
<code>\edindex@hyperref</code> : Look at the <code>hyperindex</code> option of <code>hyperref</code> before inserting <code>hyperref</code>	312
<code>\l@d@section</code> : <code>\endnotes</code> take five arguments.	237
<code>\ledinnotemark</code> : Add <code>\ledinnotemark</code>	309
<code>\ledsetnormalparstuffX</code> : <code>\ledsetnormalparstuff</code> is deprecated and becomes <code>\ledsetnormalparstuffX</code> and <code>\Xledsetnormalparstuff</code>	231
<code>\n@num</code> : <code>\n@num@ref</code> deleted	118
<code>\n@num</code> defined only one time for both <code>Eledmac</code> and <code>Eledpar</code>	118
<code>\newhookcommand@series</code> : <code>\newhookcommand@series</code> can take an optional argument.	264
<code>\newhooktoggle@series</code> : <code>\newhooktoggle@series</code> can take an optional argument.	264
<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>	233
<code>\seriesatend</code> : Fix spurious space in <code>\seriesatend</code>	261
<code>\skipnumbering</code> : <code>\skipnumbering</code> defined only one time for both <code>Eledmac</code> and <code>Eledpar</code>	130
Correct <code>\skipnumbering</code> for stanza.	130
Delete <code>\skipnumbering@reg</code>	130
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	261
Add <code>\applabel</code> and related	61
Add <code>\beforenotesX</code> and <code>\Xbeforenotes</code> features for notes set in two and three column.	1
Add <code>\hidenumbering</code>	23
Add <code>\Xcolalign</code> and <code>\colalignX</code> hooks	47
Add <code>\Xendtwolines</code> , <code>\Xendmoreethantwolines</code> , <code>\Xendtwolinesbutnotmore</code> and <code>\Xendtwolinesonlyinsamepage</code>	41
Add <code>\Xparindent</code> and <code>\hangindentX</code>	46
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>\maxnotesX</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>\maxnotesX</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>oldprintpnumspace</code> option allows to come back to previous setting	1
<code>parapparatus</code> option works now with familiar footnotes.	1
Provide <code>\@gobblefive</code>	88
v1.22.0.	
<code>\ledinnote</code> : <code>\ledinnote</code> takes a first optional argument, which is the label for hyperlinks.	309
General: Add <code>\doendnotesbysection</code> command.	26
Add option for lemma separator inside <code>endnotes</code>	44
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 <code>endnotes</code> handbook.	26
When using <code>hyperref</code> package, internal links in index or with <code>\edlineref</code> are now targeted to the top and not longer to the bottom of the lines they refer to.	1
v1.22.1.	
<code>\prevpage@num</code> : Correct double symbol when using both <code>\parafootsep</code> and <code>\Xsymlinenum</code>	190
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>	133
<code>\Serefwithpage</code> : Debug <code>\Xendtwolines</code> , <code>\Xendmoreethantwolines</code> , <code>\Xendtwolinesbutnotmore</code> and <code>\Xendtwolinesonlyinsamepage</code> when using <code>\apprefwithpage</code>	289
<code>\lemma</code> : Fix spurious space after <code>\lemma</code> command	138
<code>\newseries@</code> : Prevent spurious spaces when <code>\Afootnote</code> and similar commands are followed by spaces (bug added on 1.0.0).	251
<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.	144
General: Add <code>\Xboxlinenumalign</code> and <code>\Xendboxlinenumalign</code>	43
Add <code>\Xboxstartlinenum</code> , <code>\Xendboxstartlinenum</code> , <code>\Xboxendlinenum</code> , <code>\Xendboxendlinenum</code>	43
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.	28
Fix a bug with <code>\sameword</code> when the same lemma is used for multiple notes or for nested <code>\edtexts</code>	28
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.	28
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 \AtEveryPstart and \AtEveryPend providing to it an empty argument.	1
v1.24.1.	
General: \lemma is disabled when using 'nocritical' option.	1
v1.24.2.	
General: Fix incompatibility between 'nofamiliar' option and 'memoir' package.	1
v1.24.3.	
General: Restore marginal numbers and notes with sectioning command (bug introduced in v1.21.0)	1
v1.24.4.	
General: Fix spurious space with \edindex when using xindy+hyperref option.	1
v1.24.5.	
General: Fix a bug of indent, when a added in 1.1.0, when a \beginnumbering immediately follow a sectioning command.	1
v2.0.0.	
\@iiiminipage : Patch \@iiiminipage instead of redefining it.	303
\@xympar : Patching \@xympar instead of redefining it	293
\endminipage : Patch \endminipage instead of redefining it.	303
$\text{\@initnumbering@quote}$: $\text{\@initnumbering@sectcmd}$ becomes $\text{\@initnumbering@quote}$	356
$\text{\@l@advance@parledgroup@beforenormalnotes}$: Some code of \@l@dumboxmpfoot moved to $\text{\@l@advance@parledegroupp@beforenormalnotes}$	303
\@newsseries@ : One endnotes file by series.	257
General: \@makecol , \@reinserts and \@doclearpage are patched instead of begin redefined	274
\@doxtrafeeti becomes \@Xdo@feet ; \@doxtrafeetii becomes \@do@Xfeet ; \@opxtrafeeti becomes \@opfeetX ; $\text{\@doreinxtrafeetii}$ becomes \@X@doreinfeet ; \@doreinxtrafeeti becomes \@doreinfeetX	274
Add $\text{\@Xendinplaceofnumber}$ hook.	1
Add \@Xendnonumber hook.	1
Add nonum option for endnotes.	1
Fix a bug when printing only one series of endnotes, but wanted to keep endnotes for other series.	1
In order to have a more consistent name's convention, many names has been changed.	1
Many \LaTeX 's output macros are now patched and not override.	1
Package's name becomes \reledmac	1
Patch \@footnotemark instead of redefine it	212
Suppress indexing command specific to memoir.	309
v2.0.1.	
General: Fix a bug in \eledmac-compat option	1
Fix incompatibility between optional argument of \pstart and \numberpstarttrue	1
v2.1.0.	
General: Fix a bug with \advanceline at the beginning of a \pstart	1
Fix a bug with \chapter in optional argument of \pstart in parallel typesetting with \scrbook	1
Fix a bug with \eledchapter in parallel typesetting with \scrbook	1
Fix a bug with \setline at the beginning of a \pstart	1

Fix spacing bug with <code>\Xbhooknote</code> and <code>\bhooknoteX</code> when using them to insert text and not to execute code.	1
New tools to number stanzas	1
v2.1.1.	
General: Fix a bug with <code>\ledpbsetting{before}</code>	1
v2.1.2.	
General: Fix a bug with lineation by <code>pstart</code> and <code>tabular</code> environments (added in 2.1.0).	1
v2.1.3.	
<code>\ledsetnormalparstuffX</code> : Replaced <code>\noindent</code> with <code>\parindent</code> set to 0pt.	231
General: <code>\Xhangindent</code> and <code>\hangindentX</code> work now with all the paragraphs in the note.	1
<code>\Xnoindent</code> and <code>\noindentX</code> work now again (broken in 2.0.0).	1
Change some internal code in order to provide compatibility with \LaTeX release of october 2015	1
Fix a bug which inserted double space before paragraphed familiar notes.	1
Fix a bug with <code>\edindex</code> when using not-Latin characters without UTF-8 engines	1
v2.2.0.	
General: Fix a bug with combination of <code>\onehalfspacing</code> and two columns and three columns notes typeset.	1
Fix a bug with some setting command and optimization option.	1
Fix spurious space with paragraphed critical notes when using \LaTeX	1
Increase line list version number to ensure compatibility with new options of <code>reledpar</code> package.	1
New setting tools for endnotes: <code>\Xendnumberonlyfirstinline</code> , <code>\Xendnumberonlyfirstintwolines</code> , <code>\Xendsymmlinenumber</code> , <code>\Xendbeforenumber</code> , <code>\Xendaftersymmlinenumber</code> , <code>\Xendbeforemsymmlinenumber</code> , <code>\Xendaftersymmlinenumber</code> , <code>\Xendboxsymmlinenumber</code> , <code>\Xendhangindent</code> , <code>\Xendbhooklinenumber</code> , <code>\Xendahooklinenumber</code> , <code>\Xendbhookinplaceofnumber</code> , <code>\Xendahookinplaceofnumber</code>	1
v2.2.1.	
General: Compatibility with \LaTeX format 2015/10/01.	1
v2.2.2.	
General: Fix a bug in <code>\sethangingsymbol</code>	1
Fix a bug with old version of <code>etex</code>	1
v2.3.0.	
General: Disable empty lines as paragraph in stanza.	1
Fix incompatibility of paragraphed footnotes with <code>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.	175
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>	40
Option <code>linrangeseq</code> for critical footnotes and endnotes.	40
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>	288
<code>\edlabel</code> : Fix a bug when calling <code>\edlabel</code> in a footnotes of the rightside	277
<code>\ld@section</code> : <code>\endnotes</code> take six arguments.	237
<code>\printlines</code> : <code>\printlines</code> takes an eighth argument: the line flag	208
<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.	282
General: <code>\apprefwithpage</code> and <code>\appref</code> print double quotation mark when the label was not defined.	1
<code>\apprefwithpage</code> and <code>\appref</code> work with right side crossref.	1
<code>\apprefwithpage</code> works also when <code>noend</code> option is enabled.	1
<code>\appref</code> and <code>\apprefwithpage</code> can take <code>linrangeseq</code> optional argument.	1
<code>\edlabel</code> works now in <code>\Xfootnote</code>	1
<code>\lemma</code> can be used even when the <code>nocritical</code> is enabled.	1
Compatibility with new hook and tools of <code>reledpar</code> 2.6.0.	1
Fix spurious vertical space in <code>astanza</code> environment (<code>reledpar</code>)	1
Log now states ‘There were undefined references’ when using wrong references in <code>\edlineref</code> or <code>edpageref</code>	1
New hooks to customize page and line number appearance in endnotes.	1
New hooks: <code>\Xbhookgroup</code> and <code>\bhookgroupX</code>	1
New tools to easily make cross-reference to a passage defined by a start and an end line	61
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>	191
<code>\ld@section</code> : <code>\endnotes</code> take seven arguments.	237
General: Add dash as default page range separator for <code>\SEonlypage</code>	1
Debug <code>\SErefonlypage</code> when referring to only one page.	1
Delete parenthesis after <code>\SErefonlypage</code>	1
Fix (again) bugs with footnote numbering in parallel typesetting while using <code>ledgroup</code> environments (bug added in v2.5.0).	1
Fix a bug with <code>\SErefwithpage</code>	1
Fix bugs in compatibility with <code>innnote</code> and <code>notenumber</code> options of <code>indextools</code> package, when indexing outside of a <code>ledgroup</code>	1
New commands to make glossaries connected to page and linenumber with the <code>glossaries</code> package	1
New hooks: <code>\Xhsize</code> and <code>\hsizeX</code>	50

New hooks: <code>\Xlemmafont</code> and <code>\Xendlemmafont</code>	45
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 related, <code>\appref</code> and related.	1
When using <code>hyperref</code> package, add links from critical footnotes and critical endnotes to the line of text they refers	1
v2.7.1.	
General: Debug <code>\Xbhookgroup</code> hooks executed on columnar footnotes (moved to a larger group, to take effect).	1
v2.7.2.	
General: <code>\Xhsize</code> and <code>\hsizeX</code> become <code>\Xwidth</code> and <code>\widthX</code>	50
Fix problem of hyphenation when using <code>hyperref</code> package (added in v2.7.0).	1
v2.8.0.	
<code>\l@d@section</code> : <code>\Xendhangindent</code> and <code>\Xendafternote</code> can take values which are relative to the font size of the endnote.	237
General: <code>reledmac</code> cross-referencing can take advantage of <code>xr</code> package.	1
More <code>\edgls...</code> commands.	1
No indentation for paragraphed notes in <code>ledgroup</code> . Can be changed with <code>\Xparindent</code> and <code>\parindentX</code>	1
v2.8.1.	
General: Warnings for undefined labels are really parsable by latexmk	1
v2.8.2.	
General: Fix a bug concerning indent in a paragraph immediately following a sectioning command (bug NOT fixed on <code>reledpar</code>)	1
Fix a bug with <code>\AtEveryPstart</code> added in version 2.0.0.	1
Fix a bug with vertical space after the between-sectioning command as optional argument of a <code>\pstart</code> and <code>\pstart</code> content	1
v2.9.0.	
General: Allow continuing line numbering between normal text and parallel text, using <code>\pausenumbering</code> and <code>\resumenumbering</code> and the <code>continuousnumberingwithcolumns</code> option.	1
Fix a bug when using <code>\lineneation{page}</code> and <code>\pausenumbering...resumenumbering</code>	1
Fix a bug with three- and two-column footnote setting (added in v2.4.0).	1
Fix spurious space inside three-column familiar footnote.	1
Write correct metadata in numbered files when using <code>\pausenumbering...resumenumbering</code>	1
v2.9.1.	
General: Fix a bug when notes start with “plus” or “minus”.	1
v2.9.2.	
General: Fix a bug with <code>hyperref</code> package when a lemma starts with “plus” or “minus” (bug introduced in v. 2.7.0).	1
v2.9.3.	
General: Fix a bug with line number position and reset added by v. 2.9.0	1

v2.10.0.	
\print@lemma: Code refactoring between \parafootfmt, \twocolfootfmt, \threecolfootfmt and \normalfootfmt.	199
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.	46
New hooks: \Xwraplemma and \Xendwraplemma	46
v2.10.1.	
General: Add ‘nopenalties’ option.	1
Fix a bug introduced in v. 1.4: not paragraphed critical footnotes could prevent marginal line number from being displayed	1
v2.11.0.	
\do@actions@fixedcode: Add action 1010	161
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>	274
Add <code>\Xendpagenumberonlyfirst</code> , <code>\Xendpagenumberonlyfirstifsingle</code> , <code>\Xendpagenumberonlyfirstintwo</code> , <code>\Xendinplaceofpagenumber</code> and <code>\Xendsympagenum</code> hooks.	1
Add <code>\Xpagelinesep</code> hook.	42
Compatibility with new features of <code>reledpar</code>	1
Deleted dead code.	1
Display a warning message if using a version of \TeX that is too old.	1
Fix a bug with <code>\Xgroupbylines</code> for notes in two columns	1
Fix a bug with <code>\Xtxtbeforenotes</code> for notes in three or two columns	1
Fix a bug with ‘notenumber’ option of <code>indextools</code> package when indexing texts in familiar footnotes.	1
Fix potential bug when using <code>\edindex</code> in critical footnotes.	1
More explicit error message in case the stanza indentation is not defined.	274
New options for <code>\fnpos</code> and <code>\mpfnpos</code> to set a customized order for familiar and critical footnotes.	1
When <code>\edindex</code> is called outside of a <code>\beginnumbering...endnumbering</code> structure, it is automatically switched to <code>\index</code> , with a warning message.	1
When indexing texts in familiar footnotes with <code>\edtext</code> , refer to the line number where the footnote is called.	1
When indexing texts in sidenotes with <code>\edtext</code> , refer to the line number where the sidenote is called.	1
v2.16.1.	
General: Fix a bug with redefinition of the style of the footnote number (bug added in v2.12.0)	1
v2.16.2.	
General: Error message if <code>footmisc</code> is loaded after <code>reledmac</code> .	1
Fix a bug introduced by v2.16.1 when using non-expandable control sequence, like <code>\normalfont</code> , in the footnote number style.	1
v2.16.3.	
General: Fix a bug with <code>\SEref</code> (bug added in v2.7.0).	1
v2.16.4.	
General: Fix a bug with vertical space before sectioning command in optional argument of <code>\pstart</code> (bug added in v2.15.3).	1

v2.16.5.		
	General: Fix potential spurious spaces in endnotes.	1
v2.16.6.		
	General: Fix a bug with the line number style in <code>\doenotes</code> when referring to right side line in parallel typesetting.	1
	Take into account <code>\linenumberstyle</code> when using <code>\edlineref</code>	1
v2.16.7.		
	General: Fix a bug with <code>\msdata</code> when using multiple <code>\beginnumbering ... \endnumbering</code>	1
	Fix a bug with <code>\numberpstarttrue</code> when using multiple <code>\beginnumbering ... \endnumbering</code>	1
v2.16.8.		
	General: Fix a bug with <code>\edindex</code> in footnotes in parallel typesetting.	1
v2.17.0.		
	General: Add <code>\edglsadd</code> command.	1
	Add <code>\setmsdataposition</code> setting.	1
v2.17.1.		
	General: Fix spurious space in paragraphed footnotes when using Lua \TeX without using Right-To-Left text.	1
v2.17.2.		
	General: Change log message when numbered files still don't exist, in order to improve compatibility with <i>latexmk</i>	1
v2.17.3.		
	General: Fix a bug with <code>\doendnotesbysection</code> and <code>\doendnotes</code>	1
v2.17.4.		
	General: Fix a bug with <code>\setSErefonlypageprefixsingle</code> and <code>\setSErefonlypageprefixmore</code>	1
v2.17.5.		
	General: Fix a bug with <code>\pstartref</code> when referring to the left side in parallel typesetting.	1
v2.18.0.		
	General: Fix a bug when using a <code>\edtext</code> in two lines or more in right-to-left typesetting with X \TeX	1
	Fix a bug when using both <code>\Xnumberonlyfirstintwolines</code> or <code>\Xnumberonlyfirstinline</code> and <code>\Xparafootsep</code> and <code>\Xsymlinenum</code>	1
v2.18.1.		
	General: Fix a bug when using <code>\msdata</code> with Lua \TeX or with the <code>hyperref</code> package.	1
v2.19.0.		
	General: Add <code>\footnoteXmark</code> and <code>\footnoteXtext</code> commands.	1
	Add better compatibility with the <code>csquotes</code> package when using familiar footnotes.	1
	Fix a bug with paragraph indent after sectioning command.	1
v2.20.0.		
	General: Add <code>\AtStartEveryStanza</code> , <code>\BeforeEveryStopStanza</code> , <code>\AtEndEveryPend</code> , <code>\AtStartEveryPstart</code>	1
	Add second optional argument to <code>\pstart</code> , <code>\pend</code> and <code>\stanza</code>	1
	Add starred version of <code>\AtEveryPstart</code> , <code>\AtEveryPend</code> , <code>\AtEveryStanza</code> and <code>\AtEveryStopStanza</code>	1
	Add third and fourth optional argument to <code>\newverse</code>	1
	Fix a bug when using familiar footnotes in <code>\eledsection</code> and related.	1
	Reset font specification at the beginning of familiar footnotes.	1

v2.21.0.	
General: Add the possibility of nested <code>\sameword</code> commands.	1
Fix a bug when using formatting command in the argument of <code>\edindex</code> inside <code>\edtext</code>	1
Now, as explained in the handbook, a <code>\edindex</code> inside <code>\edtext</code> only creates an index reference to the main text, and not to the critical footnote.	1
v2.22.0.	
General: Add <code>\txtbeforenotesonlyonceX</code> and <code>\Xtxtbeforenotesonlyonce</code> hooks. . .	1
Add <code>\txtbeforenotesX</code> hook.	1
Fix a bug added in v2.16.0 when using <code>\Xtxtbeforenotes</code> with paragraphed or normal footnotes.	1
Fix a bug with three and two columns critical footnotes, broken in v. 2.17.6.	1
v2.22.1.	
General: Compatibility with new version of <code>reledpar</code>	1
Fix a bug with some commands inside <code>\sameword</code>	1
v2.22.2.	
General: Do not print footnotes at the first run.	1
v2.23.0.	
General: Add <code>swcaseinsensitive</code> option	1
Optimisation of the code added on v2.22.2 to not print footnotes at the first run.	1
v2.24.0.	
<code>\next@line@list@stuff</code> : Add <code>\next@line@list@stuff</code> internal hook.	124
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
v2.26.0.	
General: Add a mechanism of line number annotation for stanza	1
Fix bug in <code>\xpstartref</code>	1