

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	13
1.2.3 eledmac	13

*This file (**reledmac.dtx**) has version number v2.11.0a, last revised 2016/06/03.

[†]maieul at maieul dot net

1.2.4 <code>reledmac</code>	13
1.3 List of works edited with (r)(e)ledmac	14
2 How the package works	14
3 Compatibility warning	14
4 Options	14
4.1 Specific features	15
4.2 Optimizing package performance	15
5 Text lines and paragraphs numbering	16
5.1 Text lines numbering	16
5.2 Paragraphs	16
5.2.1 Basics	16
5.2.2 Automatically producing <code>\pstart ... \pend</code>	17
5.2.3 Content before specific <code>\pstart</code> and after specific <code>\pend</code>	17
5.2.4 Content before every <code>\pstart</code> and after every <code>\pend</code>	18
5.2.5 Numbering paragraphs (<code>\pstart</code>)	18
5.2.6 Languages written in Right to Left	18
5.2.7 Memory limits	18
5.3 Lineation commands	19
5.3.1 Disabling lineation	19
5.3.2 Setting lineation start and step	19
5.3.3 Setting lineation reset	20
5.3.4 Setting line number margin	20
5.3.5 Other settings	20
5.4 Changing the line numbers	20
5.4.1 Sublineation	20
5.4.2 Locking lineation	21
5.4.3 Setting and changing line number	21
5.4.4 Line number style	21
5.4.5 Skipping and hiding number	22
5.4.6 Execute code at each line	22
6 Apparatus commands	22
6.1 Terminology	22
6.2 Critical notes	23
6.2.1 The lemma	23
6.2.2 Footnotes	23
6.2.3 Endnotes	24
6.2.4 Paragraph in critical apparatus	25
6.2.5 Change lemma and line number	25
6.2.6 Changing the names of commands for critical apparatus	26
6.3 Disambiguation of identical words in the apparatus	26
6.3.1 Basic use	26

6.3.2 Notes about input encoding with UTF-8 processor	27
6.3.3 Use with <code>\lemma</code> command	27
6.3.4 Customizing	28
6.4 Apparatus of Manuscripts	29
6.4.1 Marking sections of text	29
6.4.2 Layout of the apparatus of manuscripts	29
6.4.3 Settings	30
6.5 Familiar notes	30
6.5.1 Basic use	30
6.5.2 Customizing mark	30
6.5.3 Separator for multiple footnotes	31
6.6 Changing series	31
6.6.1 Create a new series	31
6.6.2 Delete series	31
6.6.3 Series order	31
6.7 Position of critical and familiar footnotes	31
7 Critical apparatus appearance	32
7.1 Notes arrangement in a series	32
7.2 Control line number printing	33
7.2.1 Print line number only at first time	33
7.2.2 Arbitrary text before line number	33
7.2.3 Separator for line range	33
7.2.4 Abbreviate line range	33
7.2.5 Disable line number	34
7.2.6 Printing pstart number	34
7.2.7 Printing stanza number	35
7.2.8 Separator between line and subline numbers	35
7.2.9 Space around number	35
7.2.10 Space around line symbol	35
7.2.11 Space in place of number	36
7.2.12 Boxing line number and line symbol	36
7.3 For endnotes	37
7.4 Arbitrary code around line number	37
7.5 Separator between the lemma and the note	37
7.5.1 For footnotes	37
7.5.2 For endnotes	38
7.6 Font style	38
7.6.1 For line number	38
7.6.2 For the lemma	38
7.6.3 For all notes	39
7.7 Wrapping notes	39
7.7.1 Wrapping lemmas	39
7.7.2 Wrapping contents	39
7.8 Indent of notes content	39
7.9 Arbitrary code at the beginning of notes	40

7.10 Arbitrary code before inserting note	40
7.11 Options for footnotes in columns	40
7.11.1 Alignment	40
7.11.2 Size of the columns	41
7.12 Options for paragraphed footnotes	41
7.12.1 Mark separation of notes	41
7.12.2 Ragged text	41
7.13 Options for block of notes	41
7.13.1 Text before notes	41
7.13.2 Code before notes	41
7.13.3 Spacing	42
7.13.4 Rule	42
7.13.5 Maximum height	42
7.13.6 Width	43
7.14 Footnotes and the <code>reledpar</code> columns	43
7.15 Endnotes in one paragraph	43
8 Fonts	43
9 Verse	44
9.1 Basic	44
9.2 Define stanza indents	44
9.3 Repeating stanza indents	45
9.4 Manual stanza indent	45
9.5 Stanza breaking	46
9.6 Hanging symbol	46
9.7 Long verse and page break	46
9.8 Content before/after verses	46
9.9 Numbering stanza	47
9.10 Various tools	47
9.11 Notes on empty lines	47
10 Grouping	48
11 Cross referencing	48
11.1 Basic use	48
11.2 Cross-referencing to a critical note	49
11.3 Cross-referencing which return a number in any case	49
11.3.1 Cross-referencing in order to define line number of a critical note	50
11.4 Not automatic cross-referencing	50
11.5 Normal \LaTeX cross-referencing	50
11.6 References to start and end lines	50
11.6.1 Reference to main text lines	50
11.6.2 References to lines that are commented on in the apparatus	51
11.6.3 Settings	51
11.7 Compatibility with <code>xr</code> package	53

12 Side notes	53
12.1 Basics	53
12.2 Setting	53
12.2.1 Width	53
12.2.2 Vertical position	53
12.2.3 Distance to the main text	54
12.2.4 Separator between notes	54
13 Indexing	54
13.1 Basics	54
13.2 Referring to critical notes	54
13.3 Separator between page and line numbers	55
13.4 Using xindy	55
13.5 Advanced setting	56
14 Glossary	56
14.1 Preamble setting	56
14.2 Commands	56
15 Tabular material	57
16 Sectioning commands	60
16.1 Sectioning commands without line numbers or critical notes	60
16.2 Sectioning commands with line numbering and critical notes	60
16.3 Optimization	61
17 Quotation environments	61
18 Page breaks	61
18.1 Control page breaking	61
18.2 Prevent page break in a long verses	62
19 Miscellaneous	62
19.1 Known and suspected limitations	63
19.1.1 Non-standard geometry	63
19.1.2 floatrow package compatibility	63
19.1.3 ‘No room for a new’	63
19.1.4 Marginal notes	63
19.1.5 Paragraph shape	64
19.1.6 Paragraphed footnotes	64
19.1.7 Use with other packages	64
19.1.8 Parallel typesetting	65
I Implementation overview	66

II Preliminaries	66
II.1 Links with original edmac	66
II.2 Package declaration	66
II.3 Package options	67
II.4 Loading packages	68
II.5 Compatibility with Lua \TeX	69
II.6 Boolean flags	69
II.7 Messages	70
II.8 Gobbling	76
II.9 Miscellaneous commands	76
II.10 Prepare reledpar	77
II.11 Booleans provided by other optional packages which are required in any case	77
III Sectioning commands	77
IV List macros	81
V Line counting	83
V.1 Choosing the system of lineation	83
V.2 Line number margin	84
V.3 Line number initialization and increment	85
V.4 Line number locking	87
V.5 Line number style	88
V.6 Line number printing	88
V.7 Line number counters and lists	89
V.8 Line number locking counter	90
V.9 Line number associated to lemma	90
V.10 Reading the line-list file	94
V.11 Commands within the line-list file	96
V.12 Writing to the line-list file	107
VI Marking text for notes	112
VI.1 <code>\edtext</code> itself	113
VI.2 Substitute lemma	120
VI.3 Substitute line numbers	121
VI.4 Lemma disambiguation	121
VII Paragraph decomposition and reassembly	127
VII.1 Boxes, counters, <code>\pstart</code> and <code>\pend</code>	128
VII.2 Processing one line	133
VII.2.1 General process	133
VII.2.2 Process for “normal” line	133
VII.2.3 Process for line containing <code>\eledsection</code> command	135
VII.2.4 Hooks	135
VII.2.5 Sidenotes and marginal line number initialization	136

VIII Line and page number computation	137
IX Line number printing	140
X Pstart number printing in side	144
XI Restoring footnotes and penalties	145
XI.1 Add insertions to the vertical list	145
XI.2 Penalties	146
XI.3 Printing leftover notes	147
XI.4 Text before notes	148
XII Critical footnotes	149
XII.1 Fonts	149
XII.2 Individual note options	149
XII.3 Notes language	150
XII.4 General survey of the way we manage notes	151
XII.5 General setup	151
XII.6 Footnotes arrangement	152
XII.6.1 User level macro	152
XII.6.2 Normal footnote	152
XII.6.3 Paragraphed footnotes	157
XII.6.4 Columnar footnotes	164
XII.7 Critical notes presentation	170
XII.7.1 Font tools	170
XII.7.2 Pstart number in footnote	170
XII.7.3 Lemma printing	171
XII.7.4 Line number printing	172
XIII Familiar footnotes	180
XIII.1 Adjacent footnotes	180
XIII.2 Regular footnotes for numbered texts	182
XIII.3 Footnote formats	184
XIII.4 Footnote arrangement	184
XIII.4.1 User level macro	184
XIII.4.2 Normal footnotes	184
XIII.4.3 Two columns footnotes	190
XIII.4.4 Three columns footnotes	192
XIII.4.5 Paragraphed footnotes	194
XIII.5 Wrapping footnote marks in hyperlink	199
XIV Code common to both critical and familiar footnote in normal arrangement	200
XV Footnotes' width for two columns	200
XVI Footnotes' order	202

XVII Footnotes' rule	202
XVIII Specific skip for first series of footnotes	202
XVIII.0.1 Overview	202
XVIII.0.2 User level command	203
XVIII.0.3 Internal commands	203
XIX Endnotes	205
XX Generate series of notes	213
XX.1 Test if series is still existing	213
XX.2 Init specific to <code>reledpar</code>	213
XX.3 For critical footnotes	214
XX.3.1 Options	214
XX.3.2 Create inserts, needed to add notes in foot	215
XX.3.3 Create commands for critical apparatus, <code>\Afootnote</code> , <code>\Bfootnote</code> etc.	215
XX.3.4 Set standard display	218
XX.4 For familiar footnotes	218
XX.4.1 Options	218
XX.4.2 Create tools for familiar footnotes (<code>\footnotex</code>)	219
XX.5 The endnotes	220
XX.5.1 The auxiliary file	220
XX.5.2 The main macro	220
XX.5.3 The options	221
XX.6 Init standards series (A,B,C,D,E)	223
XXI Setting series display	223
XXI.1 Change series order	223
XXI.2 Test series order	223
XXI.2.1 Get the first series	224
XXI.3 Series setting	224
XXI.3.1 General way of working	224
XXI.3.2 Tools to set options	224
XXI.3.3 Tools to generate options commands	226
XXI.3.4 Options for critical notes	227
XXI.3.5 Options for familiar notes	229
XXI.3.6 Options for endnotes	230
XXI.4 Hooks for a particular footnote	231
XXI.5 Alias	232
XXII Output routine	232
XXII.1 Page number management	232
XXII.2 Extra footnotes output	232
XXII.3 Patching standard output of commands	235

XXIII Cross referencing	237
XXIII.1 Compatibility with <code>xref</code>	250
XXIV Side notes	251
XXV Minipages and such	258
XXVI Indexing	262
XXVI.1 Looking on package order	262
XXVI.2 Auxiliary macros for <code>\edindex</code>	263
XXVI.3 Code specific to <code>\edindexin</code> critical footnotes	264
XXVI.4 Analysis of command in indexed text	265
XXVI.5 Code for the formatted index	265
XXVI.6 Main code	266
XXVI.7 Hyperlink	267
XXVI.8 ‘innote’ and ‘notenumber’ option of <code>indextools</code> package	270
XXVII Glossaries	270
XXVIII Verse	272
XXVIII.1 Hanging symbol management	272
XXVIII.2 Using <code>&</code> character	273
XXVIII.3 Code category setting	273
XXVIII.4 Stanza count and indent	273
XXVIII.5 Numbering stanza	275
XXVIII.6 Stanza number in note	276
XXVIII.7 Main work	276
XXVIII.8 Restore catcode and penalties	279
XXIX Apparatus of Manuscripts	279
XXIX.1 User level macro	279
XXIX.2 Setting macro	280
XXIX.3 Counters and lists	280
XXIX.4 Auxiliary file macros	281
XXIX.5 Action macro	282
XXIX.6 Inserting footnote	286
XXIX.7 Other	286
XXX Arrays and tables	287
XXX.1 Preamble: macro as environment	287
XXX.2 Tabular environments	290
XXX.2.1 Disabling and restoring commands	291
XXX.2.2 Counters, boxes and lengths	294
XXX.2.3 Tabular typesetting	298
XXX.2.4 Environments	309
XXXI Quotation’s commands	309

XXXII Section's title commands	310
XXXII.1 Commands to disable some feature	310
XXXII.2 General overview	311
XXXII.3 \beforeeledchapter command	311
XXXII.4 Auxiliary commands	312
XXXII.5 Patching standard commands	313
XXXII.6 Main code of \eledxxx commands	317
XXXII.7 Macros written in the auxiliary file	320
XXXIII Page breaking or no page breaking depending of specific lines	322
XXXIV Long verse: prevents being separated by a page break	324
XXXV Tools for hyperref package	324
XXXVI Compatibility with eledmac	325
Appendix A Things to do when changing versions	328
Appendix A.1 Migrating from edmac to ledmac	328
Appendix A.2 Migration from ledmac to eledmac	329
Appendix A.3 Migration to eledmac 1.5.1	330
Appendix A.4 Migration to eledmac 1.12.0	330
Appendix A.5 Migration to eledmac 17.1	331
Appendix A.6 Migration to eledmac 1.21.0	331
Appendix A.6.1 \Xledsetnormalparstuffand \ledsetnormalparstuffX	331
Appendix A.6.2 Endnotes	331
Appendix A.7 Migration to eledmac 1.22.0	331
Appendix A.8 Migration to eledmac 1.23.0	331
Appendix A.9 Migration from eledmac to reledmac	332
Appendix A.9.1 Risk of 'no room for a new'	332
Appendix A.9.2 Multiple indices with memoir	332
Appendix A.9.3 Deprecated commands and options	332
Appendix A.9.4 \renewcommandreplaced by command	333
Appendix A.9.5 Commands the names of which have been changed . . .	333
Appendix A.9.6 Endnotes	335
Appendix A.9.7 Z Series	335
Appendix A.9.8 Internal commands	335
Appendix A.10 Migration to reledmac 2.1.0	335
Appendix A.11 Migration to reledmac 2.1.3	335
Appendix A.12 Migration to reledmac 2.3.0	335
Appendix A.13 Migration to reledmac 2.4.0	336
Appendix A.14 Migration to reledmac 2.5.0	336
Appendix A.15 Migration to reledmac 2.7.0	336
Appendix A.16 Migration to reledmac 2.7.2	336
Appendix A.17 Migration to reledmac 2.8.0	336
References	337

Index	337
Change History	383

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

As of 1994, we were pleased to be able to say that edmac was being used for the real-life book production of several interesting editions, such as the Latin texts of Euclid’s *Elements*,² an edition of the letters of Nicolaus Copernicus,³ Simon Bredon’s *Arithmetica*,⁴ a Latin translation by Plato of Tivoli of an Arabic astrolabe text,⁵ a Latin translation of part II of the Arabic *Algebra* by Abū Kāmil Shujā’ b. Aslam,⁶ the Latin *Rithmachia* of Werinher

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).

von Tegernsee,⁷ a middle-Dutch romance epic on the Crusades,⁸ a seventeenth-century Hungarian politico-philosophical tract,⁹ an anonymous Latin compilation from Hungary entitled *Sermones Compilati in Studio Generali Quinqueecclesiensi in Regno Ungarie*,¹⁰ the collected letters and papers of Leibniz,¹¹ Theodosius's *Spherics*, the German *Algorismus* of Sacrobosco, the Sanskrit text of the *Kāśikāvṛtti* of Vāmana and Jayāditya,¹² and the English texts of Thomas Middleton's collected works.

1.2.2 ledmac

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

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

In March 2003 Peter Wilson started an attempt to port `edmac` from 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. 32). 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. 329.

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).

⁷Menso Folkerts, ‘Die *Rithmachia* 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.

A new name was selected: `reledmac` (extended renewed `eledmac`). To migrate from `eledmac` to `reledmac`, please read Appendix A.9 p. 332.

1.3 List of works edited with (r)(e)ledmac

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.

2 How the package works

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 passes through \LaTeX to be sure that everything is correctly placed. 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.

3 Compatibility warning

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*.

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 7.13.5 p. 42.

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.

eledmac-compat help to migrate from eledmac to reledmac (see Appendix A.9.5 p. 333).

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. 61).

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. 55).

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. 60).

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. 48). 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 an optional argument in brackets. Its content will be printed before the beginning of `\pstart` / after the end of `\pend` instead of the argument of `\AtEveryPstart` / `\AtEveryPend`. If you need to start a `\pstart` with

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

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. 44) or `reledpar` (see 19.1.8 p. 65).

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

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

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

`\AtEveryPend`

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`.

`\numberpstarttrue`

`\numberpstartfalse`

`\thepstart`

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 `X \TeX` , you must switch text direction *before* the `\pstart` command.

5.2.7 Memory limits

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

`\pausenumbering`

`\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

\resumenummering
\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\resumenummering}
```

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

5.3 Lineation commands

5.3.1 Disabling lineation

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

5.3.2 Setting lineation start and step

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

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

```
\def\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

```
\def\linenumberlist{}
```

the standard numbering sequence is applied. The standard sequence is that specified by

the combination of the `firstlinenum`, `linenumincrement`, `firstsublinenum` and `linenumincrement` counter values.

5.3.3 Setting lineation reset

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

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

5.3.4 Setting line number margin

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

`\linenummargin{left}`

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

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

5.3.5 Other settings

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

5.4 Changing the line numbers

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

5.4.1 Sublineation

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

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

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

You can change the separator between line number and subline number or using `\Xsublinesep` without any option argument (7.2.8 p. 35 or using `\Xsublinesepside`. But in the second case, it will change the separator only for line number in side, not for the footnotes.

5.4.2 Locking lineation

`\startlock` The `\startlock` command, used in running text, locks the line number at its current value, until you insert `\endlock`. It can tell for itself whether you are in a patch of line or sub-line numbering. One use for line-number locking is in printing poetry: there the line numbers should be those of verse lines rather than of printed lines, even when a verse line requires several printed lines. But in this case you may use the `\stanza` mechanism, see 9 p. 44.

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

5.4.3 Setting and changing line number

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

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

5.4.4 Line number style

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

`\sublinenumberstyle`

Alph Uppercase letters (A ... Z).

alph Lowercase letters (a ... z).

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

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

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

Note that with the **Alph** or **alph** styles, ‘numbers’ must be between 1 and 26 inclusive.

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

5.4.5 Skipping and hiding number

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

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

5.4.6 Execute 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. 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 \edtext{Smith}{		2	Tuesday.
\Afootnote{Jones C, D.}}			
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 `⟨lemma⟩` may contain further `\edtext` commands. Nesting makes it possible to print an explanatory note on a long passage together with notes on variants for individual words within the passage. For example:

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

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

6.2.2 Footnotes

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

`\Afootnote` Five separate series of the footnotes are maintained; each macro takes one argument like `\Afootnote{⟨text⟩}`. 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.

`\Bfootnote` These are the main macros that you will use to construct the critical apparatus of your text.

`\Cfootnote`

`\Dfootnote`

`\Efootnote`

If you need more series of critical notes, please look at 6.6.1 p. 31.

An optional argument can be added before the text of the footnote. Its value is a comma-separated list of options. The available options are:

- `fulllines` to disable `\Xtwolines` and `\Xmorethantwolines` features for this note (cf. 7.2.4 p. 33).

- `nonum` disables line numbering for this note. A horizontal blank space is added instead. You can use `\Xinplaceoflemmaseparator` to set it (7.5.1 p. 37).
- `nosep` to disable the lemma separator for this note.
- `linangesep=<c>` to change to `<c>` the separator between start line and end line for this particular note.

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

6.2.3 Endnotes

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

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

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

`\Dendnote`

`\Eendnote`

`\Aendnote[<option>]{<text>}`.

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

- `fulllines` to disable `\Xendtwolines` and `\Xendmoreethantwolines` features for this particular note (cf. 7.2.4 p. 33).
- `nonum` to disable line number for this particular note.
- `nosep` to disable the lemma separator for this particular note. A horizontal blank space is added instead. You can use `\Xendinplaceoflemmaseparator` to set it (7.5.2 p. 38).
- `linangesep=<c>` to change to `<c>` the separator between start line and end line for this particular note.

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

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

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



```
\doendnotesbysection{B}
```

Note that by default inside endnotes no separator is used between the lemma and the content. However you can use the `\Xendlemmaseparator` macro to define one (7.5.2 p. 38).

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

6.2.4 Paragraph in critical apparatus

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

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

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

6.2.5 Change lemma and line number

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

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

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

1 Smith] Jones C, D.

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

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

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

However, you can retain the value computed by `reledmac` for any number by simply omitting it; and you can omit a sequence of vertical bars at the end of the argument. 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 $\langle lemma \rangle$ 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. 48) 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:¹⁴

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

6.3 Disambiguation of identical words in the apparatus

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

6.3.1 Basic use

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

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

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

¹⁴We use `\newcommand` and `\newcommandx` instead of classical `\let` command because the `edtabular` environments have to modify the notes definition, and we need to use the newest definition of notes. Read the handbook of `xargs` to know more about `\newcommandx`.

6.3.2 Notes about input encoding with UTF-8 processor

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

For instance, in 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 \XeTeX , add this line in your preamble: `\XeTeXinputnormalization 1`.
- If you use \LuaTeX , use the `uninormalize` package of Michal Hoftich¹⁵ with the `buffer` option set to `true`.

With these tools, \XeTeX / \LuaTeX will dynamically normalize unicode input when reading the file. Consequently, you will have no problems with the `\sameword` command.

6.3.3 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`:

¹⁵<https://github.com/michal-h21/uninormalize>.

- 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`.

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 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.

6.3.4 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}%
}
```

6.4 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. 32). However, the following *additional* commands are available.

`\setmsdataseries` The series used by default for the apparatus of manuscripts is series A. However, you can change it with `\setmsdataseries{<series>}`.

`\setmsdatalabel` As the apparatus of manuscripts consists of regular critical footnotes, a lemma is associated to them. By default, it is “Ms.”. You can change it using `\setmsdatalabel{<txt>}`.

6.5 Familiar notes

6.5.1 Basic use

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

`\footnoteB`

`\footnoteC`

`\footnoteD`

`\footnoteE`

6.5.2 Customizing mark

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

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

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

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

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

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

There are similar command triples for the other series.

6.5.3 Separator for multiple footnotes

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

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

6.6 Changing series

6.6.1 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}`.

6.6.2 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}
```

6.6.3 Series order

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

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

6.7 Position of critical and familiar footnotes

`\fnpos` `\mpfnpos` 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, 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}
```

7 Critical apparatus appearance

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 unit¹⁶, 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[s]{a}` to change the arrangement of the *s* series of critical footnotes and `\arrangementX[s]{a}` to change the arrangement of the *s* series of familiar footnotes.

The value of *a* can be one of the following

- `paragraph` formats all the footnotes of a series as a single paragraph. If you use this arrangement, you are strongly encouraged to read 19.1.6 p. 64.
- `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.

¹⁶Like `em` which is the width of an ‘m’ in a given font.

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

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

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

7.2 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.

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

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

7.2.2 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.

7.2.3 Separator for line range

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

7.2.4 Abbreviate line range

`\Xtwolines` If a lemma is printed on two subsequent lines, `reledmac` will print the first and the last

`\Xmorethantwolines`

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.

`\Xtwolinesonlyinsamepage`

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

You can disable `\Xtwolines` and related for a specific note by using the ‘[fulllines]’ argument in the note macro cf. 6.2.2 p. 23.

`\Xendtwolines`
`\Xendmorethantwolines`
`\Xendtwolinesbutnotmore`

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

7.2.5 Disable line number

`\Xnonumber`
`\Xendnonumber`

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

7.2.6 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.

7.2.7 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.

7.2.8 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{.}`.

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

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

7.2.10 Space around line symbol

`\XbeforeSYMlinenum` With `\XbeforeSYMlinenum[⟨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`.

`\XendbeforeSYMlinenum` `\XendbeforeSYMlinenum` and `\Xendaftersymlinenum` are the equivalents of `\XbeforeSYMlinenum` and `\Xaftersymlinenum` for the endnotes.

7.2.11 Space in place of number

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

7.2.12 Boxing line number and line symbol

<code>\Xboxlinenum</code>	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 <code>\Xboxlinenum[⟨s⟩]{⟨l⟩}</code> to do that. To subsequently disable this feature, use <code>\Xboxlinenum</code> with length equal to 0 pt. One use of this feature is to print line number in a column, and the note in an other column: <pre>\Xhangindent{1em} \Xafternumber{0em} \Xboxlinenum{1em}</pre>
---------------------------	---

<code>\Xboxsymlinenum</code>	<code>\Xboxsymlinenum[⟨s⟩]{⟨l⟩}</code> is the same as <code>\Xboxlinenum</code> but for the line number symbol.
<code>\Xendboxsymlinenum</code>	<code>\Xendboxsymlinenum[⟨s⟩]{⟨l⟩}</code> is the same as <code>\Xboxsymlinenum</code> but for endnotes.
<code>\Xboxlinenumalign</code>	If you put line number in box, it will be aligned left inside the box. However, you can change it using <code>\Xboxlinenumalign[⟨s⟩]{⟨text⟩}</code> where <code>⟨text⟩</code> can be the following: L to align left (default value); R to align right; C to center.

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

- `\Xboxstartlinenum[⟨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.
```

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

7.3 For endnotes

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

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

`\Xendlineprefixsingle` `\Xendlineprefixsingle[\langle s \rangle]{\langle text \rangle}` defines the text before the line number in endnotes, when there is only one line. Default value is empty.

`\Xendlineprefixmore` `\Xendlineprefixmore[\langle s \rangle]{\langle text \rangle}` 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`.

7.4 Arbitrary code around line number

`\Xendbhooklinenumber` `\Xendbhooklinenumber[\langle s \rangle]{\langle code \rangle}` is used to execute code before line number in endnotes. The code is executed before the `\Xendbeforelinenumber` space and before the `\Xendnotenumfont` font setting.

`\Xendahooklinenumber` `\Xendahooklinenumber[\langle s \rangle]{\langle code \rangle}` is used to execute code after line number in endnotes. The code is executed after the `\Xendafternumber` space.

`\Xendbhookinplaceofnumber` `\Xendbhookinplaceofnumber[\langle s \rangle]{\langle code \rangle}` is used to execute code before space or symbol which replace line number in endnotes. The code is executed before the `\Xendbeforesymlinenum` space and before the `\Xendnotenumfont` font setting.

`\Xendahookinplaceofnumber` `\Xendahookinplaceofnumber[\langle s \rangle]{\langle code \rangle}` is used to execute code after space or symbol which replace line number in endnotes. The code is executed after the `\Xendaftersymlinenum` space.

7.5 Separator between the lemma and the note

7.5.1 For footnotes

`\Xlemmaseparator` By default, in a footnote, the separator between the lemma and the note is a right bracket (`\rbracket`)¹⁷. You can use `\Xlemmaseparator[\langle s \rangle]{\langle Xlemmaseparator \rangle}` to change it. The optional argument can be used to specify the series in which it is used. Note that there is a non-breakable space between the lemma and the separator, but a **breakable** space between the separator and the following text.

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

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

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

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

¹⁷For polyglossia, when the lemma is RTL, the bracket automatically switches to a left bracket.

7.5.2 For endnotes

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

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

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

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

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

7.6 Font style

7.6.1 For line number

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

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

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

7.6.2 For the lemma

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

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

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

`\Xendlemmafont`

`\Xlemmafont{\bfseries}`

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

7.6.3 For all notes

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

7.7 Wrapping notes

7.7.1 Wrapping lemmas

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

7.7.2 Wrapping contents

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

7.8 Indent of notes content

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

7.9 Arbitrary code 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>\bhooknoteX</code> <code>\Xendbhooknote</code>	<code>\Xbhooknote[⟨s⟩]{⟨code⟩}</code> is to be used at the beginning of the critical footnotes. <code>\bhooknoteX[⟨s⟩]{⟨code⟩}</code> is to be used at the beginning of the familiar footnotes. <code>\Xendbhooknote[⟨s⟩]{⟨code⟩}</code> is to be used at the beginning of the endnotes.
---	--

7.10 Arbitrary code before inserting note

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

They allow one to add any arbitrary code just before the footnotes are added in the list of footnotes. The main use is to insert text direction code. For example, if you edit right-to-left text with `bid`, but want your critical footnote be left-to-right, use `\Xbeforeinserting\LTR`. You should also use `\Xwraplemma` to ensure your lemmas are right-to-left in a left-to-right paragraph (7.7.1 p. 39)).

Note that the changes are local to the footnote.

7.11 Options for footnotes in columns

7.11.1 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[⟨s⟩]{⟨code⟩}` for critical footnotes, and `\colalignX[⟨s⟩]{⟨code⟩}` for familiar footnotes.

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

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

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

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

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

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

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

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

7.11.2 Size of the columns

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

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

7.12 Options for paragraphed footnotes

7.12.1 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> .
<code>\parafootsepX</code>	

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

7.12.2 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.

7.13 Options for block of notes

7.13.1 Text before notes

<code>\Xtxtbeforenotes</code>	You can add text before critical notes with <code>\Xtxtbeforenotes[⟨s⟩]{⟨text⟩}</code> .
-------------------------------	--

7.13.2 Code before notes

<code>\Xbhookgroup</code>	While <code>\Xtxtbeforenotes</code> is for typesetting code before notes, <code>\Xbhookgroup</code> and <code>\bhookgroupX</code> (respectively for critical and familiar) are for executing code before a groups of notes, between the rules and the printing of the notes.
<code>\bhookgroupX</code>	

7.13.3 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.

`\preXnotes` 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 `\preXnotes{⟨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.

7.13.4 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.

7.13.5 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*.

7.13.6 Width

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

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

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

Note that changes the width of the block of notes. If you want to change the width of each column when typesetting notes in columns, use `\Xhsizetwocol`, `\Xhsizethreecol`, `\hsizetwocolX`, `\hsizethreecolX`, see 7.11.2 p. 41.

7.14 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.

7.15 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.

For those who are setting up for a large job, here is a list of the complete set of *reledmac* macros relating to fonts that are intended for manipulation by the user: `\endashchar`, `\fullstop`, `\numlabfont`, and `\rbracket`.

`\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.

`\&`

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. 46.

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. 330.

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` 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. 62 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.

Use `\AtEveryStanza{<arg>}` to automatically add content at the beginning of stanza.

- `&` 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.

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

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

Use `\AtEveryStopStanza` to automatically add content at the end of stanzas.

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` (7.2.7 p. 35).

`\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 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`.

9.11 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. 46).

10 Grouping

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

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

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

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

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

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

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

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

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

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

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

11 Cross referencing

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

11.1 Basic use

`\edlabel` First you place a label in the text using the command `\edlabel{\langle lab \rangle}`. `\langle lab \rangle` can be almost anything you like, including letters, numbers, punctuation, or a combination—anything but spaces; you might type `\edlabel{toves-3}`, for example.¹⁸

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

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

`\pstartref`

¹⁸More precisely, you should stick to characters in the \TeX categories of “letter” and “other”.

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

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

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

11.2 Cross-referencing to a critical note

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

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

If you want to refer to starting and ending lines, you should use `\appref` and related tools (11.6.2 p. 51).

11.3 Cross-referencing which return a number in any case

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

Where `#1` stands for the reference.

However, there are situations in which you will want `reledmac` to return a number without displaying any warning messages about undefined labels or the like: if you want to use the reference in a context where `ℒTEX` is looking for a number, such a warning will lead to a complaint that the number is missing. This is the case for references used within the argument to `\linenum`, for example (see 6.2.5 p. 25).

For this situation, four variants of the reference commands, with the `x` prefix, are supplied: `\xpageref`, `\xlineref`, `\xsublineref` and `\xpstartref`. 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 macros `\xxref` and `\edmakelabel` let you manipulate numbers and labels in ways which you may find helpful in tricky situations.

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 calls `\linenum` (q.v., 6.2.5 p. 25 above) and sets the beginning page, line and subline numbers to those of the place where `\edlabel{mouse}` was placed, and the ending numbers to those where `\edlabel{elephant}` occurs.

11.4 Not automatic cross-referencing

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

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

11.5 Normal L^AT_EX cross-referencing

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

11.6 References to start and end lines

11.6.1 Reference to main text lines

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

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

`\edlabelE` Use `\edlabelE{<label>}` to mark the end the end line of the passage. These two commands just create 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 end-notes.

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

11.6.2 References to lines that are commented on in the apparatus

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

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

If you use `\linenum` (6.2.5 p. 25) to refer to these labels, `reledmac` will use your line settings to refer to the passage.

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

11.6.3 Settings

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

For example, you may use:

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

Note that if you have not used `\setapprefprefixmore` is empty, argument of `\setapprefprefixsingle` will be used in any case.

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

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

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

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

Also note that `\setSerefonlypageprefixsingle` is only a shortcut for `\XendbeforepagenumberSerefonly` (see 11.6.3 p. 51). So if you use `\Xendbeforepagenumber` without any optional argument, it will override this setting.

Linked to setting of critical endnotes and footnotes 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`

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`

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. 23).

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. 24).

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
`\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.

12.2 Setting

12.2.1 Width

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

12.2.2 Vertical position

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

12.2.3 Distance to the main text

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

12.2.4 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 & line number 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`.

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\edlabel{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}}
```

13.3 Separator between page and line numbers

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

– is the default separator used by the MAKEINDEX program.

Consequently, if you want to use an other `\pagelinesep`, you have to configure your `.ist` index style file. For example if you use `:` as separator¹⁹.

```
page_compositor ":"
delim_r ":"
```

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-linenummer`

An example of such a file is provided in the “examples” folder. Read the `xindy` handbook to learn how to use it.²⁰

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

¹⁹For further detail, you can read <http://tex.stackexchange.com/a/32783/7712>.

²⁰Or, 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.²¹.

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` use, in standard, 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. 55).

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>]
```

²¹These are the recommended lines to provide the best possible compatibility between `hyperref` and `xindy`, even without using `reledmac`.

15 Tabular material

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

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

```

edarrayl      \begin{edtabularc}
edarrayc      1 & 2 & 3 \\
edarrayr      a & bb & ccc \\
edtabularl    AAA & BB & C
edtabularc    \end{edtabularc}
edtabularr

```

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

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

For example:

```

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

```

produces the following parallel pair of verses.

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

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

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

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

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

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

[illegible]

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

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

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

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

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

For example:

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

	$\begin{array}{cccc} A & 1 & 2 & 3 \\ B & 1 & 3 & 6 \\ C & 1 & 4 & 8 \\ D & 1 & 5 & 0 \end{array}$	<p>After</p>
Before		

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

`\edvertdots`

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

```

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. 17):

```

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

```

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

16.2 Sectioning commands with line numbering and critical notes

You have to use the following commands:

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

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

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

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

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

16.3 Optimization

`\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.

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 7.13.5 p. 42 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. 61 in order to know which are the available options.

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

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

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

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

19.1.4 Marginal notes

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

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

19.1.5 Paragraph shape

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

`\ballast`

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

19.1.6 Paragraphed footnotes

The restriction on explicit line-breaking in paragraphed footnotes, mentioned on 7.1 p. 32, and described in more detail on XII.6.3 p. 159, 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` (7.13.5 p. 42), 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 this²² you will find \TeX whinging ‘Missing { inserted’, and then things start to fall apart. The trick in this case is to specify either:

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

or

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

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

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

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

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

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

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

19.1.8 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 .

²²Reported 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 XII) and output routine (Section XXII) finish the main part of the processing; cross-referencing (Section XXIII) and endnotes (Section XIX) 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}[2016/06/03 v2.11.0a typeset critical editions]%
4 %

```

II.3 Package options

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

`\ifnocritical@`
`\if@noeled@sec` `\RequirePackage{xkeyval}`
`\ifnoend@` %

`\ifnofamiliar@`
`\ifnoledgroup@` The `parledgroup` option is for `reledpar`. However, it has consequence on `reledmac` internal command. So we need to define the boolean now.

`\ifparapparatus@`
`\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@`
`\newif\if@noeled@sec%`
`\DeclareOptionX{noeledsec}{\@noeled@sectrue}`

`\newif\ifnocritical@%`
`\DeclareOptionX{nocritical}{\nocritical@true}%`

`\newif\ifnofamiliar@%`
`\DeclareOptionX{nofamiliar}{\nofamiliar@true}%`

`\newif\ifnoledgroup@%`
`\DeclareOptionX{noledgroup}{\noledgroup@true}%`

`\newif\ifnoend@%`
`\DeclareOptionX{noend}{%`
`\let\l@dend@open\@gobble%`
`\let\l@dend@close\relax%`
`\global\let\l@dend@stuff=\relax%`
`\noend@true%`

`}%`
`\newif\ifnoquotation@`
`\DeclareOptionX{noquotation}{\noquotation@true}`

`\newif\ifledfinal`
`\DeclareOptionX{final}{\ledfinaltrue}`
`\DeclareOptionX{draft}{\ledfinalfalse}`
`\ExecuteOptionsX{final}`

```

39 \newif\ifparapparatus@
40 \DeclareOptionX{parapparatus}{\parapparatus@true}
41
42 \newif\iflednopbinverse
43 \DeclareOptionX{nopbinverse}{\lednopbinversetrue}
44
45 \newif\ifwidthliketwocolumns%
46 \DeclareOptionX{widthliketwocolumns}{\widthliketwocolumnstrue}%
47
48 \newif\ifcontinuousnumberingwithcolumns
49 \DeclareOptionX{continuousnumberingwithcolumns}{\
50 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 %

```

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.

```

74 \ProcessOptionsX*\relax
75
76 %

```

II.4 Loading packages

Loading package `xargs` to declare commands with optional arguments. `Etoolbox` is also used to make code clearer - for example, in dynamic command names (which can re-

place `\csname` etc.). Use `suffix` to declare commands with a starred version, `xstring` to work with strings, `ifluatex` and `ifxetex` to test if Lua_T_EX or X_Y_T_EX is running, and `ragged2e` to manage ragged justification for paragraphed notes.

```

77 \RequirePackage{xargs}
78 \RequirePackage{etoolbox}
79 \ifl@t@r\fmtversion{2015/10/01}
80 {}%
81 {\RequirePackage{etex}%
82 \csname reserveinserts\endcsname{32}%
83 }
84 \RequirePackage{suffix}
85 \RequirePackage{xstring}
86 \RequirePackage{ifluatex}
87 \RequirePackage{ragged2e}
88 \RequirePackage{ifxetex}%
89 %

```

II.5 Compatibility with Lua_T_EX

Here, we enable some primitives for Lua_T_EX.

```

90 \ifx\directlua\undefined\else%
91 \directlua{tex.enableprimitives("",{"texdir","pardir","bodydir"})}
92 \fi
93 %

```

II.6 Boolean flags

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

```

94 \newif\ifl@dmemoir
95 \@ifclassloaded{memoir}{\l@dmemoirtrue}{\l@dmemoirfalse}
96
97 %

```

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

```

98 \newif\if@ledgroup%
99 %

```

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

```

100 \newif\ifl@imakeidx
101 \@ifpackageloaded{imakeidx}{\l@imakeidxtrue}{}%False is the default value
102 %

```

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

```

103 \newif\ifl@indextools%
104 \@ifpackageloaded{indextools}{%
105   \l@indextoolstrue%
106   \l@imakeidxtrue%
107   \let\imki@wrindexentry\indtl@wrindexentry%
108 }{}%
109 %

```

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`.

`\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.

```

110 \ifdef{\if@RTL}{\newif\if@RTL}
111 %

```

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

```

112 \newif\if@firstlineofpage%
113 \newif\if@firstlineofpageR%
114 %

```

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.

```

115 \newcommand{\reledmac@warning}[1]{\PackageWarning{reledmac}{#1}}
116 %

```

`\reledmac@error` Write an error message.

```

117 \newcommand{\reledmac@error}[2]{\PackageError{reledmac}{#1}{#2}}
118 %

```

`\led@err@NumberingStarted`¹⁹ `\newcommand*\led@err@NumberingStarted}{%`

`\led@err@NumberingNotStarted`²⁰ `\reledmac@error{Numbering has already been started}{\@ehc}}`

`\led@err@NumberingShouldHaveStarted`²¹ `\newcommand*\led@err@NumberingNotStarted}{%`

```

122   \reledmac@error{Numbering was not started}{\@ehc}}

```

```

123 \newcommand*\led@err@NumberingShouldHaveStarted}{%

```

```

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

```

```

125 %

```

```

\led@err@edtextoutsidepstart26 \newcommand*{\led@err@edtextoutsidepstart}{%
127 \reledmac@error{\string\edtext\space outside numbered paragraph (\pstart\
ldots\pend)}{\@ehc}}%
128 %

```

```

\led@mess@NotesChanged29 \newcommand*{\led@mess@NotesChanged}{%
130 \typeout{\reledmac reminder: }%
131 \typeout{ The number of the footnotes in this section
132 has changed since the last run.}%
133 \typeout{ You will need to run LaTeX two more times
134 before the footnote placement}%
135 \typeout{ and line numbering in this section are
136 correct.}}
137 %

```

```

\led@mess@SectionContinued38 \newcommand*{\led@mess@SectionContinued}[1]{%
139 \message{Section #1 (continuing the previous section)}}
140 %

```

```

\led@err@LineationInNumbered41 \newcommand*{\led@err@LineationInNumbered}{%
142 \reledmac@error{You can't use \string\lineation\space within
143 a numbered section}{\@ehc}}
144 %

```

```

\led@warn@BadLineation45 \newcommand*{\led@warn@BadLineation}{%
\led@warn@BadLinenummargin46 \reledmac@warning{Bad \string\lineation\space argument}}
\led@warn@BadLockdisp47 \newcommand*{\led@warn@BadLinenummargin}{%
\led@warn@BadSublockdisp48 \reledmac@warning{Bad \string\linenummargin\space argument}}
149 \newcommand*{\led@warn@BadLockdisp}{%
150 \reledmac@warning{Bad \string\lockdisp\space argument}}
151 \newcommand*{\led@warn@BadSublockdisp}{%
152 \reledmac@warning{Bad \string\sublockdisp\space argument}}
153 %

```

```

\led@warn@NoLineFile54 \newcommand*{\led@warn@NoLineFile}[1]{%
155 \reledmac@warning{Can't find line-list file #1}}
156 %

```

```

\led@warn@LineFileObsolete57 \newcommand*{\led@warn@Obsolete}[1]{%
158 \reledmac@warning{Line-list file #1 was obsolete. We have not read it.
Please run LaTeX again.}}
159 %

```

```

\led@warn@BadAdvancelineSubline60 \newcommand*{\led@warn@BadAdvancelineSubline}{%
\led@warn@BadAdvancelineLine61   \reledmac@warning{\string\advanceline\space produced a sub-line
162                                   number less than zero.}}
163 \newcommand*{\led@warn@BadAdvancelineLine}{%
164   \reledmac@warning{\string\advanceline\space produced a line
165                                   number less than zero.}}
166 %

\led@warn@BadSetline67 \newcommand*{\led@warn@BadSetline}{%
\led@warn@BadSetlinenum68   \reledmac@warning{Bad \string\setline\space argument}}
169 \newcommand*{\led@warn@BadSetlinenum}{%
170   \reledmac@warning{Bad \string\setlinenum\space argument}}
171 %

\led@err@PstartNotNumbered72 \newcommand*{\led@err@PstartNotNumbered}{%
\led@err@PstartInPstart73   \reledmac@error{\string\pstart\space must be used within a
\led@err@PendNotNumbered74   numbered section %
\led@err@PendNoPstart75     (\string\...beginnumbering\string\endnumbering)}{\@ehc}}%
\led@err@AutoparNotNumbered76 \newcommand*{\led@err@PstartInPstart}{%
\led@err@NumberingWithoutPstart77 \reledmac@error{\string\pstart\space encountered while another
178   \string\pstart\space was in effect}{\@ehc}}
179 \newcommand*{\led@err@PendNotNumbered}{%
180   \reledmac@error{\string\pend\space must be used within a
181   numbered section}{\@ehc}}
182 \newcommand*{\led@err@PendNoPstart}{%
183   \reledmac@error{\string\pend\space must follow a \string\pstart}{\@ehc}}
184 \newcommand*{\led@err@AutoparNotNumbered}{%
185   \reledmac@error{\string\autopar\space must be used within a
186   numbered section}{\@ehc}}
187 \newcommand*{\led@err@NumberingWithoutPstart}{%
188   \reledmac@error{\string\beginnumbering...\string\endnumbering\space
without \string\pstart}{\@ehc}}%
189 %

\led@warn@BadAction90 \newcommand*{\led@warn@BadAction}{%
191   \reledmac@warning{Bad action code, value \next@action.}}
192 %

\led@warn@DuplicateLabel93 \newcommand*{\led@warn@DuplicateLabel}[1]{%
\led@warn@AppLabelOutEdtext94   \reledmac@warning{Duplicate definition of label `#1'\@gobble}%
\led@warn@RefUndefined95   \@latex@warning@no@line{Label `#1' multiply defined}%
\led@warn@RefUndefined96   }%
197 \newcommand*{\led@warn@AppLabelOutEdtext}[1]{%
198   \reledmac@warning{\string\applabel\space outside of \string\edtext\space
`#1' on page \the\pageno.}}%

```



```

199 \newcommand*\led@warn@RefUndefined}[1]{%
200   \G@refundefinedtrue%
201   \reledmac@warning{Reference `#1' on page \the\pageno\space undefined.%
202     Using `000'.}%
203   \@latex@warning{Reference `#1' undefined\on@line}%
204 }%
205 \newcommand*\led@warn@pairRefUndefined}[1]{%
206   \G@refundefinedtrue%
207   \reledmac@warning{Reference `#1:start' and/or `#1:end' on page \the\
208     pageno\space undefined.
209     Using `??'.}%
210   \@latex@warning{Reference `#1:start' and/or `#1:end' undefined\on@line}%
211 }
212 %

```

```

\led@warn@NoMarginpars12 \newcommand*\led@warn@NoMarginpars}{%
213   \reledmac@warning{You can't use \string\marginpar\space in numbered text
214   }}
215 %

```

```

\led@warn@BadSidenotemargin15 \newcommand*\led@warn@BadSidenotemargin}{%
216   \reledmac@warning{Bad \string\sidenotemmargin\space argument}}
217 %

```

```

\led@warn@NoIndexFile18 \newcommand*\led@warn@NoIndexFile}[1]{%
219   \reledmac@warning{Undefined index file #1}}
220 %

```

```

\led@warn@SeriesStillExist21 \newcommand*\led@warn@SeriesStillExist}[1]{%
222   \reledmac@warning{Series #1 is still existing !}%
223 }%
224 %

```

```

\led@err@ManySidenotes25 \newcommand*\led@err@ManySidenotes}{%
\led@err@ManyLeftnotes26   \ifledRcol{%
\led@err@ManyRightnotes27     \reledmac@warning{\itemcount@\space sidenotes on line \the\line@numR\
space p. \the\page@numR}%
228   \else%
229     \reledmac@warning{\itemcount@\space sidenotes on line \the\line@num\
space p. \the\page@num}%
230   \fi%
231 }%
232 \newcommand*\led@err@ManyLeftnotes}{%
233   \ifledRcol{%

```

```

234 \reledmac@warning{\itemcount@\space leftnotes on line \the\line@numR\
space p. \the\page@numR}%
235 \else%
236 \reledmac@warning{\itemcount@\space leftnotes on line \the\line@num\
space p. \the\page@num}%
237 \fi%
238 }%
239 \newcommand{\led@err@ManyRightnotes}{%
240 \ifledRcol%
241 \reledmac@warning{\itemcount@\space rightnotes on line \the\line@numR\
space p. \the\page@numR}%
242 \else%
243 \reledmac@warning{\itemcount@\space rightnotes on line \the\line@num\
space p. \the\page@num}%
244 \fi%
245 }%
246 %

```

```

\led@err@TooManyColumns47 \newcommand*\led@err@TooManyColumns{%
\led@err@UnequalColumns48 \reledmac@error{Too many columns}{\@ehc}}
\led@err@LowStartColumn49 \newcommand*\led@err@UnequalColumns{%
\led@err@HighEndColumn50 \reledmac@error{Number of columns is not equal to the number
\led@err@ReverseColumns51 in the previous row (or \protect\ \space forgotten?)}{\@ehc}}
252 \newcommand*\led@err@LowStartColumn{%
253 \reledmac@error{Start column is too low}{\@ehc}}
254 \newcommand*\led@err@HighEndColumn{%
255 \reledmac@error{End column is too high}{\@ehc}}
256 \newcommand*\led@err@ReverseColumns{%
257 \reledmac@error{Start column is greater than end column}{\@ehc}}
258 %

```

```

\led@err@EdtextWithoutFootnote59 \newcommand{\led@err@EdtextWithoutFootnote}{%
260 \reledmac@error{edtext without Xfootnote. Check syntax.}{\@ehc}%
261 }%
262 %

```

```

\led@err@FootnoteWithoutEdtext63 \newcommand{\led@err@FootnoteWithoutEdtext}{%
264 \reledmac@error{Xfootnote without edtext. Check syntax.}{\@ehc}%
265 }%
266 %

```

```

\led@error@ImakeidxAfterEledmac67 \newcommand{\led@error@ImakeidxAfterEledmac}{%
268 \reledmac@error{Imakeidx must be loaded before reledmac.}{\@ehc}%
269 }%
270 %

```

```

\newcommand{\led@error@IndextoolsAfterEledmac}{%
\reledmac@error{Indextools must be loaded before reledmac.}{\@ehc}%
}%
%

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

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

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

\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@msdatawithoutstop03 \newcommand{\led@warning@msdatawithoutstop}{%
304 \reledmac@warning{\string\msdata\space without corresponding \string\
stopmsdata}%
305 }%
306 %

```

II.8 Gobbling

Here, we define some commands which gobble their arguments.

```

\@gobblethree07 \providecommand*\@gobblethree}[3]{}
\@gobblefour08 \providecommand*\@gobblefour}[4]{}
\@gobblefive09 \providecommand*\@gobblefive}[5]{}
310 %

```

II.9 Miscellaneous commands

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

```

311 \ifledfinal
312 \newcommand*\showlemma[1]{#1}
313 \else
314 \newcommand*\showlemma[1]{\underline{#1}}
315 \fi
316
317 %

```

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

```

318 \let\linenumberlist=\empty
319
320 %

```

`\@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).

```

321 \newcount\@l@tempcnta \newcount\@l@tempcntb
322 %

```

II.10 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 eledpar manual.

`\ifl@dpaging`

`\ifl@dprintingpages`

`\ifl@dprintingcolumns`

```

323 \newif\ifl@dpairing
324 \newif\ifl@dpaging%
325 \newif\ifl@dprintingpages%
326 \newif\ifl@dprintingcolumns%
327 \newif\ifpst@rtedL
328 \newcount\l@dnumpststartsL
329 %

```

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

`\ifledRcol@`

```

330 \newif\ifledRcol
331 \newif\ifledRcol@
332 %

```

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

```

333 \newif\ifnumberingR
334 %

```

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

```

335 \newif\ifXnote@%
336 %

```

II.11 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`

```

337 \providebool{indtl@innote}%
338 \providebool{indtl@notenumber}%
339 %

```

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. \LaTeX 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.

```

340 \newcount\section@num
341 \section@num=0
342 \let\extensionchars=\empty
343 %

```

`\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.

`\numberingtrue`

`\numberingfalse`

```

344 \newif\ifnumbering
345 %

```

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

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

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

```

346 \newcommand*{\beginnumbering}{%
347   \ifnumbering
348     \led@err@NumberingStarted
349   \endnumbering
350   \fi
351   \global\numberingtrue
352   \global\advance\section@num \@ne
353   \initnumbering@reg
354   \message{Section \the\section@num }%
355   \line@list@stuff{\jobname.\extensionchars\the\section@num}%

```

```

356 \l@dend@stuff
357 \setcounter{pstart}{1}
358 \ifl@dpairing
359 \global\l@dnumpstartsL \z@
360 \global\pst@rtedLfalse
361 %

```

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.

```

362 \else
363 \beginngroup
364 \global\@afterindenttrue%In order to reestablish normal feature if the \
beginngroup was not here
365 \initnumbering@quote
366 \ifwidthliketwocolumns%
367 \cuse{setwidthliketwocolumns@\columns@position}%
368 \cuse{setpositionliketwocolumns@\columns@position}%
369 \fi%
370 \fi
371 \gdef\eled@sections@{ }%
372 \if@noeled@sec\else%
373 \makeatletter\inputIfFileExists{\jobname.eledsec\the\section@num}{ }{\
makeatother%
374 \immediate\openout\eled@sectioning@out=\jobname.eledsec\the\section@num
\relax%
375 \fi%
376 }
377 \newcommand*{\initnumbering@reg}{%
378 \global\pst@rtedLfalse
379 \global\l@dnumpstartsL \z@
380 \global\absline@num \z@
381 \gdef\normal@page@break{}
382 \gdef\l@prev@pb{}
383 \gdef\l@prev@nopb{}
384 \global\line@num \z@
385 \global\subline@num \z@
386 \global\@lock \z@
387 \global\sub@lock \z@
388 \global\sublines@false
389 \global\let\next@page@num=\relax
390 \global\let\sub@change=\relax
391 \resetprevline@
392 \resetprevpage@num
393 \global\stopmsdata@inserted@true%
394 }

```

```
395
396 %
```

\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.

```
397 \def\endnumbering{%
398   \ifnumbering
399     \global\numberingfalse
400     \normal@pars
401     \ifnum\l@dnumpstartsL=0%
402       \led@err@NumberingWithoutPstart%
403     \fi%
404     \ifl@dpairing
405       \global\pst@rtedLfalse
406     \else
407       \ifx\insertlines@list\empty\else
408         \global\noteschanged@true
409       \fi
410       \ifx\line@list\empty\else
411         \global\noteschanged@true
412       \fi
413     \fi
414     \ifnoteschanged@
415       \led@mess@NotesChanged
416     \fi
417   \else
418     \led@err@NumberingNotStarted
419   \fi
420   \autoparfalse
421   \if@noeled@sec\else%
422     \immediate\closeout\eled@sectioning@out%
423   \fi%
424   \ifl@dpairing\else
425     \global\l@dnumpstartsL=\z@%
426   \endgroup
427 \fi
428 }
429 %
```

\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.²³

```
430 \newcommand{\pausenumbering}{%
431   \ifautopar\global\autopar@pausettrue\fi%
432   \endnumbering\global\numberingtrue}
433 %
```

²³Peter Wilson's thanks to Wayne Sullivan, who suggested the idea behind these macros.

The `\resumenumbers` 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 `\resumenumbers` to ensure that `\pausenumbering` was actually invoked.

```

434 \newcommand*{\resumenumbers}{%
435   \ifnumbering
436     \ifautopar@pause\autopar\fi
437     \global\pst@rtedLtrue
438     \global\advance\section@num \@ne
439     \led@mess@SectionContinued{\the\section@num}%
440     \line@list@stuff{\jobname.\extensionchars\the\section@num}%
441     \ledend@stuff
442     \ifl@dpairing\else%
443       \begingroup%
444       \initnumbering@quote%
445       \ifwidthliketwocolumns%
446         \csuse{setwidthliketwocolumns@\columns@position}%
447         \csuse{setpositionliketwocolumns@\columns@position}%
448       \fi%
449     \fi%
450     \ifcontinuousnumberingwithcolumns%
451       \ifdefined\line@numR%
452         \ifnum\line@numR>\line@num%
453           \expandafter\setlinenum\expandafter{\the\line@numR}%
454         \fi%
455         \ifnum\last@page@numR>\last@page@num%
456           \global\last@page@num=\last@page@numR%
457         \fi%
458       \fi%
459     \fi%
460   \else
461     \led@err@NumberingShouldHaveStarted
462     \endnumbering
463     \beginnumbering
464   \fi}
465
466 %
467 %

```

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 `LATEX3` 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.

```
468 \newcommand*{\list@create}[1]{%
469   \global\let#1=\empty%
470 }%
471 %
```

`\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 .

```
472 \newcommand*{\list@clear}[1]{%
473   \global\let#1=\empty%
474 }
475 %
```

`\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`.

```
476 \newtoks\led@toksa \newtoks\led@toksb
477 \global\led@toksa={\}
478 \long\def\xright@appenditem#1\to#2{%
479   \global\led@toksb=\expandafter{#2}%
480   \xdef#2{\the\led@toksb\the\led@toksa\expandafter{#1}}%
481   \global\led@toksb={}}
482 %
```

`\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`.

```
483 \long\def\xleft@appenditem#1\to#2{%
484   \global\led@toksb=\expandafter{#2}%
485   \xdef#2{\the\led@toksa\expandafter{#1}\the\led@toksb}%
486   \global\led@toksb={}}
487 %
```

`\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.

```

488 \def\gl@p#1\to#2{\expandafter\gl@poff#1\gl@poff#1#2}
489 \long\def\gl@poff\#1#2\gl@poff#3#4{\gdef#4{#1}\gdef#3{#2}}
490
491 %

```

V Line counting

V.1 Choosing the system of lineation

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

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

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

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

```

492 \newif\ifbypage@
493 \newif\ifbypstart@
494 %

```

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@R495 \newif\ifbypage@R
\ifbypstart@R496 \newif\ifbypstart@R
497 %

```

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

```

498 \newcommand*{\lineation}[1]{
499 %

```

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

```

500 \ifnumbering
501 \led@err@LineationInNumbered
502 \else
503 %

```

If the argument is page.

```

504 \def\@tempa{#1}\def\@tempb{page}%
505 \ifx\@tempa\@tempb
506   \global\bypage@true
507   \global\bystart@false
508   \unless\ifnocritical@%
509     \Xpstart[] [false]%
510   \fi%
511 %

```

If the argument is pstart.

```

512 \else
513   \def\@tempb{pstart}%
514   \ifx\@tempa\@tempb
515     \global\bypage@false
516     \global\bystart@true
517     \unless\ifnocritical@%
518       \Xpstart%
519     \fi%
520 %

```

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

```

521 \else
522   \def\@tempb{section}%
523   \ifx\@tempa\@tempb
524     \global\bypage@false
525     \global\bystart@false
526     \unless\ifnocritical@%
527       \Xpstart[] [false]%
528     \fi%
529 %

```

In other case, it is an error.

```

530 \else
531   \led@warn@BadLineation
532   \fi
533 \fi
534 \fi
535 \fi}}
536 %

```

V.2 Line number margin

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

```

537 \newcount\line@margin
538

```

```

539 \newcommand*{\linenummargin}[1]{%
540   \l@dgetline@margin{#1}%
541   \ifnum\l@dtempcntb>\m@ne
542     \ifledRcol
543       \global\line@marginR=\l@dtempcntb
544       \led@warn@setting@in@rightside{\linenummargin}%
545     \else
546       \global\line@margin=\l@dtempcntb
547     \fi
548   \fi}}
549
550 \newcommand*{\l@dgetline@margin}[1]{%
551   \def\@tempa{#1}\def\@tempb{left}%
552   \ifx\@tempa\@tempb
553     \l@dtempcntb \z@
554   \else
555     \def\@tempb{right}%
556     \ifx\@tempa\@tempb
557       \l@dtempcntb \@ne
558     \else
559       \def\@tempb{outer}%
560       \ifx\@tempa\@tempb
561         \l@dtempcntb \tw@
562       \else
563         \def\@tempb{inner}%
564         \ifx\@tempa\@tempb
565           \l@dtempcntb \thr@@
566         \else
567           \led@warn@BadLinenummargin
568           \l@dtempcntb \m@ne
569         \fi
570       \fi
571     \fi
572   \fi}
573
574 %

```

V.3 Line number initialization and increment

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

```

575 \newcounter{firstlinenum}
576   \setcounter{firstlinenum}{5}
577 \newcounter{linenumincrement}
578   \setcounter{linenumincrement}{5}

```

579 %

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

```
580 \newcounter{firstsublinenum}
581 \setcounter{firstsublinenum}{5}
582 \newcounter{sublinenumincrement}
583 \setcounter{sublinenumincrement}{5}
584
585 %
```

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

```
\linenumincrement
\firstsublinenum
\sublinenumincrement
586 \newcommand*\firstlinenum[1]{%
587 \ifledRcol%
588 \setcounter{firstlinenumR}{#1}%
589 \led@warn@setting@in@rightside{\firstlinenum}%
590 \else%
591 \setcounter{firstlinenum}{#1}%
592 \fi%
593 }
594
595 \newcommand*\linenumincrement[1]{%
596 \ifledRcol%
597 \setcounter{linenumincrementR}{#1}%
598 \led@warn@setting@in@rightside{\linenumincrement}%
599 \else%
600 \setcounter{linenumincrement}{#1}%
601 \fi%
602 }
603 \newcommand*\firstsublinenum[1]{%
604 \ifledRcol%
605 \setcounter{firstsublinenumR}{#1}%
606 \led@warn@setting@in@rightside{\firstsublinenum}%
607 \else%
608 \setcounter{firstsublinenum}{#1}%
609 \fi%
610 }
611 \newcommand*\sublinenumincrement[1]{%
612 \ifledRcol%
613 \setcounter{sublinenumincrementR}{#1}%
614 \led@warn@setting@in@rightside{\sublinenumincrement}%
615 \else%
616 \setcounter{sublinenumincrement}{#1}%
617 \fi%
618 }
619
620 %
```

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.

```

621 \newcount\lock@disp
622 \newcommand{\lockdisp}[1]{\%
623   \l@dgetlock@disp{#1}%
624   \ifnum\l@dtempcntb>\m@ne
625     \global\lock@disp=\l@dtempcntb
626   \else
627     \led@warn@BadLockdisp
628   \fi}}
629 \newcommand*{\l@dgetlock@disp}[1]{
630   \def\@tempa{#1}\def\@tempb{first}%
631   \ifx\@tempa\@tempb
632     \l@dtempcntb \z@
633   \else
634     \def\@tempb{last}%
635     \ifx\@tempa\@tempb
636       \l@dtempcntb \@ne
637     \else
638       \def\@tempb{all}%
639       \ifx\@tempa\@tempb
640         \l@dtempcntb \tw@
641       \else
642         \l@dtempcntb \m@ne
643       \fi
644     \fi
645   \fi}
646
647 %

```

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

```

648 \newcount\sublock@disp
649 \newcommand{\sublockdisp}[1]{\%
650   \l@dgetlock@disp{#1}%
651   \ifnum\l@dtempcntb>\m@ne
652     \global\sublock@disp=\l@dtempcntb
653   \else
654     \led@warn@BadSublockdisp
655   \fi}}
656
657 %

```

V.5 Line number style

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

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

`\linenumr@p` and `\sublinenumr@p`.

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

`\sublinenumrep`

`\sublinenumr@p`

```

658 \newcommand*{\linenumberstyle}[1]{%
659   \def\linenumrep##1{\@nameuse{##1}}
660 \newcommand*{\sublinenumberstyle}[1]{%
661   \def\sublinenumrep##1{\@nameuse{##1}}
662 %

```

Initialise the number styles to arabic.

```

663 \linenumberstyle{arabic}
664 \let\linenumr@p\linenumrep
665 \sublinenumberstyle{arabic}
666 \let\sublinenumr@p\sublinenumrep
667
668 %

```

V.6 Line number printing

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

`\rightlinenum`

`\linenumsep`

`\numlabfont`

`\ledlinenum`

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

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

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

```

669 \newlength{\linenumsep}
670 \setlength{\linenumsep}{1pc}
671 \newcommand*{\numlabfont}{\normalfont\scriptsize}
672 \newcommand*{\ledlinenum}{%
673   \bgroup%
674   \ifluatex%
675     \texdir TLT%

```



```

676 \fi%
677 \numlabfont\linenumrep{\line@num}%
678 \ifsublines@
679 \ifnum\subline@num>0\relax
680 \unskip%
681 \Xsublinesep@side%
682 \sublinenumrep{\subline@num}%
683 \fi
684 \fi%
685 \egroup%
686 }%
687
688 \newcommand*{\leftlinenum}{%
689 \ledlinenum
690 \kern\linenumsep}
691 \newcommand*{\rightlinenum}{%
692 \kern\linenumsep
693 \ledlinenum}
694
695 %

```

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`.

```

696 \newcount\line@num
697 %

```

\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.

```

698 \newcount\subline@num
699 %

```

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

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

`\sublines@false`

```

700 \newif\ifsublines@
701 %

```

`\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.

```

702 \newcount\absline@num
703 %

```

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`

```

704 \newcount\@lock
705 \newcount\sub@lock
706 %

```

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`

- `\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/cmr/m/n.
```

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

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

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

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

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

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

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

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

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

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

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

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

Here are the commands to create these lists:

```

707 \list@create{\line@list}
708 \list@create{\insertlines@list}
709 \list@create{\actionlines@list}
710 \list@create{\actions@list}
711
712 %

```

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

```

713 \newcount\page@num
714 \newcount\endpage@num
715 \newcount\endline@num

```

```

716 \newcount\endsubline@num
717 %

```

`\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.

```

718 \newif\ifnoteschanged@
719 %

```

`\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@20 \newcommand*\resetprevline@{%
721   \def\do##1{\global\csundef{prevline##1}}%
722   \dolistloop{\@series}%
723 }
724 %

```

`\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@25 \newcommand*\resetprevpage@num{%
726   \def\do##1{%
727     \ifcsdef{prevpage##1@num}{%
728       \global\csname prevpage##1@num\endcsname=\z@%
729       \global\csname prevpage##1@numR\endcsname=\z@%
730     }%
731     {}%
732     \ifcsdef{##1prevpage@num}{%
733       \global\csname ##1prevpage@num\endcsname=\z@%
734       \global\csname ##1prevpage@numR\endcsname=\z@%
735     }%
736     {}%
737   }%
738   \dolistloop{\@series}%
739 }
740 %

```

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.

```

741 \newread\@inputcheck
742 \newcommand*{\read@linelist}[1]{%
743   \ifledRcol%
744     \list@clearing@regR%
745   \else%
746     \list@clearing@reg%
747   \fi%
748 %

```

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).

```

749   \list@clear{\maxlinesinpar@list}
750 %

```

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

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

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

```

751 \get@linelistfile{#1}%
752 \ifcontinuousnumberingwithcolumns
753   \global\page@numR=\page@numR\relax
754   \global\last@page@numR=\last@page@numR\relax
755   \global\page@num=\page@num\relax
756   \global\last@page@num=\last@page@num\relax
757 \fi
758 \@stopmsd%Security if last \endms{} is forgotten
759 \unless\ifledRcol%Get the last line of the last page
760   \cnumgdef{@lastabsline@forpage@the\page@num}{the\absline@num}%
761   \cnumgdef{@lastline@forpage@the\page@num}{the\line@num}%
762 \else%
763   \cnumgdef{@lastabsline@forpageR@the\page@numR}{the\absline@numR}%
764   \cnumgdef{@lastline@forpageR@the\page@numR}{the\line@numR}%

```

```

765 \fi%
766 \endgroup
767 %

```

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.

```

768 \ifledRcol
769   \global\page@numR=\m@ne
770   \ifx\actionlines@listR\empty
771     \gdef\next@actionlineR{1000000}%
772   \else
773     \gl@p\actionlines@listR\to\next@actionlineR
774     \gl@p\actions@listR\to\next@actionR
775   \fi
776 \else
777   \global\page@num=\m@ne
778   \ifx\actionlines@list\empty
779     \gdef\next@actionline{1000000}%
780   \else
781     \gl@p\actionlines@list\to\next@actionline
782     \gl@p\actions@list\to\next@action
783   \fi
784 \fi
785 }
786 %

```

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

```

787 \newcommand*{\list@clearing@reg}{%
788   \list@clear{\line@list}%
789   \list@clear{\insertlines@list}%
790   \list@clear{\actionlines@list}%
791   \list@clear{\actions@list}%
792   \list@clear{\linesinpar@listL}%
793   \list@clear{\linesonpage@listL}%
794 }%
795 %

```

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

```

796 \newcommand*{\get@linelistfile}[1]{%
797   \InputIfFileExists{#1}{%
798     \global\noteschanged@false
799     \begingroup
800       \catcode`\[1 \catcode`\]=2
801       \makeatletter \catcode`\^M=9}{%

```

```

802 \led@warn@NoLineFile{#1}%
803 \global\noteschanged@true
804 \begingroup}%
805 }
806
807 %

```

This version of `\read@linelist` creates list macros containing data for the entire section, so they could get rather large. It would be no more difficult to read the line-list file incrementally rather than all at once: we could read, at the start of each paragraph, only the commands relating to that paragraph. But this would require that we have two line-lists open at once, one for reading, one for writing, and on systems without version numbers we would have to do some file renaming outside of \TeX for that to work. We have retained this slower approach to avoid that sort of hacking about, but have provided the `\pausenumbers` and `\resumenumbers` macros to help you if you run into macro memory limitations (see 5.2.7 p. 18 above).

V.11 Commands within the line-list file

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

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

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

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

```

808 \newcommand{\line@list@version}[1]{%
809   \IfStrEq{#1}{\this@line@list@version}%
810   {}%
811   {\ifledRcol%
812     \led@warn@Obsolete{\jobname.\extensionchars\the\section@num}%
813     \else%
814     \led@warn@Obsolete{\jobname.\extensionchars\the\section@num}%
815     \fi%
816     \endinput%
817   }%
818 }%
819 %

```


`\@nl` `\@nl` does everything related to the start of a new line of numbered text.

`\@nl@reg` In order to get the `\setlinenum` to work Peter Wilson had to slip in some new code at the start of the macro, to get the timing of the actions correct. The problem was that his original naive implementation of `\setlinenum` had a unfortunate tendency to change the number of the last line of the *preceding* paragraph. The new code is sort of based on the page number handling and `\setline`. It seems that a lot of fiddling with the line number internals is required.

In November 2004 in order to accurately determine page numbers Peter Wilson added these to the macro. It is now:

```
\@nl{\page counter number}\{printed page number\}
```

We do not (yet) use the printed number (i.e., the `\thepage`) but it may come in handy later. The macro `\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`.

```
820
821 \newcommand*\@nl}[2]{%
822   \fix@page{#1}%
823   \ifledRcol%
824     \@nl@regR%
825   \else%
826     \@nl@reg%
827   \fi%
828 }
829 \newcommand*\@nl@reg{%
830   \ifx\l@dchset@num\relax \else
831     \advance\absline@num \@ne
832     \set@line@action
833     \let\l@dchset@num=\relax
834     \advance\absline@num \m@ne
835     \advance\line@num \m@ne
836   \fi
837   %
```

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

```
838   \advance\absline@num \@ne
839     \ifx\next@page@num\relax \else
840       \page@action
841       \let\next@page@num=\relax
842     \fi
843     \ifx\sub@change\relax \else
844       \ifnum\sub@change>\z@
845         \sublines@true
846       \else
847         \sublines@false
848       \fi
849       \sub@action
850       \let\sub@change=\relax
```

```

851 \fi
852 %

```

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

```

853 \ifcase\@lock
854 \or
855 \@lock \tw@
856 \or \or
857 \@lock \z@
858 \fi
859 \ifcase\sub@lock
860 \or
861 \sub@lock \tw@
862 \or \or
863 \sub@lock \z@
864 \fi
865 %

```

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

```

866 \ifsublines@
867 \ifnum\sub@lock<\tw@
868 \advance\subline@num \@ne
869 \fi
870 \else
871 \ifnum\@lock<\tw@
872 \advance\line@num \@ne \subline@num \z@
873 \fi
874 \fi}
875
876 %

```

`\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`.

```

877 \newcount\last@page@num
878 \last@page@num=-10000
879
880 \newcommand*{\fix@page}[1]{%
881 \iflabeledRcol
882 \ifnum #1=\last@page@numR
883 \else
884 \csnumgdef{\lastabsline@forpageR@the\page@numR}{\the\absline@numR}%
885 \csnumgdef{\lastline@forpageR@the\page@numR}{\the\line@numR}%
886 \ifbypage@R
887 \line@numR \z@ \subline@numR \z@
888 \fi
889 \page@numR=#1\relax
890 \last@page@numR=#1\relax
891 \def\next@page@numR{#1}%

```

```

892 \fi
893 \else
894 \ifnum #1=\last@page@num
895 \else
896 \csnumgdef{\lastabsline@forpage@the\page@num}{\the\absline@num}%
897 \csnumgdef{\lastline@forpage@the\page@num}{\the\line@num}%
898 \ifbypage@
899 \line@num \z@ \subline@num \z@
900 \fi
901 \page@num=#1\relax
902 \last@page@num=#1\relax
903 \def\next@page@num{#1}%
904 \listxadd{\normal@page@break}{\the\absline@num}
905 \fi
906 \fi}
907 %

```

\@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.

\@lopR

```

908 \newcommand*{\@pend}[1]{}
909 \newcommand*{\@pendR}[1]{}
910 \newcommand*{\@lopL}[1]{}
911 \newcommand*{\@lopR}[1]{}
912
913 %

```

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

```

914 \newcommand*{\sub@on}{\ifsublines@
915 \let\sub@change=\relax
916 \else
917 \def\sub@change{1}%
918 \fi}
919 \newcommand*{\sub@off}{\ifsublines@
920 \def\sub@change{-1}%
921 \else
922 \let\sub@change=\relax
923 \fi}
924
925 %

```

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

```

926
927 \newcommand*{\@adv}[1]{%

```

```

928 \ifsublines@
929 \ifledRcol
930 \advance\subline@numR by #1\relax
931 \ifnum\subline@numR<\z@
932 \led@warn@BadAdvancelineSubline
933 \subline@numR \z@
934 \fi
935 \else
936 \advance\subline@num by #1\relax
937 \ifnum\subline@num<\z@
938 \led@warn@BadAdvancelineSubline
939 \subline@num \z@
940 \fi
941 \fi
942 \else
943 \ifledRcol
944 \advance\line@numR by #1\relax
945 \ifnum\line@numR<\z@
946 \led@warn@BadAdvancelineLine
947 \line@numR \z@
948 \fi
949 \else
950 \advance\line@num by #1\relax
951 \ifnum\line@num<\z@
952 \led@warn@BadAdvancelineLine
953 \line@num \z@
954 \fi
955 \fi
956 \fi
957 \set@line@action}
958
959 %

```

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

```

960
961 \newcommand*{\@set}[1]{%
962 \ifledRcol
963 \ifsublines@
964 \subline@numR=#1\relax
965 \else
966 \line@numR=#1\relax
967 \fi
968 \set@line@action
969 \else
970 \ifsublines@
971 \subline@num=#1\relax
972 \else
973 \line@num=#1\relax

```

```

974 \fi
975 \set@line@action
976 \fi}
977
978 %

```

\l@d@set The `\l@d@set{<num>}` macro sets the line number for the next `\pstart` to the value specified as its argument. This is used to implement `\setlinenum`.

\l@dchset@num `\l@dchset@num` is a flag to the `\@nl?` macro. If it is not `\relax` then a linenum change is to be done.

```

979
980 \newcommand*{\l@d@set}[1]{%
981 \ifledRcol
982 \line@numR=#1\relax
983 \advance\line@numR \@ne
984 \def\l@dchset@num{#1}
985 \else
986 \line@num=#1\relax
987 \advance\line@num \@ne
988 \def\l@dchset@num{#1}
989 \fi}
990 \let\l@dchset@num\relax
991
992 %

```

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

```

993
994 \newcommand*{\page@action}{%
995 \ifledRcol
996 \xright@appenditem{\the\absline@numR}\to\actionlines@listR
997 \xright@appenditem{\next@page@numR}\to\actions@listR
998 \else
999 \xright@appenditem{\the\absline@num}\to\actionlines@list
1000 \xright@appenditem{\next@page@num}\to\actions@list
1001 \fi}
1002 %

```

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

```

1003
1004 \newcommand*{\set@line@action}{%
1005 \ifledRcol
1006 \xright@appenditem{\the\absline@numR}\to\actionlines@listR
1007 \ifsublines@
1008 \@l@dttempcnta=-\subline@numR
1009 \else
1010 \@l@dttempcnta=-\line@numR

```

```

1011 \fi
1012 \advance\@l@tempcnta by -5000\relax
1013 \xright@appenditem{\the\@l@tempcnta}\to\actions@listR
1014 \else
1015 \xright@appenditem{\the\absline@num}\to\actionlines@list
1016 \ifsublines@
1017 \@l@tempcnta=-\subline@num
1018 \else
1019 \@l@tempcnta=-\line@num
1020 \fi
1021 \advance\@l@tempcnta by -5000\relax
1022 \xright@appenditem{\the\@l@tempcnta}\to\actions@list
1023 \fi}
1024 %

```

\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.

```

1025
1026 \newcommand*{\sub@action}{%
1027 \ifl@edRcol
1028 \xright@appenditem{\the\absline@num}\to\actionlines@listR
1029 \ifsublines@
1030 \xright@appenditem{-1001}\to\actions@listR
1031 \else
1032 \xright@appenditem{-1002}\to\actions@listR
1033 \fi
1034 \else
1035 \xright@appenditem{\the\absline@num}\to\actionlines@list
1036 \ifsublines@
1037 \xright@appenditem{-1001}\to\actions@list
1038 \else
1039 \xright@appenditem{-1002}\to\actions@list
1040 \fi
1041 \fi}
1042 %

```

\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.

```

1043 \newcommand*{\lock@on}{\futurelet\next\do@lockon}
1044
1045 \newcommand*{\do@lockon}{%
1046 \ifx\next\lock@off

```

```

1047 \global\let\lock@off=\skip@lockoff
1048 \else
1049 \ifledRcol
1050 \do@lockonR
1051 \else
1052 \do@lockonL
1053 \fi
1054 \fi}
1055
1056
1057 \newcommand*{\do@lockonL}{%
1058 \xright@appenditem{\the\absline@num}\to\actionlines@list
1059 \ifsublines@
1060 \xright@appenditem{-1005}\to\actions@list
1061 \ifnum\sub@lock=\z@
1062 \sub@lock \@ne
1063 \else
1064 \ifnum\sub@lock=\thr@@
1065 \sub@lock \@ne
1066 \fi
1067 \fi
1068 \else
1069 \xright@appenditem{-1003}\to\actions@list
1070 \ifnum\@lock=\z@
1071 \@lock \@ne
1072 \else
1073 \ifnum\@lock=\thr@@
1074 \@lock \@ne
1075 \fi
1076 \fi
1077 \fi}
1078
1079 %

```

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

```

\do@lockoff
\do@lockoffL
\skip@lockoff
1080 \newcommand*{\do@lockoffL}{%
1081 \xright@appenditem{\the\absline@num}\to\actionlines@list
1082 \ifsublines@
1083 \xright@appenditem{-1006}\to\actions@list
1084 \ifnum\sub@lock=\tw@
1085 \sub@lock \thr@@
1086 \else
1087 \sub@lock \z@
1088 \fi
1089 \else
1090 \xright@appenditem{-1004}\to\actions@list
1091 \ifnum\@lock=\tw@
1092 \@lock \thr@@
1093 \else

```

```

1094 \lock \z@
1095 \fi
1096 \fi}
1097
1098 \newcommand*{\do@lockoff}{%
1099 \ifledRcol
1100 \do@lockoffR
1101 \else
1102 \do@lockoffL
1103 \fi}
1104 \newcommand*{\skip@lockoff}{\global\let\lock@off=\do@lockoff}
1105 \global\let\lock@off=\do@lockoff
1106
1107 %

```

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

```

1108 \newcommand*{\n@num}{%
1109 \ifledRcol%
1110 \xright@appenditem{\the\absline@numR}\to\actionlines@listR
1111 \xright@appenditem{-1007}\to\actions@listR
1112 \else%
1113 \xright@appenditem{\the\absline@num}\to\actionlines@list%
1114 \xright@appenditem{-1007}\to\actions@list%
1115 \fi%
1116 }%
1117
1118 %

```

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

```

1119 \newcommand*{\n@num@stanza}{%
1120 \ifledRcol%
1121 \xright@appenditem{\the\absline@numR}\to\actionlines@listR%
1122 \xright@appenditem{-1008}\to\actions@listR%
1123 \else%
1124 \xright@appenditem{\the\absline@num}\to\actionlines@list%%
1125 \xright@appenditem{-1008}\to\actions@list%
1126 \fi%
1127 }
1128 %

```

\ifl@dhidenumber `\hidenum` hides number in margin. It uses action code 1009.

\hidenum

```

\h@num29 \newif\ifl@dhidenumber
1130 \newcommand*{\hidenum}{
1131 \ifledRcol%

```



```

1132 \write\linenum@outR{\string\hide@num}%
1133 \else%
1134 \write\linenum@out{\string\hide@num}%
1135 \fi%
1136 }%
1137 \newcommand*{\hide@num}{%
1138 \ifledRcol%
1139 \xright@appenditem{\the\absline@numR}\to\actionlines@listR%
1140 \xright@appenditem{-1009}\to\actions@listR%
1141 \else%
1142 \xright@appenditem{\the\absline@num}\to\actionlines@listR%
1143 \xright@appenditem{-1009}\to\actions@listR%
1144 \fi%
1145 }
1146 %

```

\@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.

```

1147 \newcount\insert@count
1148 %

```

- #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.

```

1149 \newcommand*{\dummy@ref}[2]{#2}
1150 %

```

\@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.

```

1151 \newcommand*{\@ref}[2]{%
1152 \ifledRcol%
1153 \@ref@regR{#1}{#2}%
1154 \else%
1155 \@ref@reg{#1}{#2}%
1156 \fi%
1157 }%
1158 \newcommand*{\@ref@reg}[2]{%
1159 \global\insert@count=#1\relax
1160 \global\advance\@edtext@level by 1%
1161 \loop\ifnum\insert@count>\z@

```

```

1162 \xright@appenditem{\the\absline@num}\to\insertlines@list
1163 \global\advance\insert@count \m@ne
1164 \repeat
1165 %

```

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.

```

1166 \begingroup
1167 \let\@ref=\dummy@ref
1168 \let\@lopL\@gobble
1169 \let\page@action=\relax
1170 \let\sub@action=\relax
1171 \let\set@line@action=\relax
1172 \let\@lab=\relax
1173 \let\@lemma=\relax%
1174 \let\@sw\@gobblethree%
1175 #2
1176 \global\endpage@num=\page@num
1177 \global\endline@num=\line@num
1178 \global\endsubline@num=\subline@num
1179 \endgroup
1180 %

```

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

```

1181 \xright@appenditem%
1182 {\the\page@num|\the\line@num|}%
1183 \ifsublines@ \the\subline@num \else 0\fi|}%
1184 \the\endpage@num|\the\endline@num|}%
1185 \ifsublines@ \the\endsubline@num \else 0\fi}\to\line@list
1186 %

```

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.

```

1187 \expandafter\list@create\expandafter{\csname sw@list@edtext@tmp@\the\
@edtext@level\endcsname}%
1188 \providebool{lemmacommand@\the\@edtext@level}%
1189 \boolfalse{lemmacommand@\the\@edtext@level}%
1190 %

```

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

```

1191 #2%
1192 %

```

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.

```

1193 \ifnum\@edtext@level>0%
1194 \def\create@this@edtext@level{\expandafter\list@create\expandafter{\
csname sw@list@edtext@\the\@edtext@level\endcsname}}%
1195 \ifcsundef{sw@list@edtext@\the\@edtext@level}{\create@this@edtext@level
}{}%
1196 \letcs{\@tmp}{sw@list@edtext@\the\@edtext@level}%
1197 \letcs{\@tmpp}{sw@list@edtext@tmp@\the\@edtext@level}
1198 \xright@appenditem{\expandonce\@tmpp}\to\@tmp%
1199 \global\cslet{sw@list@edtext@\the\@edtext@level}{\@tmp}%
1200 \fi%
1201 %
Decrease edtext level counter.
1202 \global\advance\@edtext@level by -1%
1203 %
1204 }
1205
1206 %

```

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`.

```

1207 \newwrite\linenum@out
1208 %

```

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

We can manage pretty well by means of the `\iffirst@linenum@out` flag; its 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.

```

1209 \newif\iffirst@linenum@out
1210 \iffirst@linenum@out@true
1211 %

```

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

```
1212 \newcommand{\this@line@list@version}{5}%
1213 %
```

`\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.

```
1214 \newcommand*{\line@list@stuff}[1]{%
1215 %
```

First, use the commands of the previous section to interpret the line-list file from the last run.

```
1216 \read@linelist{#1}%
1217 %
```

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.

```
1218 \iffirst@linenum@out@
1219 \immediate\closeout\linenum@out%
1220 \global\first@linenum@out@false%
1221 \immediate\openout\linenum@out=#1\relax%
1222 \immediate\write\linenum@out{\string\line@list@version{\
this@line@list@version}}%
1223 \ifl@dpaging%
1224 \immediate\write\linenum@out{\string\@par@sync@option{\
@par@this@sync@option}}%
1225 \fi%
1226 \else
1227 %
```

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.

```
1228 \if@minipage%
1229 \leavevmode%
1230 \fi%
1231 \closeout\linenum@out%
1232 \openout\linenum@out=#1\relax%
1233 \write\linenum@out{\string\line@list@version{\this@line@list@version}}%
1234 %
1235 \ifl@dpaging%
1236 \write\linenum@out{\string\@par@sync@option{\@par@this@sync@option}}%
1237 %
1238 \fi%
```

```

1237 \fi}
1238
1239 %

```

\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.

```

1240 \newcommand*{\new@line}{%
1241   \IfStrEq{\led@pb@setting}{after}%
1242     {\xifinlist{\the\absline@num}{\l@prev@nopb}%
1243       {\xifinlist{\the\absline@num}{\normal@page@break}%
1244         {\numgdef{\@next@page}{\c@page+\@ne}%
1245           \write\linenum@out{\string\@nl[\@next@page][\@next@page]}}%
1246         }%
1247       {\write\linenum@out{\string\@nl[\the\c@page][\thepage]}}}%
1248   }%
1249   {\write\linenum@out{\string\@nl[\the\c@page][\thepage]}}}%
1250 }%
1251 \IfStrEq{\led@pb@setting}{before}%
1252   {\numgdef{\next@absline}{\the\absline@num+\@ne}%
1253     \xifinlist{\next@absline}{\l@prev@nopb}%
1254       {\xifinlist{\the\absline@num}{\normal@page@break}%
1255         {\numgdef{\nc@page}{\c@page+\@ne}%
1256           \write\linenum@out{\string\@nl[\nc@page][\nc@page]}}%
1257         }%
1258       {\write\linenum@out{\string\@nl[\the\c@page][\thepage]}}}%
1259   }%
1260   {\write\linenum@out{\string\@nl[\the\c@page][\thepage]}}}%
1261 }%
1262 {}%
1263 \IfStrEqCase{\led@pb@setting}{{before}{\relax}{after}{\relax}}{\write\linenum@out{\string\@nl[\the\c@page][\thepage]}}%
1264 }
1265
1266 %

```

\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 **\flag@end** 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.

```

1267 \newif\if@noneed@Footnote%
1268
1269 \newcommand*{\flag@start}{%
1270   \ifledRcol%
1271     \edef\next{\write\linenum@outR{%
1272       \string\@ref[\the\insert@countR][ ]}%
1273     \next%

```

```

1274 \ifnum\insert@countR<1%
1275   \if@noneed@Footnote\else%
1276     \led@err@EdtextWithoutFootnote%
1277   \fi%
1278 \fi%
1279 \else%
1280   \edef\next{\write\linenum@out{%
1281     \string\@ref[\the\insert@count] []}%
1282   \next%
1283   \ifnum\insert@count<1%
1284     \if@noneed@Footnote\else%
1285       \led@err@EdtextWithoutFootnote%
1286     \fi%
1287   \fi%
1288 \fi}%
1289
1290 %

```

\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.

```

1291
1292
1293 \newcommand*{\startsub}{\dimen0\lastskip
1294   \ifdim\dimen0>0pt \unskip \fi
1295   \ifledRcol \write\linenum@outR{\string\sub@on}%
1296   \else      \write\linenum@out{\string\sub@on}%
1297   \fi
1298   \ifdim\dimen0>0pt \hskip\dimen0 \fi}
1299 \def\endsub{\dimen0\lastskip
1300   \ifdim\dimen0>0pt \unskip \fi
1301   \ifledRcol \write\linenum@outR{\string\sub@off}%
1302   \else      \write\linenum@out{\string\sub@off}%
1303   \fi
1304   \ifdim\dimen0>0pt \hskip\dimen0 \fi}
1305
1306 %

```

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

```

1307 \newcommand*{\advanceline}[1]{\leavevmode%
1308   \ifledRcol \write\linenum@outR{\string\@adv[#1]}%
1309   \else      \write\linenum@out{\string\@adv[#1]}%
1310   \fi}%
1311 }
1312 %

```

\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.

```

1313 \newcommand*{\setline}[1]{%
1314   \leavevmode%
1315   \ifnum#1<\z@
1316     \led@warn@BadSetline
1317   \else
1318     \ifledRcol \write\linenum@outR{\string\@set[#1]}%
1319     \else      \write\linenum@out{\string\@set[#1]}%
1320     \fi
1321   \fi}
1322 %
1323 %
1324 %

```

\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.

```

1325 \newcommand*{\setlinenum}[1]{%
1326   \ifnum#1<\z@
1327     \led@warn@BadSetlinenum
1328   \else
1329     \ifledRcol \write\linenum@outR{\string\l@d@set[#1]}
1330     \else      \write\linenum@out{\string\l@d@set[#1]} \fi
1331   \fi}
1332 %
1333 %
1334 %

```

\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.

```

1335 \newcommand*{\startlock}{%
1336   \ifledRcol \write\linenum@outR{\string\lock@on}%
1337   \else      \write\linenum@out{\string\lock@on}%
1338   \fi}
1339 \def\endlock{%
1340

```

```

1341 \ifledRcol \write\linenum@outR{\string\lock@off}%
1342 \else      \write\linenum@out{\string\lock@off}%
1343 \fi}
1344 %

```

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

```

\l@dskipnumbertrue
\l@dskipnumberfalse
\skipnumbering
1345 \newif\ifl@dskipnumber
1346 \newif\ifl@dskipversenumber%
1347 \newcommand*{\skipnumbering}{%
1348   \leavevmode%
1349   \ifledRcol%
1350     \ifinstanza%
1351       \write\linenum@outR{\string\n@num@stanza}%
1352     \else%
1353       \write\linenum@outR{\string\n@num}%
1354     \fi%
1355     \advanceline{-1}%
1356   \else%
1357     \ifinstanza%
1358       \write\linenum@out{\string\n@num@stanza}%
1359     \else%
1360       \write\linenum@out{\string\n@num}%
1361     \fi%
1362     \advanceline{-1}%
1363   \fi%
1364 }%
1365 %
1366 %

```

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. 133). Instead, the appropriate note-generating command is appended to the list macro `\inserts@list`, and when `\pend` completes the paragraph it inserts all the notes at the proper places.

Note that we do not provide previous-note information, although it is often wanted; your own macros must handle that. We can not do it correctly without keeping track of what kind of notes have gone past: it is not just a matter of remembering the line numbers associated with the previous invocation of `\edtext`, because that might have been for a different kind of note. It is preferable for your footnote macros to store and recall this kind of information if they need it.

VI.1 `\edtext` itself

The various note-generating macros might want to request that commands be executed not at once, but in close connection with the start or end of the lemma. For example, footnote numbers in the text should be connected to the end of the lemma; or, instead of a single macro to create a note listing variants, you might want to use several macros in series to create individual variants, which would each add information to a private macro or token register, which in turn would be formatted and output when all of #2 for the lemma has been read.

`\end@lemmas` To accomodate this, we provide a list macro to which macros may add commands that should subsequently be executed at the end of the lemma when that lemma is added to the text of the paragraph. A macro should add its contribution to `\end@lemmas` by using `\xleft@appenditem`. (Anything that needs to be done at the *start* of the lemma may be handled using `\aftergroup`, since the commands specified within `\edtext`'s second argument are executed within a group that ends just before the lemma is added to the main text.)

`\end@lemmas` is intended for the few things that need to be associated with the end of the lemma, like footnote numbers. Such numbers are not implemented in the current

version, and indeed no use is currently made of `\end@lemmas` or of the `\aftergroup` trick. The general approach would be to define a macro to be used within the second argument of `\edtext` that would add the appropriate command to `\end@lemmas`.

Commands that are added to this list should always take care not to do anything that adds possible line-breaks to the output; otherwise line numbering could be thrown off.

```
1367 \list@create{\end@lemmas}
1368 %
```

`\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.

```
1369 \newcommand{\dummy@edtext}[2]{#1}
1370 %
```

`\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`.

```
1371 \newcommand{\dummy@edtext@showlemma}[2]{\showlemma{#1}}%
1372 %
```

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

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

The first class is font-changing macros. We suppress expansion for them by letting them become equal to zero.²⁴ 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

²⁴Since 'control sequences equivalent to characters are not expandable'—*The \TeX book*, answer to Exercise 20.14.

`\edtext` itself, if it appears within its own argument, do nothing but copy its first argument.

Finally, we execute `\morenoexpands`. The version of `\morenoexpands` defined here does nothing; but you may define a version of your own when you need to add more expansion suppressions as needed with your macros. That makes it possible to make such additions without needing to copy or modify the standard `reledmac` code. If you define your own `\morenoexpands`, you must be very careful about spaces: if the macro adds any spaces to the text when it runs, extra space will appear in the main text when `\edtext` is used.

(A related problem, not addressed by these two macros, is that of characters whose category code is changed by any 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 character, using Lua_{TeX} or Xe_{La}TeX.)

```

1373 \newcommand*{\no@expands}{%
1374   \let\select@lemmafnt=0%
1375   \let\startsub=\relax \let\endsub=\relax
1376   \let\startlock=\relax \let\endlock=\relax
1377   \let\edlabel=\@gobble
1378   \let\setline=\@gobble \let\advanceline=\@gobble
1379   \let\sameword\sameword@inedtext%
1380   \let\edtext=dummy@edtext
1381   \l@dtabnoexpands
1382   \morenoexpands}
1383 \let\morenoexpands=\relax
1384
1385 %

```

`\@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.

```

1386 \newcommand{\@tag}{}
1387 %

```

`\@edtext@level` This counter is increased by 1 at each level of `\edtext`. That is useful for some commands which can have a different behavior if called inside or outside of the `{\lemma}` argument.

```

1388 \newcount\@edtext@level%
1389 \@edtext@level=0%
1390 %

```

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

```
1391 \newcounter{edtext}
1392 \renewcommand{\theedtext}{edt@arabic{edtext}}%
1393 %
```

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

```
1394 \newcommand{\edtext}[2]{\leavevmode%
1395 %
```

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

```
1396 \ifnumberedpar%
1397 %
```

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

```
1398 \global\advance\@edtext@level by 1%
1399 %
```

We also increase the `edtext` LaTeX counter to insert `hypertarget` if the `hyperref` package is loaded.

```
1400 \stepcounter{edtext}%
1401 %
```

By default, we do not use `\lemma`

```
1402 \global\@lemmacommand@false%
1403 %
```

```
1404 \begingroup%
1405 %
```

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`.

```
1406 \ifledRcol%
1407 \ifcvoid{sw@list@edtextR@the\@edtext@level}%
1408 {\global\let\sw@inthisedtext\empty}%
1409 {\expandafter\gl@p\csname sw@list@edtextR@the\@edtext@level\
endcsname\to\sw@inthisedtext}%
1410 \else%
1411 \ifcvoid{sw@list@edtext@the\@edtext@level}%
1412 {\global\let\sw@inthisedtext\empty}%
1413 {\expandafter\gl@p\csname sw@list@edtext@the\@edtext@level\
endcsname\to\sw@inthisedtext}%
1414 \fi%
1415 %
```

\@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.

```

1416     \global\renewcommand{\@tag}{\%
1417         \no@expands #1%
1418     }%
1419 %

```

\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`.

```

1420     \set@line%
1421 %

```

`\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`.

```

1422     \ifledRcol \global\insert@countR \z@%
1423     \else      \global\insert@count \z@ \fi%
1424 %

```

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.

```

1425     \ignorespaces #2\relax%
1426 %

```

With `polyglossia`, you must track whether the language reads left to right (English) or right to left (Arabic).

```

1427     \@ifundefined{xpg@main@language}{%if not polyglossia
1428         \flag@start}%
1429     {\if@RTL\flag@end\else\flag@start\fi%
1430     }%
1431 %

```

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

```

1432     \if@lemmacommand@%
1433     \ifledRcol%
1434         \write\linenum@outR{\string\@lemma}%
1435     \else%

```

```

1436     \write\linenum@out{\string\@lemma}%
1437     \fi%
1438     \fi%
1439 %

```

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.

```

1440     \endgroup%
1441     \ifdef{\hypertarget}%
1442     {%
1443         \csedef{thisedtext@the\@edtext@level}{\theedtext}%
1444         \Hy@raisedlink@left{\hypertarget{\csuse{thisedtext@the\@edtext@level}:start}{}}%
1445         \showlemma{#1}%
1446         \Hy@raisedlink{\hypertarget{\csuse{thisedtext@the\@edtext@level}:end}{}}%
1447     }%
1448     {%
1449         \showlemma{#1}%
1450     }%
1451 %

```

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.

```

1452     \ifx\end@lemmas\empty \else%
1453         \glp\end@lemmas\to\x@lemma%
1454         \x@lemma%
1455         \global\let\x@lemma=\relax%
1456     \fi%
1457     \@ifundefined{xpg@main@language}{%if not polyglossia
1458         \flag@end}%
1459         {\ifRTL\flag@start\else\flag@end\fi% With polyglossia, you must
track whether the language reads left to right (English) or right to left
(Arabic).
1460         }%
1461 %

```

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`.

```

1462 \global\@noneed@Footnotefalse%
1463 \global\advance\@edtext@level by -1%
1464 \global\@lemmacommand@false%
1465 %

```

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

```

1466 \else%
1467 \showlemma{#1} (\textbf{\textsc{Edtext outside numbered paragraph}})\
led@err@edtextoutsidestart%
1468 \fi%
1469 }%
1470
1471 \newcommand*{\flag@end}{%
1472 \ifledRcol%
1473 \write\linenum@outR{]}%
1474 \else%
1475 \write\linenum@out{]}%
1476 \fi}%
1477
1478 %

```

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

```

1479 \newif\ifnumberline
1480 \numberlinetrue
1481 %

```

\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.

```

1482 \newcommand*{\set@line}{%
1483 \ifledRcol
1484 \ifx\line@listR\empty
1485 \global\noteschanged@true
1486 \xdef\l@d@nums{000|000|000|000|000|000|\edfont@info}%
1487 \else
1488 \glp\line@listR\to\@tempb
1489 \xdef\l@d@nums{\@tempb|\edfont@info}%
1490 \global\let\@tempb=\undefined
1491 \fi
1492 \else

```

```

1493 \ifx\line@list\empty
1494 \global\noteschanged@true
1495 \xdef\l@d@nums{000|000|000|000|000|000|\edfont@info}%
1496 \else
1497 \gl@p\line@list\to\@tempb
1498 \xdef\l@d@nums{\@tempb|\edfont@info}%
1499 \global\let\@tempb=\undefined
1500 \fi
1501 \fi}
1502
1503 %

```

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

```

1504 \newcommand*{\edfont@info}{\f@encoding/\f@family/\f@series/\f@shape}
1505
1506 %

```

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. 117).

```

1507 \newcommand*{\lemma}[1]{%
1508 \global\@lemmacommand@true%
1509 \global\renewcommand{\@tag}{%
1510 \no@expands #1%
1511 }%
1512 \ignorespaces%
1513 }%
1514 %

```

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

```

1515 \newcommand{\@lemma}{%
1516 \booltrue{lemmacommand@the\@edtext@level}%
1517 }%
1518 %

```

\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.

```

1519 \newif\if@lemmacommand@%
1520 %

```


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. 90): 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.

```

1521 \newcommand*{\linenum}[1]{%
1522   \xdef\@tempa{#1|||||noexpand\\l@d@nums}%
1523   \global\let\l@d@nums=\empty
1524   \expandafter\line@set\@tempa|\\ignorespaces}
1525 %

```

`\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.

```

1526 \def\line@set#1|#2\\#3|#4\\{%
1527 \gdef\@tempb{#1}%
1528 \ifx\@tempb\empty
1529 \l@{#3}%
1530 \else
1531 \l@{#1}%
1532 \fi
1533 \gdef\@tempb{#4}%
1534 \ifx\@tempb\empty\else
1535 \l@{#4}%
1536 \fi}
1537 %

```

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

```

1538 \newcommand{\l@d@add}[1]{\xdef\l@d@nums{\l@d@nums#1}}
1539
1540 %

```

VI.4 Lemma disambiguation

The mechanism which counts the occurrence of a same word in a same line is quite complex, because, when L^AT_EX 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 `eledmac` 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 explain how for same `\sameword` absolute ranks are transformed to relative rank.

```
At line 1:
absolute rank 1 becomes relative rank 1-0 = 1
1 is stored for this \sameword and the line 1
At line 2:
absolute rank 2 becomes relative rank 2-1 = 1
absolute rank 3 becomes relative rank 3-2 = 2
3 is stored for this \sameword and the line 2
At line 3:
no \sameword for this line.
3 is stored for this \sameword and the line 3
At line 4:
absolute rank 4 becomes relative rank 4-3 = 1
3 is stored for this \sameword and the line 4
```

2. Create lists of lists of `\sameword` by depth of `\edtext`. That is: create a list for `\edtext` of level 1, a list for `\edtext` of level 2, a list for `\edtext` of level 3 etc. For each `\edtext` in these list, we store all the relative rank of `\saweword` which are called as lemma information, that is 1) or called in the first argument of `\sameword` 2) or called in the `\lemma` macro of the second argument of `\sameword` AND marked by the optional argument of `\saweword` in first argument of `\edtext`.

For example, suppose a line with nested `\edtexts` which contains some word marked by `\sameword` and having the following relative rank:

bar¹ foo¹ foo² bar² foo³ (A)(B) foo⁴ bar³ (C) foo⁵ (D) bar⁴ (E)

In this example, all lemma information for `\edtext` is framed. The text in parenthesis is the content of critical notes associated to the preceding frame. As you can see, we have two level 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 is `\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 list of relative rank 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 method to know the absolute line number associated to a `\sameword` is required because a `\edtext` can be overlap many lines, but `\sameword` can't get it.
 - * We get the value associated, when reading the auxiliary file, to the pair compose by the current marked word and the current absolute line number. Let this value be n .
 - If $n > 1$, that mean the current word appears more than once time in its line. In this case, we call `\showwordrank` with the word as first argument and r as 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 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 dynamic name of macro with `\csname`.²⁵

²⁵See <http://tex.stackexchange.com/q/244538/7712>.

Because there is a bug with `\detokenize` and \XeTeX when using non BMP characters²⁶, we detokenize only for not \XeTeX engines. In any case, in \XeTeX , a `\csname` construction can contain UTF-8 characters without a problem, as UTF-8 characters are not managed with category code, but instead read directly as UTF-8 characters.

```

1541 \newcommand{\get@sw@txt}[1]{%
1542   \ifxetex%
1543     \xdef\sw@txt{#1}%
1544   \else%
1545     \expandafter\xdef\expandafter\sw@txt\expandafter{\detokenize{#1}}%
1546   \fi%
1547 }%
1548 %

```

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

```

1549 \newcommandx{\sameword}[2][1,usedefault]{%
1550   \leavevmode%
1551   \get@sw@txt{#2}%
1552 %

```

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

```

1553   \unless\ifledRcol%
1554     \csnumgdef{sw@\sw@txt}{\csuse{sw@\sw@txt}+\@ne}%
1555 %

```

Then, write its value to the numbered file.

```

1556   \protected@write\linenum@out{\string\@sw{\sw@txt}{\csuse{sw@\sw@txt}
1557 }}{#1}}%
1557 %

```

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

```

1558   \else%
1559     \csnumgdef{sw@\sw@txt}{\csuse{sw@\sw@txt}+\@ne}%
1560     \protected@write\linenum@outR{\string\@sw{\sw@txt}{\csuse{sw@\sw@txt}
1561 }}{#1}}%
1561   \fi%
1562 %

```

And print the word.

```

1563   #2%
1564 }%
1565 %

```

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

²⁶<http://sourceforge.net/p/xetex/bugs/108/>

```
\if@addsw66 \newif\if@addsw%
1567 %
```

\@sw The command printed in the auxiliary files.

```
1568 \newcommand{\@sw}[3]{%
1569   \get@sw@txt{#1}%
1570   \unless\ifledRcol%
1571 %
```

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

```
1572   \csxdef{sw@sw@txt @\the\absline@num @\the\section@num}{#2}%
1573 %
```

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

```
1574   \numdef{\prev@line}{\the\absline@num-1}%
1575   \ifcsundef{sw@sw@txt @\prev@line @\the\section@num}{%
1576     \csnumgdef{sw@sw@txt @\prev@line @\the\section@num}{#2-1}%
1577   }{%
1578 %
```

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

```
1579   \numdef{\the@sw}{#2-\csuse{sw@sw@txt @\prev@line @\the\section@num}}%
1580 %
```

And do the same thing for the right side.

```
1581 \else%
1582   \csxdef{sw@sw@txt @\the\absline@numR @\the\section@numR @R}{#2}%
1583   \numdef{\prev@line}{\the\absline@numR-1}%
1584   \ifcsundef{sw@sw@txt @\prev@line @\the\section@numR @R}{%
1585     \csnumgdef{sw@sw@txt @\prev@line @\the\section@numR @R}{#2-1}%
1586   }{%
1587     \numdef{\the@sw}{#2-\csuse{sw@sw@txt @\prev@line @\the\section@numR @R}}%
1588   }%
1589 %
```

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

```
1590 \@tempcnta=\@edtext@level
1591 \@whilenum{\@tempcnta>0}\do{%
1592   \ifcsdef{sw@list@edtext@tmp@\the\@tempcnta}%
1593   {%
1594     \@addswfalse%
1595     \notbool{lemmacommand@\the\@tempcnta}%
1596     {\@addswtrue}%
1597     {\IfStrEq{#3}{inlemma}%
1598      {\@addswtrue}%
1599     }%
1600   }
```

```

1600         \def\do##1{%
1601             \ifnumequal{##1}{\the\@tempcnta}%
1602                 {\@addswtrue\listbreak}%
1603                 {}%
1604             }%
1605             \docsvlist{#3}%
1606         }%
1607     }%
1608     \if@addsw%
1609         \letcs{\@tmp}{sw@list@edtext@tmp@\the\@tempcnta}%
1610         \ifledRcol%
1611             \xright@appenditem{{\the@sw}{\the\absline@numR}}\to\@tmp%
1612         \else%
1613             \xright@appenditem{{\the@sw}{\the\absline@num}}\to\@tmp%
1614         \fi%
1615         \cslet{sw@list@edtext@tmp@\the\@tempcnta}{\@tmp}%
1616     \fi%
1617 }%
1618 {}%
1619 \advance\@tempcnta by -1%
1620 }%
1621 }%
1622 %

```

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

```

1623 \newcommandx{\sameword@inedtext}[2][1,usedefault]{%
1624     \get@sw@txt{#2}%
1625     \unless\ifledRcol%
1626 %

```

Just a precaution.

```

1627     \ifx\sw@list@inedtext\empty%
1628         \def\the@sw{999}%
1629         \def\this@absline{-99}%
1630     \else%
1631 %

```

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

```

1632     \gl@p\sw@list@inedtext\to\@tmp%
1633     \edef\the@sw{\expandafter\@firstoftwo\@tmp}%
1634     \edef\this@absline{\expandafter\@secondoftwo\@tmp}%
1635     \fi%
1636 %

```

First, calculate the number of occurrences of the word in the current line

```

1637     \ifcsdef{sw@\sw@txt @\this@absline @\the\section@num}{%
1638         \numdef{\prev@line}{\this@absline-1}%

```

```

1639 \numdef{\sw@atthisline}{\csuse{sw@sw@txt @\this@absline @\the\
section@num}-\csuse{sw@sw@txt @\prev@line @\the\section@num}}%
1640 }%
1641 {\numdef{\sw@atthisline}{0}}%
1642 %

```

Finally, print the rank, but only if there is more than one occurrence of the word in the current line.

```

1643 \ifnumgreater{\sw@atthisline}{1}%
1644 {\showwordrank{#2}{\the@sw}}%
1645 {#2}%
1646 %

```

And the same for right side.

```

1647 \else%
1648 \ifx\sw@list@inedtext\empty%
1649 \def\the@sw{999}%
1650 \def\this@absline{-99}%
1651 \else%
1652 \gl@p\sw@list@inedtext\to\@tmp%
1653 \edef\the@sw{\expandafter\@firstoftwo\@tmp}%
1654 \edef\this@absline{\expandafter\@secondoftwo\@tmp}%
1655 \fi%
1656 \ifcsdef{sw@sw@txt @\this@absline @\the\section@numR @R}{%
1657 \numdef{\prev@line}{\this@absline-1}%
1658 \numdef{\sw@atthisline}{\csuse{sw@sw@txt @\this@absline @\the\
section@numR @R}-\csuse{sw@sw@txt @\prev@line @\the\section@numR @R}}%
1659 }%
1660 {\numdef{\sw@atthisline}{0}}%
1661 \ifnumgreater{\sw@atthisline}{1}%
1662 {\showwordrank{#2}{\the@sw}}%
1663 {#2}%
1664 \fi%
1665 }%
1666 %

```

`\showwordrank`⁶⁷ % Finally, the way the rank will be printed.

```

1668 \newcommand{\showwordrank}[2]{%
1669 #1\textsuperscript{#2}%
1670 }%
1671 %

```

VII Paragraph decomposition and reassembly

In order to be able to count the lines of text and affix line numbers, we add an extra stage of processing for each paragraph. We send the paragraph into a box register, rather than straight onto the vertical list, and when the paragraph ends we slice the paragraph into

its component lines; to each line we add any notes or line numbers, add a command to write to the line-list, and then at last send the line to the vertical list. This section contains all the code for this processing.

VII.1 Boxes, counters, \pstart and \pend

`\raw@text` Here are numbers and flags that are used internally in the course of the paragraph decomposition.

`\ifnumberedpar@` When we first form the paragraph, it goes into a box register, `\raw@text`, instead of onto the current vertical list. The `\ifnumberedpar@` flag will be true while a paragraph is being processed in that way. `\num@lines` will store the number of lines in the paragraph when it is complete. When we chop it up into lines, each line in turn goes into the `\one@line` register, and `\par@line` will be the number of that line within the paragraph.

```

1672 \newbox\raw@text
1673 \newif\ifnumberedpar@
1674 \newcount\num@lines
1675 \newbox\one@line
1676 \newcount\par@line
1677 %

```

`\pstart` `\pstart` starts the paragraph by clearing the `\inserts@list` list and other relevant variables, and then arranges for the subsequent text to go into the `\raw@text` box.

`\AtEveryPstart` `\pstart` needs to appear at the start of every paragraph that is to be numbered; the `\autopar` command below may be used to insert these commands automatically.

`\numberpstarttrue` `\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.

`\numberpstartfalse` `\pstart` needs to appear at the start of every paragraph that is to be numbered; the `\autopar` command below may be used to insert these commands automatically.

`\labelpstarttrue` 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.

`\labelpstartfalse`

`\thepstart`

```

1678
1679 \newcommand{\AtEveryPstart}[1]{%
1680   \ifstrempy{#1}%
1681     {\xdef\at@every@pstart{}}%
1682     {\gdef\at@every@pstart{\noindent#1}}%
1683 }%
1684 \xdef\at@every@pstart{}%
1685
1686 \newcounter{pstart}
1687 \renewcommand{\thepstart}{\bfseries\@arabic{c@pstart}. }
1688 \newif\ifnumberpstart
1689 \numberpstartfalse
1690 \newif\iflabelpstart
1691 \labelpstartfalse
1692 \newcommandx*\pstart[1][1]{%
1693   \normal@pars%
1694   \ifstrempy{#1}{\at@every@pstart}{\noindent#1}%
1695   \ifluatex%
1696     \edef\l@luatexttextdir@L{\the\textdir}%

```



```

1697 \fi%
1698 \@nobraektrue%
1699 \ifnumbering \else%
1700 \led@err@PstartNotNumbered%
1701 \beginnumbering%
1702 \fi%
1703 \ifnumberedpar@%
1704 \led@err@PstartInPstart%
1705 \pend%
1706 \fi%
1707 \list@clear{\inserts@list}%
1708 \global\let\next@insert=\empty%
1709 \begingroup%
1710 \global\advance \l@dnumpstartsL\@ne
1711 \global\setbox\raw@text=\vbox\bgroup%
1712 \ifautopar\else%
1713 \ifnumberpstart%
1714 \ifinstanza\else%
1715 \ifsidepstartnum\else%
1716 \thepstart%
1717 \fi%
1718 \fi%
1719 \fi%
1720 \fi%
1721 \numberedpar@true%
1722 \iflabelpstart\protected@edef\@currentlabel%
1723 {\p@pstart\thepstart}
1724 \fi%
1725 \l@dzeroopenalties%
1726 \ignorespaces%because not automatically ignored if an optional argument
is used (classical TeX behavior for space after commands)
1727 }
1728 %

```

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

```

1729 \newcommand*{\pend}[1][1]{\ifnumbering \else%
1730 \led@err@PendNotNumbered%
1731 \fi%
1732 \global\l@dskipversenumberfalse%
1733 \ifnumberedpar@ \else%
1734 \led@err@PendNoPstart%
1735 \fi%
1736 %

```

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.

```

1737 \l@dzeroopenalties%
1738 \endgraf\global\num@lines=\prevgraf\egroup%
1739 \global\par@line=0%
1740 %

```

We check if lineation is by pstart: in this case, we reset 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.

```

1741 \csnumdef{pstartline}{0}%
1742 \loop\ifvbox\raw@text%
1743   \csnumdef{pstartline}{\pstartline+\@ne}%
1744   \do@line%
1745   \ifbypstart@%
1746     \ifnumequal{pstartline}{1}{%
1747       \bgroup%
1748       \let\leavevmode\relax%
1749       \setline{1}%
1750       \egroup%
1751       \resetprevline@}{}%
1752     \fi%
1753   \repeat%
1754 %

```

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

```

1755 \flush@notes%
1756 \endgroup%
1757 \ignorespaces%
1758 %

```

Increase pstart counter.

```

1759 \ifnumberpstart%
1760   \pstartnumtrue%
1761 \fi%
1762 \addtocounter{pstart}{1}%
1763 %

```

Print the optional argument of \pend or the content printed after every \pend

```

1764 \normal@pars%
1765 \ifstrempy{#1}{\at@every@pend}{\noindent#1}%
1766 %

```

Restore standard nobreak setting and autopar setting. Normally, \if@nobreak is equal to true only immediately after a sectioning command (read latex.ltx file). As a \pstart...\pend structure can't contain any sectioning command, we set \if@nobreak to false.

```

1767 \@nobreakfalse%
1768 \ifautopar%
1769   \autopar%
1770 \fi%
1771 }
1772
1773 %

```

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.

```

\AtEveryPend 74
\at@every@pend 75 \newcommand{\AtEveryPend}[1]{%
1776   \ifstrepty{#1}%
1777     {\xdef\at@every@pend{}}%
1778     {\gdef\at@every@pend{\noindent#1}}%
1779 }%
1780 \xdef\at@every@pend{}%
1781
1782 %

```

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

```

1783 \newcommand*\l@dzzeropenalties{%
1784   \brokenpenalty \z@ \clubpenalty \z@
1785   \displaywidowpenalty \z@ \interlinepenalty \z@ \predisplaypenalty \z@
1786   \postdisplaypenalty \z@ \widowpenalty \z@}
1787
1788 %

```

`\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.

```

1789 \newif\ifautopar
1790 \autoparfalse
1791 \newcommand*{\autopar}{
1792   \ifledRcol
1793     \ifnumberingR \else
1794       \led@err@AutoparNotNumbered
1795       \beginnumberingR
1796       \fi
1797     \else
1798       \ifnumbering \else
1799         \led@err@AutoparNotNumbered
1800         \beginnumbering
1801         \fi
1802       \fi
1803       \autopartrue
1804       \everypar{\setbox0=\lastbox
1805         \endgraf \vskip-\parskip
1806         \pstart \noindent \kern\wd0 \ifnumberpstart\ifinstanza\else\thepstart\fi\fi
1807         \let\par=\pend}%
1808       \ignorespaces}
1809 %

```

`\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.

```

1810 \newcommand*{\normal@pars}{\ifautopar\everypar{}\fi\let\par\endgraf}
1811
1812 %

```

`\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.

```

1813 \newif\ifautopar@pause
1814 %

```

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`

```

1815 \newcommand*{\l@dunhbox@line}[1]{\unhbox #1}
1816 \newcommand*{\do@line}{%
1817   {\vbadness=10000
1818     \splittopskip=\z@
1819     \do@linehook
1820   \l@demptyd@ta
1821     \global\setbox\one@line=\vsplit\raw@text to\baselineskip}%
1822   \unvbox\one@line \global\setbox\one@line=\lastbox
1823   \getline@num
1824   \IfStrEq{\led@pb@setting}{before}{\led@check@pb\led@check@nopb}{%
1825     \ifnum\@lock>\@ne
1826       \inserthangingsymboltrue
1827     \else
1828       \inserthangingsymbolfalse
1829     \fi
1830     \check@pb@in@verse
1831     \ifl@dhidenumber%
1832       \global\l@dhidenumberfalse%
1833       \f@x@l@cks%
1834     \else%
1835       \affixline@num%
1836     \fi%
1837   }%

```

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

```

1838 \xifinlist{\the\l@dunpstartsl}{\eled@sections@@}%
1839   {\print@eledsection}%
1840   {\print@line}%
1841   \IfStrEq{\led@pb@setting}{after}{\led@check@pb\led@check@nopb}{%
1842   }%
1843 }%

```

VII.2.2 Process for “normal” line

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

```

1844 \def\print@line{
1845   %

```

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

```

1846     \affixpstart@num%
1847   %

```

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

```
1848 \hb@xt@ \linewidth{ %
1849 %
```

User hook.

```
1850 \do@insidelinehook %
1851 %
```

Left line number

```
1852 \l@dld@ta %
1853 %
```

Prepare text to be inserted before notes.

```
1854 \if@firstlineofpage %
1855 \set@Xtxtbeforenotes %
1856 \global\@firstlineofpagefalse %
1857 \fi %
1858 %
```

Insert footnotes made of manuscripts data.

```
1859 \insert@msdata %
1860 %
```

Restore marginal and footnotes.

```
1861 \add@inserts\affixside@note %
1862 %
```

Print left notes.

```
1863 \l@dlsn@te %
1864 %
```

Boxes the line, writes information about new line in the numbered file.

```
1865 {\ledllfill\hb@xt@ \wd\one@line{\new@line %
1866 %
```

If we use Lua \TeX then restore the direction.

```
1867 \ifluatex %
1868 \textdir\l@luatextextdir@L %
1869 \fi %
1870 %
```

Insert, if needed, the hanging symbol.

```
1871 \inserthangingsymbol %
1872 %
```

And so, print the line.

```
1873 \l@dunhbox@line{\one@line}} %
1874 %
```

Right line number

```
1875 \ledrlfill\l@drd@ta%
1876 %
```

Print right notes.

```
1877 \l@drsn@te
1878 }}%
1879 %
```

And reinsert penalties (for page breaking)...

```
1880 \add@penalties%
1881 }
1882 %
```

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.

```
1883 \def\print@eledsection{%
1884   \if@firstlineofpage%
1885     \set@Xtxtbeforenotes%
1886     \global\@firstlineofpagefalse%
1887   \fi%
1888   \insert@msdata%
1889   \add@inserts\affixside@note%
1890   \numdef{\temp@}{\l@dnumpstartsL-1}%
1891   \xifinlist{\temp@}{\eled@sections@}{\@nbreaktrue}{\@nbreakfalse}%
1892   \@eled@sectioningtrue%
1893   \csuse{eled@sectioning@the\l@dnumpstartsL}%
1894   \@eled@sectioningfalse%
1895   \global\csundef{eled@sectioning@the\l@dnumpstartsL}%
1896   \if@RTL%
1897     \hspace{-3\paperwidth}%
1898     {\hbox{\l@dunhbox@line{\one@line}} \new@line}%
1899   \else%
1900     \hspace{3\paperwidth}%
1901     {\new@line \hbox{\l@dunhbox@line{\one@line}}}%
1902   \fi%
1903   \vskip-\baselineskip%
1904 }
1905 %
```

VII.2.4 Hooks

\do@linehook Two hooks into \do@line. The first is called at the beginning of \do@line, the second
\do@insidelinehook is called in the line box. The second can, for example, have a \markboth command

inside, the first ca not.

```
1906 \newcommand*{\do@linehook}{}
1907 \newcommand*{\do@insidelinehook}{}
1908 %
```

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

```
1909 \newcommand*{\dolinehook}[1]{\gdef\do@linehook{#1}}%
1910 \newcommand*{\doinsidelinehook}[1]{\gdef\do@insidelinehook{#1}}%
1911
1912 %
```

VII.2.5 Sidenotes and marginal line number initialization

`\l@emptyd@ta` Nulls the `\...d@ta`, which may later hold line numbers. Similarly for `\l@dcsnotetext`,
`\l@dld@ta` `\l@dcsnotetext@l`, `\l@dcsnotetext@r` for the texts of the sidenotes, left and right
`\l@drd@ta` notes.

```
\l@dcsnotetext
1913 \newcommand*{\l@emptyd@ta}{%
\l@dcsnotetext@l
1914 \gdef\l@dld@ta{}%
\l@dcsnotetext@r
1915 \gdef\l@drd@ta{}%
1916 \gdef\l@dcsnotetext@l{}%
1917 \gdef\l@dcsnotetext@r{}%
1918 \gdef\l@dcsnotetext{}%
1919
1920 %
```

`\l@dlsn@te` Zero width boxes of the left and right side notes, together with their kerns.

```
\l@drsn@te
1921 \newcommand{\l@dlsn@te}{%
1922 \hb@xt@ \z@{\hss\box\l@dldp@rbox\kern\ledlsnotesep}}
1923 \newcommand{\l@drsn@te}{%
1924 \hb@xt@ \z@{\kern\ledrsnotesep\box\l@drp@rbox\hss}}
1925
1926 %
```

`\ledllfill` These macros are called at the left (`\ledllfill`) and the right (`\ledllfill`) of each
`\ledrlfill` numbered line. The initial definitions correspond to the original code for `\do@line`.

```
1927 \newcommand*{\ledllfill}{\hfil}
1928 \newcommand*{\ledrlfill}{}
1929
1930 %
```


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.

```

1931 \newcommand*{\getline@num}{%
1932   \global\advance\absline@num \@ne%
1933   \do@actions
1934   \do@ballast
1935   \ifnumberline
1936     \ifsublines@
1937       \ifnum\sub@lock<\tw@
1938         \global\advance\subline@num \@ne
1939       \fi
1940     \else
1941       \ifnum\@lock<\tw@
1942         \global\advance\line@num \@ne
1943         \global\subline@num \z@
1944       \fi
1945     \fi
1946   \fi
1947 }
1948 %

```

`\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 XI.2 p. 146).

`\ballast@count` First we set up the required counters; they are initially set to zero, and will remain so unless you type `\setcounter{ballast}{\langle some figure \rangle}` in your document.

`\c@ballast`

```

1949 \newcount\ballast@count
1950 \newcounter{ballast}
1951 \setcounter{ballast}{0}
1952 %

```

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.

```

1953 \newcommand*{\do@ballast}{\global\ballast@count \z@
1954   \begingroup
1955     \advance\absline@num \@ne
1956     \ifnum\next@actionline=\absline@num
1957       \ifnum\next@action>-1001\relax
1958         \global\advance\ballast@count by -\c@ballast
1959       \fi
1960     \fi

```

```

1961 \endgroup}
1962 %

```

`\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.

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`.

```

1963 \newcommand*{\do@actions}{%
1964   \global\let\do@actions@next=\relax
1965   \ifnum\absline@num<\next@actionline\else
1966   %

```

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.

```

1967   \ifnum\next@action>-1001
1968     \global\page@num=\next@action
1969     \global\@firstlineofpagetrue%
1970     \ifbypage@
1971       \global\line@num=\z@ \global\subline@num=\z@
1972       \resetprevline@
1973     \fi
1974     \add@msdata@firstlineofpage%
1975   %

```

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`.)

```

1976   \else
1977     \ifnum\next@action<-4999
1978       \@l@tempcnta=-\next@action
1979       \advance\@l@tempcnta by -5001
1980       \ifsublines@
1981         \global\subline@num=\@l@tempcnta
1982       \else
1983         \global\line@num=\@l@tempcnta
1984       \fi
1985   %

```

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

```

1986   \else
1987     \@l@tempcnta=-\next@action
1988     \advance\@l@tempcnta by -1000
1989     \do@actions@fixedcode
1990   \fi
1991 \fi
1992 %

```

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.

```

1993   \ifx\actionlines@list\empty
1994       \gdef\next@actionline{1000000}%
1995   \else
1996       \gl@p\actionlines@list\to\next@actionline
1997       \gl@p\actions@list\to\next@action
1998       \global\let\do@actions@next=\do@actions
1999   \fi
2000 \fi
2001 %

```

Make the recursive call, if necessary.

```

2002 \do@actions@next}
2003
2004 %

```

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

```

2005 \newcommand*{\do@actions@fixedcode}{%
2006   \ifcase\@l@dttempcnta
2007   \or% % 1001 = starting sublineation
2008       \global\sublines@true
2009   \or% % 1002 = ending sublineation
2010       \global\sublines@false
2011   \or% % 1003 = starting locking number
2012       \global\@lock=\@ne
2013   \or% % 1004 = ending locking number
2014       \ifnum\@lock=\tw@
2015           \global\@lock=\thr@@
2016       \else
2017           \global\@lock=\z@
2018       \fi
2019   \or% % 1005 = starting locking subnumber
2020       \global\sub@lock=\@ne
2021   \or% % 1006 = ending locking subnumber
2022       \ifnum\sub@lock=\tw@
2023           \global\sub@lock=\thr@@
2024       \else
2025           \global\sub@lock=\z@
2026       \fi
2027   \or% % 1007 = skipping numbering
2028       \l@dskipnumbertrue
2029   \or% % 1008 = skipping numbering in stanza
2030       \l@dskipversenumbertrue%
2031   \or% % 1009 = hiding number
2032       \l@dhiddenumbertrue

```

```

2033 \or% % 1010 = inserting msdata
2034 \add@msdata%
2035 \else
2036 \led@warn@BadAction
2037 \fi}
2038
2039
2040 %

```

IX 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.

```

2041 \newcommand*{\affixline@num}{%
2042 %

```

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).

```

2043 \ifledgroupnotesL@else
2044 \ifnumberline
2045 \ifl@dskipnumber
2046 \global\l@dskipnumberfalse
2047 \else
2048 \ifsublines@
2049 \@l@tempcntb=\subline@num
2050 \ifnum\subline@num>\c@firstsublinenum
2051 \@l@tempcnta=\subline@num
2052 \advance\@l@tempcnta by-\c@firstsublinenum
2053 \divide\@l@tempcnta by\c@sublinenumincrement
2054 \multiply\@l@tempcnta by\c@sublinenumincrement
2055 \advance\@l@tempcnta by\c@firstsublinenum
2056 \else

```

```

2057         \@l@dttempcnta=\c@firstsublinenum
2058     \fi
2059 %

```

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.

```

2060         \ch@cksub@l@ck
2061 %

```

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

```

2062     \else
2063         \@l@dttempcntb=\line@num
2064 %

```

Check on the `\linenumberlist` If it is `\empty` use the standard algorithm.

```

2065         \ifx\linenumberlist\empty
2066             \ifnum\line@num>\c@firstlinenum
2067                 \@l@dttempcnta=\line@num
2068                 \advance\@l@dttempcnta by-\c@firstlinenum
2069                 \divide\@l@dttempcnta by\c@linenumincrement
2070                 \multiply\@l@dttempcnta by\c@linenumincrement
2071                 \advance\@l@dttempcnta by\c@firstlinenum
2072             \else
2073                 \@l@dttempcnta=\c@firstlinenum
2074         \fi
2075     \else
2076 %

```

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

```

2077         \@l@dttempcnta=\line@num
2078         \edef\rem@inder{\linenumberlist,\number\line@num,}%
2079         \edef\sc@n@list{\def\noexpand\sc@n@list
2080             ###1,\number\@l@dttempcnta,####2|{\def\noexpand\rem@inder
2081 {####2}}}%
2082         \sc@n@list\expandafter\sc@n@list\rem@inder|
2083         \ifx\rem@inder\empty%
2084             \advance\@l@dttempcnta\@ne
2085         \fi
2086 %

```

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

```

2087         \ch@ck@l@ck
2088     \fi
2089 %

```

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

```

2090 \ifnum\@l@tempcnta=\@l@tempcntb
2091 \ifl@dskipversenumber\else
2092 %

```

If we got here, we are going to print a line number; so now we need to calculate a number that will tell us which side of the page will get the line number. We start from `\line@margin`, which asks for one side always if it is less than 2; and then if the side does depend on the page number, we simply add the page number to this side code—because the values of `\line@margin` have been devised so that this produces a number that is even for left-margin numbers and odd for right-margin numbers.

For \LaTeX we have to consider two column documents as well. In this case Peter Wilson thought we need to put the numbers at the outside of the column — the left of the first column and the right of the second. Do the `twocolumn` stuff before going on with the original code.

`\l@dld@ta` A left line number is stored in `\l@dld@ta` and a right one in `\l@drd@ta`.

```

\l@drd@ta
2093 \if@twocolumn
2094 \if@firstcolumn
2095 \gdef\l@dld@ta{\llap{{\leftlinenum}}}%
2096 \else
2097 \gdef\l@drd@ta{\rlap{{\rightlinenum}}}%
2098 \fi
2099 \else
2100 \l@tempcntb=\line@margin
2101 \ifnum\@l@tempcntb>\@ne
2102 \advance\@l@tempcntb \page@num
2103 \fi
2104 \ifodd\@l@tempcntb
2105 \gdef\l@drd@ta{\rlap{{\rightlinenum}}}%
2106 \else
2107 \gdef\l@dld@ta{\llap{{\leftlinenum}}}%
2108 \fi
2109 \fi
2110 \fi
2111 \fi
2112 %

```

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

```

2113 \f@x@l@cks
2114 \fi
2115 \fi
2116 \fi
2117 }
2118
2119 %

```

`\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.

```

2120 \newcommand*{\ch@cksub@l@ck}{%
2121   \ifcase\sub@lock
2122   \or
2123     \ifnum\sublock@disp=\@ne
2124       \@l@tempcntb=\z@ \@l@tempcnta=\@ne
2125     \fi
2126   \or
2127     \ifnum\sublock@disp=\tw@ \else
2128       \@l@tempcntb=\z@ \@l@tempcnta=\@ne
2129     \fi
2130   \or
2131     \ifnum\sublock@disp=\z@
2132       \@l@tempcntb=\z@ \@l@tempcnta=\@ne
2133     \fi
2134   \fi}
2135 %

```

Similarly for line numbers.

```

2136 \newcommand*{\ch@ck@l@ck}{%
2137   \ifcase\@lock
2138   \or
2139     \ifnum\lock@disp=\@ne
2140       \@l@tempcntb=\z@ \@l@tempcnta=\@ne
2141     \fi
2142   \or
2143     \ifnum\lock@disp=\tw@ \else
2144       \@l@tempcntb=\z@ \@l@tempcnta=\@ne
2145     \fi
2146   \or
2147     \ifnum\lock@disp=\z@
2148       \@l@tempcntb=\z@ \@l@tempcnta=\@ne
2149     \fi
2150   \fi}
2151 %

```

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

```

2152 \newcommand*{\f@x@l@cks}{%
2153   \ifcase\@lock
2154   \or
2155     \global\@lock=\tw@
2156   \or \or
2157     \global\@lock=\z@
2158   \fi
2159   \ifcase\sub@lock
2160   \or

```

```

2161 \global\sub@lock=\tw@
2162 \or \or
2163 \global\sub@lock=\z@
2164 \fi}
2165
2166 %

```

X 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:

- 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.

```

\leftpstartnum67
\rightpstartnum68 \newif\ifsidepstartnum
\ifsidepstartnum69 \newcommand*{\affixpstart@num}{%
2170 \ifsidepstartnum
2171 \if@twocolumn
2172 \if@firstcolumn
2173 \gdef\l@dld@ta{\llap{\leftpstartnum}}}%
2174 \else
2175 \gdef\l@drd@ta{\rlap{\rightpstartnum}}}%
2176 \fi
2177 \else
2178 \l@dttempcntb=\line@margin
2179 \ifnum\l@dttempcntb>\@ne
2180 \advance\l@dttempcntb \page@num
2181 \fi
2182 \ifodd\l@dttempcntb
2183 \gdef\l@drd@ta{\rlap{\rightpstartnum}}}%
2184 \else
2185 \gdef\l@dld@ta{\llap{\leftpstartnum}}}%
2186 \fi
2187 \fi
2188 \fi
2189
2190 }
2191 %
2192
2193 \newif\ifpstartnum

```



```

2194 \pstartnumtrue
2195 \newcommand*{\leftpstartnum}{
2196     \ifpstartnum\thepstart
2197     \kern\linenumsep\fi
2198     \global\pstartnumfalse
2199 }
2200 \newcommand*{\rightpstartnum}{
2201     \ifpstartnum
2202     \kern\linenumsep
2203     \thepstart
2204     \fi
2205     \global\pstartnumfalse
2206 }
2207 %

```

XI 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.

XI.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.

```

2208 \list@create{\inserts@list}
2209 %

```

`\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`.

```

2210 \newcommand*{\add@inserts}{%
2211     \global\let\add@inserts@next=\relax
2212 %

```

If `\inserts@list` is empty, there are not any more notes or insertions for this paragraph, and we need not waste our time.

```

2213 \ifx\inserts@list\empty \else
2214 %

```

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.

```

2215 \ifx\next@insert\empty
2216 \ifx\insertlines@list\empty
2217 \global\noteschanged@true
2218 \gdef\next@insert{100000}%
2219 \else
2220 \glp\insertlines@list\to\next@insert
2221 \fi
2222 \fi
2223 %

```

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.

```

2224 \ifnum\next@insert=\absline@num
2225 \glp\inserts@list\to\@insert
2226 \@insert
2227 \global\let\@insert=\undefined
2228 \global\let\next@insert=\empty
2229 \global\let\add@inserts@next=\add@inserts
2230 \fi
2231 \fi
2232 %

```

Make the recursive call, if necessary.

```

2233 \add@inserts@next}
2234
2235 %

```

XI.2 Penalties

`\add@penalties` `\add@penalties` is the last macro used by `\do@line`. It adds up the club, widow, and interline penalties, and puts a single penalty of the appropriate size back into the paragraph; these penalties get removed by the `\vsplit` operation. `\displaywidowpenalty` and `\brokenpenalty` are not restored, since we have no easy way to find out where we should insert them.

In this code, `\num@lines` is the number of lines in the whole paragraph, and `\par@line` is the line we are working on at the moment. The count `\@l@dttempcnta` is used to calculate and accumulate the penalty; it is initially set to the value of `\ballast@count`, which has been worked out in `\do@ballast` above (VIII p. 137). Finally, the penalty is checked to see that it does not go below -10000 .

```

2236 \newcommand*{\add@penalties}{\@l@tempcnta=\ballast@count
2237 \ifnum\num@lines>\@ne
2238 \global\advance\par@line \@ne
2239 \ifnum\par@line=\@ne
2240 \advance\@l@tempcnta \clubpenalty
2241 \fi
2242 \@l@tempcntb=\par@line \advance\@l@tempcntb \@ne
2243 \ifnum\@l@tempcntb=\num@lines
2244 \advance\@l@tempcnta \widowpenalty
2245 \fi
2246 \ifnum\par@line<\num@lines
2247 \advance\@l@tempcnta \interlinepenalty
2248 \fi
2249 \fi
2250 \ifnum\@l@tempcnta=\z@
2251 \relax
2252 \else
2253 \ifnum\@l@tempcnta>-10000
2254 \penalty\@l@tempcnta
2255 \else
2256 \penalty -10000
2257 \fi
2258 \fi}
2259
2260 %

```

XI.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.

```

2261 \newcommand*{\flush@notes}{%
2262 \@xloop
2263 \ifx\inserts@list\empty \else
2264 \gl@p\inserts@list\to\@insert
2265 \@insert
2266 \global\let\@insert=\undefined
2267 \repeat}
2268
2269 %

```

\@xloop `\@xloop` is a variant of the PLAIN \TeX `\loop` macro, useful when it's hard to construct a positive test using the \TeX `\if` commands—as in `\flush@notes` above. One types

`\@xloop ... \if ... \else ... \repeat`, and the action following `\else` is repeated as long as the `\if` test fails. (This macro will work wherever the `PLAIN TeX \loop` is used, too, so we could just call it `\loop`; but it seems preferable not to change the definitions of any of the standard macros.)

This variant of `\loop` was introduced by Alois Kabsch in *TUGboat* 8 (1987), pp. 184–5.

```
2270 \def\@xloop#1\repeat{%
2271   \def\body{#1\expandafter\body\fi}%
2272   \body}
2273
2274 %
```

XI.4 Text before notes

`\set@Xtxtbeforenotes` The `\set@Xtxtbeforenotes` macro resets the `Xtxtbeforenotes@⟨series⟩@typeset` boolean to false. Just before the first note of the `⟨series⟩` in a page, the `Xtxtbeforenotes` will be inserted.

```
2275 \newcommand{\set@Xtxtbeforenotes}{%
2276   \unless\ifnocritical@%
2277     \def\do##1{%
2278       \global\togglefalse{Xtxtbeforenotes@##1@typeset}%
2279     }%
2280     \dolistloop{\@series}%
2281   \fi%
2282 }%
2283 %
```

`\insert@Xtxtbeforenotes` `\insert@Xtxtbeforenotes{⟨series⟩}`, called when inserting a note, 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.

```
2284 \newcommand{\insert@Xtxtbeforenotes}[1]{%
2285   \nottoggle{Xtxtbeforenotes@#1@typeset}{%
2286     \global\toggletrue{Xtxtbeforenotes@#1@typeset}%
2287     \ifcsvoid{Xtxtbeforenotes@#1}{%
2288       \ifcsstring{series@display#1}{paragraph}%
2289       {\noindent\cuse{Xtxtbeforenotes@#1}}%
2290       {\expandafter\insert\cuse{#1footins}\endcsname%
2291         \bgroup%
2292         \noindent\strut\cuse{Xnotefontsize@#1}\cuse{Xtxtbeforenotes@
2293         #1}%
2294         \egroup%
2295       }%
2296     }%
2297   }%
2298 }
```

```

2298 }%
2299 %

```

XII 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.

XII.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.

```

2300 \def\select@lemmafont#1|#2|#3|#4|#5|#6|#7|{\select@@lemmafont#7|}%
2301 \def\select@@lemmafont#1/#2/#3/#4|{%
2302   {\fontencoding{#1}\fontfamily{#2}\fontseries{#3}\fontshape{#4}%
2303    \selectfont}%
2304
2305 %

```

XII.2 Individual note options

`\footnoteoptions@` The `\footnoteoption@[<side>]{<options>}{<value>}` changes the value of on options of Xfootnote, to switch between true and false.

```

2306 \newcommand*\footnoteoptions@[3]{%
2307   \def\do##1{%
2308     \ifstrequal{#1}{L}{% In Leftside
2309       \xright@appenditem{\noexpand\setkeys[mac]{#3footnoteoption}{\
unexpanded{##1}}}{\to\inserts@list%
2310         \global\advance\insert@count \@ne% Increment the left insert
counter.
2311       }%
2312     }%
2313     \xright@appenditem{\noexpand\setkeys[mac]{#3footnoteoption}{\
unexpanded{##1}}}{\to\inserts@listR%
2314       \global\advance\insert@countR \@ne% Increment the right insert
counter insert.

```

```

2315 }%
2316 }%
2317 \notblank{#2}{\docsvlist{#2}}{}}% Parsing all options
2318 }
2319 %

```

XII.3 Notes language

`\footnotelang@lua` `\footnotelang@lua` is called to remember the information about the direction of a lemma when Lua²TeX is used.

```

2320 \newcommand*{\footnotelang@lua}[1][1=L,usedefault]{%
2321   \ifstrequal{#1}{L}{%
2322     \xright@appenditem{\csxdef{footnote@luatextextdir}{\the\textdir}}{\to\
inserts@list%Know the dir of lemma
2323     \global\advance\insert@count \@ne%
2324     \xright@appenditem{\csxdef{footnote@luatexpardir}{\the\pardir}}{\to\
inserts@list%Know the dir of lemma
2325     \global\advance\insert@count \@ne%
2326   }%
2327   {%
2328     \xright@appenditem{\csxdef{footnote@luatextextdir}{\the\textdir}}{\to\
inserts@listR%Know the dir of lemma
2329     \global\advance\insert@countR \@ne%
2330     \xright@appenditem{\csxdef{footnote@luatexpardir}{\the\pardir}}{\to\
inserts@listR%Know the dir of lemma
2331     \global\advance\insert@countR \@ne%
2332   }%
2333 }
2334 %

```

`\footnotelang@poly` `\footnotelang@poly` is called to remember the information about the language of a lemma when polyglossia is used.

```

2335 \newcommand*{\footnotelang@poly}[1][1=L,usedefault]{%
2336   \ifstrequal{#1}{L}{%
2337     \if@RTL%
2338       \xright@appenditem{\csxdef{footnote@dir}{@RTLtrue}}{\to\
inserts@list%Know the language used in the lemma
2339       \global\advance\insert@count \@ne%
2340     }else
2341       \xright@appenditem{\csxdef{footnote@dir}{@RTLfalse}}{\to\
inserts@list%Know the language of lemma
2342       \global\advance\insert@count \@ne%
2343     }fi%
2344     \xright@appenditem{\csxdef{footnote@lang}{\expandonce\language}}{\
to\inserts@list%Know the language of lemma
2345     \global\advance\insert@count \@ne%
2346   }%

```

```

2347 {%
2348   \if@RTL
2349     \xright@appenditem{{\csxdef{footnote@dir}{@RTLtrue}}}\to\
inserts@listR%Know the language of lemma
2350     \global\advance\insert@countR \@ne%
2351   \else
2352     \xright@appenditem{{\csxdef{footnote@dir}{@RTLfalse}}}\to\
inserts@listR%Know the language of lemma
2353     \global\advance\insert@countR \@ne%
2354   \fi
2355   \xright@appenditem{{\csxdef{footnote@lang}{\expandonce\language}}}\
to\inserts@listR%Know the language of lemma
2356   \global\advance\insert@countR \@ne%
2357 }%
2358 }
2359 %

```

XII.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”.

XII.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`.

```

2360 \newcommand*{\footplitskips}{%
2361   \interlinepenalty=\interfootnotelinepenalty
2362   \unless\ifl@dprintingpages%
2363     \floatingpenalty=\@MM%
2364   \fi%
2365   \splittopskip=\ht\strutbox \splitmaxdepth=\dp\strutbox
2366   \leftskip=\z@skip \rightskip=\z@skip}
2367
2368 %

```

`\normalfootnoterule` `\normalfootnoterule` is a standard footnote-rule macro, for use by a `footstart` macro: just the same as the PLAIN \TeX footnote rule.

```

2369 \let\normalfootnoterule=\footnoterule
2370 %

```

XII.6 Footnotes arrangement

XII.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.

```

2371 \newcommandx{\Xarrangement}[2][1,usedefault]{%
2372   \def\do##1{%
2373     \csname Xarrangement@#2\endcsname{##1}%
2374   }%
2375   \ifstrepty{#1}%
2376     {%
2377       \dolistloop{\@series}%
2378     }%
2379     {
2380       \docsvlist{#1}%
2381     }%
2382   }%
2383 %

```

XII.6.2 Normal footnote

`\Xarrangement@normal` We can now define all the parameters for the series of footnotes; initially they use the “normal” footnote formatting.

What we want to do here is to insert something like the following for each footnote series. (This is an example, not part of the actual `reledmac` code.)

```
\skip\Afootins=12pt plus5pt minus5pt
```



```

\count\Afootins=1000
\dimen\Afootins=0.8\vsiz
\let\Afootnote=\normalvfootnote \let\Afootfmt=\normalfootfmt
\let\Afootstart=\normalfootstart \let\Afootgroup=\normalfootgroup
\let\Afootnoterule=\normalfootnoterule

```

(Read *The TeXbook* in order to understand what are the counter, skip and dimen associated to an insertion.)

Instead of repeating ourselves, we define a `\Xarrangement@normal` macro that makes all these assignments for us, for any given series letter. This command is called when people use `\Xarrangement[⟨series⟩]{normal}`

Now we set up the `\Xarrangement@normal` macro itself. It takes one argument: the footnote series letter.

```

2384 \newcommand*{\Xarrangement@normal}[1]{%
2385   \csgdef{series@display#1}{normal}
2386   \expandafter\let\csname #1footstart\endcsname=\normalfootstart
2387   \expandafter\let\csname v#1footnote\endcsname=\normalvfootnote
2388   \expandafter\let\csname #1footfmt\endcsname=\normalfootfmt
2389   \expandafter\let\csname #1footgroup\endcsname=\normalfootgroup
2390   \expandafter\let\csname #1footnoterule\endcsname=%
2391                                     \normalfootnoterule
2392   \count\csname #1footins\endcsname=1000
2393   \dimen\csname #1footins\endcsname=\csuse{Xmaxhnotes@#1}
2394   \skip\csname #1footins\endcsname=\csuse{Xbeforenotes@#1}%
2395   \advance\skip\csname #1footins\endcsname by\csuse{Xafterrule@#1}%
2396   %

```

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.

```

2397   \csxdef{default@#1footins}{1000}%Use this to confine the notes to one
2398   side only
2399   %

```

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).

```

2399   \ifnoledgroup@else%
2400     \expandafter\let\csname mpv#1footnote\endcsname=\mpnormalvfootnote
2401     \expandafter\let\csname mp#1footgroup\endcsname=\mpnormalfootgroup
2402     \count\csname mp#1footins\endcsname=1000
2403     \dimen\csname mp#1footins\endcsname=\csuse{Xmaxhnotes@#1}
2404     \skip\csname mp#1footins\endcsname=\csuse{Xbeforenotes@#1}%
2405     \advance\skip\csname mp#1footins\endcsname by\csuse{Xafterrule@#1}%
2406   \fi
2407 }
2408
2409 %

```

`\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.

```

2410 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\normalvfootnote}[2]{%
2411   \insert@Xtxtbeforenotes{#1}%
2412   \csuse{Xbeforeinserting@#1}%
2413   \insert\csname #1footins\endcsname\bgroup
2414   \hsize=\expandafter\dimexpr\csuse{Xwidth@#1}\relax%
2415   \noindent\csuse{Xbhooknote@#1}%
2416   \csuse{Xnotefontsize@#1}%
2417   \footsplitskips
2418   \ifl@dpairing\ifl@dpadding\else%
2419     \setXnoteswidthliketwocolumns@{#1}%
2420   \fi\fi%
2421   \setXnotespositionliketwocolumns@{#1}%
2422   \spaceskip=\z@skip \xspaceskip=\z@skip
2423   \csname #1footfmt\endcsname #2{#1}\egroup}
2424 %

```

`\mpnormalvfootnote` And a somewhat different version for minipages.

```

2425 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\mpnormalvfootnote}[2]{%
2426   \global\setbox\@nameuse{mp#1footins}\vbox{%
2427     \insert@Xtxtbeforenotes{#1}%
2428     \unvbox\@nameuse{mp#1footins}
2429     \noindent\csuse{Xbhooknote@#1}%
2430     \csuse{Xnotefontsize@#1}%
2431     \hsize\columnwidth
2432     \@parboxrestore
2433     \color@begingroup
2434     \csname #1footfmt\endcsname #2{#1}\color@endgroup}}
2435 %
2436 %

```

`\normalfootfmt` `\normalfootfmt` is a ‘normal’ macro to take the footnote line and page number information (see V.9 p. 90), 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. 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.

```

2437 %
2438 %
2439 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\normalfootfmt}[4]{%
2440   \Xledsetnormalparstuff{#4}%
2441   \hangindent=\csuse{Xhangindent@#4}
2442   \everypar{\hangindent=\csuse{Xhangindent@#4}}%

```

```

2443 \strut{\printlinefootnote{#1}{#4}}%
2444 \print@lemma{#1}{#2}{#4}%
2445 \cuse{Xwrapcontent@#4}{#3}%
2446 \strut\par}
2447 %

```

\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 \skipXfootins 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 \preXnotes@ 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.

```

2448 \newcommand*{\normalfootstart}[1]{%
2449 %

```

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 XVIII p. 202. Here is part of this algorithm, when the block of notes are ready to be printed.

```

2450 \ifdimequal{0pt}{\preXnotes@}{}%
2451 {%
2452 \iftoggle{preXnotes@}{%
2453 \togglefalse{preXnotes@}%
2454 \skip\cename #1footins\endcename=%
2455 \glueexpr\cuse{preXnotes@}+\cuse{Xafterrule@#1}\relax%
2456 }%
2457 }%
2458 }%
2459 \vskip\skip\cename #1footins\endcename%
2460 %

```

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 XV p. 200 for the general description of the problem.

```

2461 \leftskip0pt \rightskip0pt
2462 \ifl@dpairing\else%
2463 \hsize=\old@hsize%

```

```

2464 \fi%
2465 \setXnoteswidthliketwocolumns@{#1}%
2466 \setXnotespositionliketwocolumns@{#1}%
2467 %

```

And now, print the footnote's rule to finish the footnote's introduction.

```

2468 \print@Xfootnoterule{#1}%
2469 }%
2470 %

```

\normalfootgroup \normalfootgroup is a standard footnote-grouping macro: it sends the contents of the footnote-insert box to the output page without alteration.

```

2471 \newcommand*{\normalfootgroup}[1]{%
2472 \csuse{Xhookgroup@#1}%
2473 \unvbox\csname #1footins\endcsname%
2474 \hsize=\old@hsize%
2475 }%
2476
2477 %

```

\mpnormalfootgroup A somewhat different version for minipages. Note that, in this case, we do not make distinctions between the \Xfootgroup and \Xfootstarts macros.

```

2478 \unless\ifnoledgroup@
2479 \newcommand*{\mpnormalfootgroup}[1]{%
2480 \vskip\skip\@nameuse{mp#1footins}
2481 \ifl@dpairing\ifparledgroup%
2482 \leavevmode\marks\parledgroup@{begin}%
2483 \marks\parledgroup@series{#1}%
2484 \marks\parledgroup@type{Xfootnote}%
2485 \fi\fi\normalcolor%
2486 \ifparledgroup%
2487 \ifl@dpairing%
2488 \else%
2489 \setXnoteswidthliketwocolumns@{#1}%
2490 \setXnotespositionliketwocolumns@{#1}%
2491 \print@Xfootnoterule{#1}%
2492 \fi%
2493 \else%
2494 \setXnoteswidthliketwocolumns@{#1}%
2495 \setXnotespositionliketwocolumns@{#1}%
2496 \print@Xfootnoterule{#1}%
2497 \fi%
2498 \setlength{\parindent}{0pt}
2499 \csuse{Xhookgroup@#1}%
2500 \unvbox\csname mp#1footins\endcsname}}
2501 \fi
2502 %

```

XII.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.

```

2503 \newcommand*\Xarrangement@paragraph}[1]{%
2504   \csgdef{series@display#1}{paragraph}
2505   \expandafter\newcount\csname #1prevpage@num\endcsname
2506   \expandafter\newcount\csname #1prevpage@numR\endcsname%
2507   \expandafter\let\csname #1footstart\endcsname=\parafootstart
2508   \expandafter\let\csname v#1footnote\endcsname=\paravfootnote
2509   \expandafter\let\csname #1footfmt\endcsname=\parafootfmt
2510   \expandafter\let\csname #1footgroup\endcsname=\parafootgroup
2511   \count\csname #1footins\endcsname=1000
2512   \csxdef{default@#1footins}{1000}%Use this to confine the notes to one
side only
2513   \dimen\csname #1footins\endcsname=\csuse{Xmaxhnotes@#1}
2514   \skip\csname #1footins\endcsname=\csuse{Xbeforenotes@#1}%
2515   \advance\skip\csname #1footins\endcsname by\csuse{Xafterrule@#1}%
2516   \para@footsetup{#1}
2517   %

```

And the extra setup for minipages.

```

2518 \ifnoledgroup@else
2519   \expandafter\let\csname mpv#1footnote\endcsname=\mpparavfootnote
2520   \expandafter\let\csname mp#1footgroup\endcsname=\mpparafootgroup
2521   \count\csname mp#1footins\endcsname=1000
2522   \dimen\csname mp#1footins\endcsname=\csuse{Xmaxhnotes@#1}
2523   \skip\csname mp#1footins\endcsname=\csuse{Xbeforenotes@#1}%
2524   \advance\skip\csname mp#1footins\endcsname by\csuse{Xafterrule@#1}%
2525   \fi
2526 }
2527 %

```

`\footfudgefiddle` For paragraphed footnotes T_EX has to estimate the amount of space required. If it underestimates this then the notes may get too long and run off the bottom of the text block. `\footfudgefiddle` can be increased from its default 64 (say, to 70) to increase the estimate.

```

2528 \providecommand{\footfudgefiddle}{64}
2529 %

```

\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 \TeX not \hsize. Peter Wilson have also included \footfudgefiddle.

```

2530 \newcommand*{\para@footsetup}[1]{\csuse{Xhookgroup@#1}\csuse{
Xnotesize@#1}
2531 \setXnoteswidthliketwocolumns@{#1}%
2532 \ifcempty{Xwidth@#1}%
2533 {}%
2534 {\columnwidth=\expandafter\dimexpr\csuse{Xwidth@#1}\relax}%
2535 \dimen0=\baselineskip
2536 \multiply\dimen0 by 1024
2537 \divide \dimen0 by \columnwidth \multiply\dimen0 by \footfudgefiddle\
relax
2538 \csxdef{#1footfudgefactor}{%
2539 \expandafter\strip@pt\dimen0 }}
2540
2541 %

```

\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.

```

2542 \newcommand*{\parafootstart}[1]{%
2543 \rightskip=0pt \leftskip=0pt%
2544 \nottoggle{Xparindent@#1}{\parindent=\z@}{}%
2545 \ifdimequal{0pt}{\preXnotes@}{}%
2546 {%
2547 \iftoggle{preXnotes@}{%
2548 \togglefalse{preXnotes@}%
2549 \skip\csname #1footins\endcsname=%
2550 \glueexpr\csuse{preXnotes@}+\csuse{Xafterrule@#1}\relax%
2551 }%
2552 }%
2553 }%
2554 \vskip\skip\csname #1footins\endcsname%
2555 \setXnoteswidthliketwocolumns@{#1}%
2556 \setXnotespositionliketwocolumns@{#1}%

```

```

2557 \print@Xfootnoterule{#1}%
2558 \let\bidirTL@everypar\@empty%
2559 \noindent\leavevmode}
2560 %

```

`\paravfootnote` `\paravfootnote` is a version of the `\vfootnote` command that is used for paragraphed notes. It gets appended to the `\inserts@list` list by an outer-level footnote command like `\Afootnote`. The first argument is the note series letter; the second is the full text of the printed note itself, including line numbers, lemmata, and footnote text.

The initial model for this insertion is, of course, the `\insert\footins` definition in *The TeXbook*, p. 398. There, the footnotes are first collected up in `hboxes`, and these `hboxes` are later unpacked and stuck together into a paragraph.

However, Michael Downes has pointed out that because text in `hboxes` gets typeset in restricted horizontal mode, there are some undesirable side-effects if you later want to break such text across lines. In restricted horizontal mode, where \TeX does not expect to have to break lines, it does not insert certain items like `\discretionary`s. If you later unbox these `hboxes` and stick them together, as the *TeXbook* macros do to make these footnotes, you lose the ability to hyphenate after an explicit hyphen. This can lead to overfull `hboxes` when you would not expect to find them, and to the uninitiated it might be very hard to see why the problem had arisen.²⁷

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.²⁸ So changes from one language to another will not invoke the proper hyphenation rules in such footnotes. Since critical editions often do deal with several languages, especially in a footnotes, we really ought to get this bit of code right.

To get around these problems, Wayne suggested emendations to the *TeXbook* versions of these macros which are broadly the same as those described by Michael: the central idea (also suggested by Donald Knuth in a letter to Michael) is to avoid collecting the text in an `hbox` in the first place, but instead to collect it in a `vbox` whose width is (virtually) infinite. The text is therefore typeset in unrestricted horizontal mode, as a paragraph consisting of a single long line. Later, there is an extra level of unboxing to be done: we have to unpack the `vbox`, as well as the `hboxes` inside it, but that is not too hard. For details, we refer you to Michael's article, where the issues are clearly explained.²⁹ 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.³⁰ 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

²⁷Michael Downes, 'Line Breaking in `\unhboxed` Text', *TUGboat* **11** (1990), pp. 605–612.

²⁸See *The TeXbook*, p. 455 (editions after January 1990).

²⁹Wayne 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.

³⁰'Line Breaking', p. 610.

here, since we feel that the problem is exiguous, and `reledmac` is quite baroque enough already. If you think you are having this problem, look up Michael's solutions.

One more thing; we set `\leftskip` and `\rightskip` to zero. This has the effect of neutralizing any such skips which may apply to the main text (cf. XII.6.2 p. 155 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.

```

2561 \newcommand*{\paravfootnote}[2]{%
2562   \csuse{Xbeforeinserting@#1}%
2563   \insert\csname #1footins\endcsname
2564   \bgroup
2565     \csuse{Xnotefontsize@#1}
2566     \footplitskips
2567     \setbox0=\vbox{\hsize=\maxdimen%
2568       \let\bidirTL@everypar\@empty%
2569       \insert@Xtxtbeforenotes{#1}%
2570       \noindent\csuse{Xhooknote@#1}%
2571       \csname #1footfmt\endcsname #2{#1}}%
2572     \setbox0=\hbox{\Xunvxh{0}{#1}}%
2573     \dp0=0pt
2574     \ht0=\csname #1footfudgefactor\endcsname\wd0
2575   %

```

Here we produce the contents of the footnote from box 0, and add a penalty of 0 between boxes in this insert.

```

2576   \if@RTL\noindent \leavevmode\fi\box0%
2577   \penalty0
2578 \egroup}
2579
2580 %

```

The final penalty of 0 was added here at Wayne's suggestion to avoid a weird page-breaking problem, which occurs on those occasions when \TeX attempts to split foot paragraphs. After trying out such a split (see *The TeXbook*, p. 124), \TeX inserts a penalty of -10000 here, which nearly always forces the break at the end of the whole footnote paragraph (since individual notes can't be split) even when this leads to an overfull vbox. The change above results in a penalty of 0 instead which allows, but does not force, such breaks. This penalty of 0 is later removed, after page breaks have been decided, by the `\unpenalty` macro in `\makehboxofhboxes`. So it does not affect how the footnote paragraphs are typeset (the notes still have a penalty of -10 between them, which is added by `\parafootfmt`).

`\mpparavfootnote` This version is for minipages.

```

2581 \newcommand*{\mpparavfootnote}[2]{%
2582   \global\setbox\@nameuse{mp#1footins}\vbox{%
2583     \unvbox\@nameuse{mp#1footins}%
2584     \csuse{Xnotefontsize@#1}

```



```

2585 \footsplitskips
2586 \setbox0=\vbox{\hsize=\maxdimen%
2587   \let\bidir@RTL@everypar\@empty%
2588   \insert@Xtxtbeforenotes{#1}%
2589   \noindent\color@begingroup%
2590   \csuse{Xbhooknote@#1}%
2591   \csname #1footfmt\endcsname #2{#1}\color@endgroup}%
2592 \setbox0=\hbox{\Xunvxh{0}{#1}}%
2593 \dp0=\z@
2594 \ht0=\csname #1footfudgefactor\endcsname\wd0
2595 \box0
2596 \penalty0
2597 }}
2598
2599 %

```

\Xunvxh Here is (modified) Michael's definition of `\unvxh`, used above. Michael's macro also takes care to remove some unwanted penalties and glue that \TeX automatically attaches to the end of paragraphs. When \TeX finishes a paragraph, it throws away any remaining glue, and then tacks on the following items: a `\penalty` of 10000, a `\parfillskip` and a `\rightskip` (*The TeXbook*, pp. 99–100). `\unvxh` cancels these unwanted paragraph-final items using `\unskip` and `\unpenalty`.

```

2600 \newcommand*{\Xunvxh}[2]{%
2601   \setbox0=\vbox{\unvbox#1%
2602     \global\setbox1=\lastbox}%
2603   \unhbox1
2604   \unskip           % remove \rightskip,
2605   \unskip           % remove \parfillskip,
2606   \unpenalty        % remove \penalty of 10000,
2607   \hskip\csuse{Xafternote@#2}\relax}% but add the glue to go between the
notes
2608
2609 %

```

\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).

```

2610 \newcommand*{\parafootfmt}[4]{%
2611   \Xinsertparafootsep{#4}%
2612   \ledsetnormalparstuff@common%
2613   \printlinefootnote{#1}{#4}%
2614   \print@lemma{#1}{#2}{#4}%
2615   \csuse{Xwrapcontent@#4}{#3}%
2616   \penalty-10 }
2617 %

```

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.

```

2618 \newcommand*{\parafootgroup}[1]{%
2619   \hsize=\expandafter\dimexpr\csuse{Xwidth@#1}\relax%
2620   \unvbox\csname #1footins\endcsname
2621   \ifcsstring{Xragged@#1}{L}{\RaggedLeft}{}%
2622   \ifcsstring{Xragged@#1}{R}{\RaggedRight}{}%
2623   \makehboxofhboxes
2624   \setbox0=\hbox{\unhbox0 \removehboxes}%
2625   \csuse{Xbhookgroup@#1}%
2626   \csuse{Xnotefontsize@#1}%
2627   \unhbox0\par%
2628   \global\hsize=\old@hsize%
2629 }%
2630
2631 %

```

`\mpparafootgroup` The minipage version.

```

2632 \newcommand*{\mpparafootgroup}[1]{%
2633   \setXnoteswidthliketwocolumns@{#1}%
2634   \vskip\skip\@nameuse{mp#1footins}
2635   \ifl@dpairing\ifparledgroup%
2636     \leavevmode\marks\parledgroup@{begin}%
2637     \marks\parledgroup@series{#1}%
2638     \marks\parledgroup@type{Xfootnote}%
2639   \fi\fi\normalcolor
2640   \ifparledgroup%
2641     \ifl@dpairing%
2642     \else%
2643       \setXnoteswidthliketwocolumns@{#1}%
2644       \setXnotespositionliketwocolumns@{#1}%
2645       \print@Xfootnoterule{#1}%
2646     \fi%
2647   \else%
2648     \setXnoteswidthliketwocolumns@{#1}%
2649     \setXnotespositionliketwocolumns@{#1}%
2650     \print@Xfootnoterule{#1}%
2651   \fi%
2652   \unvbox\csname mp#1footins\endcsname
2653   \ifcsstring{Xragged@#1}{L}{\RaggedLeft}{}%

```

```

2654 \ifcsstring{Xragged@#1}{R}{\RaggedRight}{}%
2655 \makehboxofhboxes
2656 \setbox0=\hbox{\unhbox0 \removehboxes}%
2657 \csuse{Xbhookgroup@#1}%
2658 \csuse{Xnotefontsize@#1}%
2659 \nottoggle{Xparindent@#1}{\parindent=\z@}{}%
2660 \unhbox0\par}}
2661
2662 %

```

And finally, the two macros which are required to transform the long horizontal box stored in the insert' box to a printable text.

```

\makehboxofhboxes 63 \newcommand*{\makehboxofhboxes}{\setbox0=\hbox{}}%
\removehboxes 64 \loop
2665 \unpenalty
2666 \setbox2=\lastbox
2667 \ifhbox2
2668 \setbox0=\hbox{\box2\unhbox0}%
2669 \repeat}
2670
2671 \newcommand*{\removehboxes}{\setbox0=\lastbox
2672 \ifhbox0{\removehboxes}\unhbox0 \fi}
2673
2674 %

```

Insertion of the footnotes separator The command `\Xinsertparafootsep{<series>}` must be called at the beginning of `\parafootftm`.

```

\prevpage@num 75 \newcommand{\Xinsertparafootsep}[1]{%
\Xinsertparafootsep 76 \ifledRcol{%
2677 \ifnumequal{\csuse{#1prevpage@numR}}{\page@numR}%
2678 {\ifcsdef{prevline#1}% Be sur \prevline#1 exists.
2679 {\ifnumequal{\csuse{prevline#1}}{\line@numR}%
2680 {\ifcsemtty{Xsymlinenum@#1}{\csuse{Xparafootsep@#1}}{}}}%
2681 {\csuse{Xparafootsep@#1}}}%
2682 }%
2683 {\csuse{Xparafootsep@#1}}}%
2684 }%
2685 {}%
2686 \global\csname #1prevpage@numR\endcsname=\page@numR%
2687 \else%
2688 \ifnumequal{\csuse{#1prevpage@num}}{\page@num}%
2689 {\ifcsdef{prevline#1}% Be sur \prevline#1 exists.
2690 {\ifnumequal{\csuse{prevline#1}}{\line@num}%
2691 {\ifcsemtty{Xsymlinenum@#1}{\csuse{Xparafootsep@#1}}{}}}%
2692 {\csuse{Xparafootsep@#1}}}%
2693 }%

```

```

2694     {\csuse{Xparafootsep@#1}}}%
2695   }%
2696   {}%
2697   \global\csname #1prevpage@num\endcsname=\page@num%
2698   \fi%
2699 }
2700 %

```

XII.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 into a number (#2) of columns, each with a space (#3) between the top baseline and the top of the `\vbox`. The `\Xrigidbalance` `\dosplits` `\splitoff` `\@h` `\@k` `\rigidbalance` macro is taken from *The TeXbook*, p. 397, with a slight change to the syntax of the arguments so that they do not depend on white space. Note also the extra unboxing in `\splitoff`, which allows the new `\vbox` to have its natural height as it goes into the alignment.

The \TeX `\line` macro has no relationship to the TeX `\line`. The \TeX equivalent is `\@@line`.

We do not call directly `\rigidbalance`, but we call `\Xrigidbalance` for critical notes and `\rigidbalanceX` for familiar notes. Both of them call `\rigidbalance`.

```

2701 \newcount\@k \newdimen\@h
2702 \newcommand*\Xrigidbalance}[3]{%
2703   \hsize=\expandafter\dimexpr\csuse{Xwidth@}\@currentseries\relax%
2704   \rigidbalance{#1}{#2}{#3}%
2705 }%
2706
2707 \newcommand*\rigidbalanceX}[3]{%
2708   \hsize=\expandafter\dimexpr\csuse{widthX@}\@currentseries\relax%
2709   \rigidbalance{#1}{#2}{#3}%
2710 }%
2711
2712 \newcommand*\rigidbalance}[3]{%
2713   \setbox0=\box#1 \@k=#2 \@h=#3%
2714   \@@line{\splittopskip=\@h \vbadness=\@M \hfilneg
2715     \valign{##\vfil\cr\dosplits}}}%
2716
2717 \newcommand*\dosplits{\ifnum\@k>0 \noalign{\hfil}\splitoff
2718   \global\advance\@k-1\cr\dosplits\fi}
2719
2720 \newcommand*\splitoff{\dimen0=\ht0
2721   \divide\dimen0 by\@k \advance\dimen0 by\@h
2722   \setbox2 \vsplit0 to \dimen0
2723   \unvbox2 }
2724
2725 %

```

Three columns

```
\Xarrangement@threecol 2726 \newcommand*{\Xarrangement@threecol}[1]{%
2727 \csgdef{series@display#1}{threecol}
2728 \expandafter\let\csname v#1footnote\endcsname=\threecolvfootnote
2729 \expandafter\let\csname #1footfmt\endcsname=\threecolfootfmt
2730 \expandafter\let\csname #1footgroup\endcsname=\threecolfootgroup
2731 \dimen\csname #1footins\endcsname=\csuse{Xmaxhnotes@#1}%
2732 \skip\csname #1footins\endcsname=\csuse{Xbeforenotes@#1}%
2733 \advance\skip\csname #1footins\endcsname by\csuse{Xafterrule@#1}%
2734 \threecolfootsetup{#1}
2735 %
```

The additional setup for minipages.

```
2736 \ifnoledgroup@else
2737 \expandafter\let\csname mpv#1footnote\endcsname=\mpnormalvfootnote
2738 \expandafter\let\csname mp#1footgroup\endcsname=\mpthreecolfootgroup
2739 \skip\csname mp#1footins\endcsname=\csuse{Xbeforenotes@#1}%
2740 \advance\skip\csname mp#1footins\endcsname by\csuse{Xafterrule@#1}%
2741 \mpthreecolfootsetup{#1}
2742 \fi
2743 }
2744 %
2745 %
```

The `\footstart` and `\footnoterule` macros for these notes assume the normal values (XII.6.2 p. 155 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.

```
2746 \newcommand*{\threecolfootsetup}[1]{%
2747 \count\csname #1footins\endcsname 333
2748 \csxdef{default@#1footins}{333}%Use this to confine the notes to one
side only
2749 \multiply\dimen\csname #1footins\endcsname \thr@@}
2750 %
```

`\mpthreecolfootsetup` The setup for minipages.

```

2751 \newcommand*\mpthreecolfootsetup}[1]{%
2752   \count\csname mp#1footins\endcsname 333
2753   \multiply\dimen\csname mp#1footins\endcsname \thr@@}
2754
2755 %

```

`\threecolvfootnote` `\threecolvfootnote` is the `\vfootnote` command for three-column notes. The call to `\Xnotefontsize@<s>` ensures that the `\splittopskip` and `\splitmaxdepth` take their values from the right `\strutbox`: the one used in a footnotes. Note especially the importance of temporarily reducing the `\hsize` to 0.3 of its normal value. This determines the widths of the individual columns. So if the normal `\hsize` is, say, 10 cm, then each column will be $0.3 \times 10 = 3$ cm wide, leaving a gap of 1 cm spread equally between columns (i.e., .5 cm between each).

The arguments are #1 the note series letter and #1 the full text of the note (including numbers, lemma and text).

```

2756 \notbool{parapparatus@}\newcommand*\newcommand*\threecolvfootnote}[2]{%
2757   \csuse{Xbeforeinserting@#1}%
2758   \insert\csname #1footins\endcsname\bgroup%
2759   \hsize=\expandafter\dimexpr\csuse{Xwidth@#1}\relax%
2760   \noindent\csuse{Xbhooknote@#1}%
2761   \csuse{Xnotefontsize@#1}%
2762   \footssplitskips%
2763   \csname #1footfmt\endcsname #2{#1}\egroup}
2764 %

```

`\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.

```

2765 \notbool{parapparatus@}\newcommand*\newcommand*\threecolfootfmt}[4]{%
2766   \normal@pars%
2767   \hsize \csuse{Xsizethreecol@#4}%
2768   \nottoggle{Xparindent@#4}\parindent=\z@}%
2769   \tolerance=5000%
2770   \hangindent=\csuse{Xhangindent@#4}%
2771   \everypar{\hangindent=\csuse{Xhangindent@#4}}%
2772   \@tempdima=\parindent%
2773   \csuse{Xcolalign@#4}%
2774   \parindent=\@tempdima%
2775   \strut{%
2776   \hspace{\parindent}%
2777   \printlinefootnote{#1}{#4}%
2778   }%
2779   \print@lemma{#1}{#2}{#4}%
2780   \csuse{Xwrapcontent@#4}{#3}%
2781   \strut\par\allowbreak}
2782 %

```

`\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.

```

2783 \newcommand*{\threecolfootgroup}[1]{%
2784   \csuse{Xbhookgroup@#1}\par%
2785   \splittopskip=\ht\strutbox
2786   \expandafter
2787   \Xrigidbalance\csname #1footins\endcsname \thr@@ \splittopskip}
2788   %

```

`\mpthreecolfootgroup` The setup for minipages.

```

2789 \newcommand*{\mpthreecolfootgroup}[1]{%
2790   \vskip\skip\@nameuse{mp#1footins}
2791   \ifl@dpairing\ifparledgroup%
2792     \leavevmode\marks\parledgroup@{begin}%
2793     \marks\parledgroup@series{#1}%
2794     \marks\parledgroup@type{Xfootnote}%
2795   \fi\fi\normalcolor
2796   \ifparledgroup%
2797     \ifl@dpairing%
2798       \else%
2799         \setXnoteswidthliketwocolumns@{#1}%
2800         \setXnotespositionliketwocolumns@{#1}%
2801         \print@Xfootnoterule{#1}%
2802       \fi%
2803     \else%
2804       \setXnoteswidthliketwocolumns@{#1}%
2805       \setXnotespositionliketwocolumns@{#1}%
2806       \print@Xfootnoterule{#1}%
2807     \fi%
2808   \csuse{Xbhookgroup@#1}\par%
2809   \splittopskip=\ht\strutbox
2810   \expandafter
2811   \Xrigidbalance\csname mp#1footins\endcsname \thr@@ \splittopskip}}
2812   %
2813   %

```

Two columns

`\Xarrangement@twocol`₁₄ `\newcommand*{\Xarrangement@twocol}[1]{%`

```

2815 \csgdef{series@display#1}{twocol}
2816 \expandafter\let\csname v#1footnote\endcsname=\twocolvfootnote
2817 \expandafter\let\csname #1footfmt\endcsname=\twocolfootfmt
2818 \expandafter\let\csname #1footgroup\endcsname=\twocolfootgroup
2819 \dimen\csname #1footins\endcsname=\csuse{Xmaxhnotes@#1}%
2820 \skip\csname #1footins\endcsname=\csuse{Xbeforenotes@#1}%
2821 \advance\skip\csname #1footins\endcsname by\csuse{Xafterrule@#1}%
2822 \twocolfootsetup{#1}
2823 %

```

The additional setup for minipages.

```

2824 \ifnoledgroup@else
2825 \expandafter\let\csname mpv#1footnote\endcsname=\mpnormalvfootnote
2826 \expandafter\let\csname mp#1footgroup\endcsname=\mptwocolfootgroup
2827 \skip\csname mp#1footins\endcsname=\csuse{Xbeforenotes@#1}%
2828 \advance\skip\csname mp#1footins\endcsname by\csuse{Xafterrule@#1}%
2829 \mptwocolfootsetup{#1}
2830 \fi
2831 }
2832 %
2833 %

```

`\twocolfootsetup` Here is a series of macros which are very similar to their three-column counterparts. In this case, each note is assumed to contribute only a half a line of text. And the notes are set in columns giving a gap between them of one tenth of the `\hsz`.

```

\twocolfootfmt
\twocolfootgroup
2834 \newcommand*{\twocolfootsetup}[1]{%
2835 \count\csname #1footins\endcsname 500
2836 \csxdef{default@#1footins}{500}%
2837 \multiply\dimen\csname #1footins\endcsname \tw@}
2838 %

2839 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\twocolvfootnote}[2]{%
2840 \csuse{Xbeforeinserting@#1}%
2841 \insert\csname #1footins\endcsname\bgroup%
2842 \hsz=\expandafter\dimexpr\csuse{Xwidth@#1}\relax%
2843 \noindent\csuse{Xhooknote@#1}%
2844 \csuse{Xnotefontsize@#1}%
2845 \footsplitskips%
2846 \csname #1footfmt\endcsname #2{#1}\egroup}
2847 %

2848 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\twocolfootfmt}[4]{% 4th
arg is optional, for backward compatibility
2849 \normal@pars%
2850 \hsz \csuse{Xhsizetwocol@#4}%
2851 \nottoggle{Xparindent@#4}{\parindent=\z@}{}%
2852 \tolerance=5000%
2853 \hangindent=\csuse{Xhangindent@#4}%
2854 \everypar{\hangindent=\csuse{Xhangindent@#4}}%

```



```

2855 \@tempdima=\parindent%
2856 \csuse{Xcolalign@#4}%
2857 \parindent=\@tempdima%
2858 \strut{%
2859 \hspace{\parindent}%
2860 \printlinefootnote{#1}{#4}%
2861 }%
2862 \print@lemma{#1}{#2}{#4}%
2863 \csuse{Xwrapcontent@#4}{#3}%
2864 \strut\par\allowbreak}
2865 %

2866 \newcommand*{\twocolfootgroup}[1]{%
2867 \csuse{Xbhookgroup@#1}\par%
2868 \splittopskip=\ht\strutbox
2869 \expandafter
2870 \Xrigidbalance\csname #1footins\endcsname \tw@ \splittopskip}
2871
2872 %

```

`\mptwocolfootsetup` The versions for minipages.

`\mptwocolfootgroup`

```

2873 \newcommand*{\mptwocolfootsetup}[1]{%
2874 \count\csname mp#1footins\endcsname 500
2875 \multiply\dimen\csname mp#1footins\endcsname \tw@}
2876 %

2877 \newcommand*{\mptwocolfootgroup}[1]{%
2878 \vskip\skip\@nameuse{mp#1footins}
2879 \ifl@dpairing\ifparledgroup%
2880 \leavevmode\marks\parledgroup@{begin}%
2881 \marks\parledgroup@series{#1}%
2882 \marks\parledgroup@type{Xfootnote}%
2883 \fi\fi\normalcolor
2884 \ifparledgroup%
2885 \ifl@dpairing%
2886 \else%
2887 \setXnoteswidthliketwocolumns@{#1}%
2888 \setXnotespositionliketwocolumns@{#1}%
2889 \print@Xfootnoterule{#1}%
2890 \fi%
2891 \else%
2892 \setXnoteswidthliketwocolumns@{#1}%
2893 \setXnotespositionliketwocolumns@{#1}%
2894 \print@Xfootnoterule{#1}%
2895 \fi%
2896 \csuse{Xbhookgroup@#1}\par%
2897 \splittopskip=\ht\strutbox
2898 \expandafter
2899 \Xrigidbalance\csname mp#1footins\endcsname \tw@ \splittopskip}}

```

```
2900
2901 %
```

XII.7 Critical notes presentation

Here, we define some commons macro which are used in order to print a critical notes, that is a note with 1) line number 2) lemma 3) lemma separator 4) text associated to the lemma.

XII.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.

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.

```
2902 \def\endashchar{\textnormal{--}}
2903
2904 \newcommand*{\fullstop}{\textnormal{.}}
2905 \def\Xsublinesep@side{\fullstop}
2906
2907 \newcommand*{\rbracket}{\textnormal{%
2908   \csuse{text\csuse{footnote@lang}}{%
2909     \ifluatex%
2910       \ifdefstring{footnote@luatextextdir}{TRT}{\thinspace[]}{\thinspace
2911     }}%
2912     \else%
2913     \thinspace[]%
2914     \fi}%
2915   }}
2916
2917 %
```

XII.7.2 Pstart number in footnote

`\printpstart` The `\printpstart` macro prints the pstart number for a note.

```
2918 \newcommand{\printpstart}[0]{%
```

```

2919 \ifboolexpr{bool{l@dpairing} or bool{l@dprintingpages} or bool{
l@dprintingcolumns}}{%
2920 \ifledRcol%
2921 \thepstartR%
2922 \else%
2923 \thepstartL%
2924 \fi%
2925 }{%
2926 \thepstart%
2927 }%
2928 }
2929 %

```

XII.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).

```

2930 %
2931 \newcommand{\print@lemma}[3]{%
2932 \bgroup%
2933 \nottoggle{Xlemmadisablefontselection@#3}%
2934 {\select@lemmafnt#1|}%
2935 }%
2936 \bgroup%
2937 \csuse{Xlemmafnt@#3}%Deprecated
2938 \csuse{Xwraplemma@#3}{#2}%
2939 \egroup%
2940 \group%
2941 \iftoggle{nosep@}{%
2942 \hskip\csuse{Xinplaceoflemmaseparator@#3}%
2943 \relax%
2944 }%
2945 {\ifcsemt{Xlemmaseparator@#3}%
2946 {%
2947 \hskip\csuse{Xinplaceoflemmaseparator@#3}%
2948 \relax%
2949 }%
2950 {%
2951 \nobreak%
2952 \hskip\csuse{Xbeforelemmaseparator@#3}%
2953 \csuse{Xlemmaseparator@#3}%
2954 \hskip\csuse{Xafterlemmaseparator@#3}%
2955 \relax%
2956 }%
2957 }%
2958 }%
2959 %

```

XII.7.4 Line number printing

`\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`.

```

2960 \newcommand{\printlinefootnote}[2]{%
2961   \l@dp@rsefootspec#1|%
2962   \iftoggle{Xnumberonlyfirstintwolines@#2}{%
2963     \edef\lineinfo@{\l@dparsedstartline - \l@dparsedstartsub - \
2964       \l@dparsedendline - \l@dparsedendsub}%
2965     }%
2966     {%
2967       \edef\lineinfo@{\l@dparsedstartline - \l@dparsedstartsub}%
2968     }%
2969     \iftoggle{nonum@}{%Try if the line number must printed for this specific
2970       not (by default, yes)
2971       \hspace{\csuse{Xinplaceofnumber@#2}}%
2972     }%
2973     {%
2974       \iftoggle{Xnonumber@#2}%Try if the line number must printed (by
2975       default, yes)
2976       {%
2977         \hspace{\csuse{Xinplaceofnumber@#2}}%
2978       }%
2979       {%
2980         \iftoggle{Xnumberonlyfirstinline@#2}% If for this series the
2981         line number must be printed only in the first time.
2982         {%
2983           \ifcsdef{prevline#2}%
2984           {%Be sure the \prevline exists.
2985           \ifcsequal{prevline#2}{\lineinfo@}%Try it
2986           {%
2987             \ifcsequal{Xsymlinenum@#2}%
2988             {%
2989               \hspace{\csuse{Xinplaceofnumber@#2}}%
2990             }%
2991             {\printlinefootnotearea{#2}}%
2992           }%
2993         }%
2994         {%
2995           \printlinefootnotearea{#1}{#2}%
2996         }%
2997       }%
2998     }%

```

```

2999         \printlinefootnotearea{#1}{#2}%
3000     }%
3001     \csxdef{prevline#2}{\lineinfo@}%
3002 }%
3003 }%
3004 }%
3005 }%
3006 }
3007 %

```

\printsymlinefootnotearea This macro prints the space before the line symbol, changes the font, when prints the line symbol and the space after it.

```

3008 \newcommand{\printsymlinefootnotearea}[1]{%
3009     \hspace{\csuse{Xbeforesymlinenum@#1}}%
3010     \csuse{Xnotenumfont@#1}%
3011     \ifdimequal{\csuse{Xboxsymlinenum@#1}}{\z@}%
3012         {\csuse{Xsymlinenum@#1}}%
3013         {\hbox to \csuse{Xboxsymlinenum@#1}%
3014             {\csuse{Xsymlinenum@#1}\hfill}}%
3015     }%
3016     \hspace{\csuse{Xaftersymlinenum@#1}}%
3017 }%
3018 %

```

\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.)

```

3019 \newcommand{\printlinefootnotearea}[2]{%
3020     \printXbeforenumber{#2}%
3021     \csuse{Xnotenumfont@#2}%
3022     \boxfootnotenumbers{#1}{#2}%
3023     \printXafternumber{#2}%
3024 }%
3025 %

```

\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.

```

3026 \newcommand{\boxfootnotenumbers}[2]{%
3027     \ifdimequal{\csuse{Xboxlinenum@#2}}{\z@}{%
3028         \printlinefootnotenumbers{#1}{#2}%
3029     }%
3030     {%
3031         \hbox to \csuse{Xboxlinenum@#2}%
3032         {%
3033             \IfSubStr{RC}{\csuse{Xboxlinenumalign@#2}}{\hfill}{}%

```

```

3034 \printlinefootnotenumbers{#1}{#2}%
3035 \IfSubStr{LC}{\csuse{Xboxlinenumalign@#2}}{\hfill}{}%
3036 }%
3037 }%
3038 }%
3039 %

```

\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.

```

3040 \newcommand{\printlinefootnotenumbers}[2]{%
3041 \xdef\@currentseries{#2}%
3042 \ifboolexpr{%
3043 (togl{Xpstart@#2} and bool{numberpstart})%
3044 or togl{Xpstarteverytime@#2}}%
3045 {\printpstart}{}%
3046 \iftoggle{Xstanza@#2}{%
3047 \ifnumberstanza%
3048 \printstanza%
3049 \csuse{Xstanzaseparator@#2}%
3050 \fi%
3051 }{}%
3052 \iftoggle{Xonlypstart@#2}{%}{%
3053 \csuse{Xtxtbeforenumber@#2}%
3054 \printlines#1||\ifledRcol@{\@Rlineflag\fi}%
3055 }%
3056 %

```

\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.)

```

3057 \newcommand{\printXbeforenumber}[1]{%
3058 \hspace{\csuse{Xbeforenumber@#1}}%
3059 }%
3060 %

```

\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

```

3061 \newcommand{\printXafternumber}[1]{%
3062 \iftoggle{Xnonbreakableafternumber@#1}{\nobreak}{}%
3063 \hspace{\csuse{Xafternumber@#1}}%
3064 }%
3065 %

```

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. 90: the starting page, line, and sub-line numbers, followed by the ending page, line, and sub-line numbers, and then the font specifier for the lemma.

edmac' creator have defined six boolean in order to know which component of line number description we have to print:

- \ifl@d@pnum for page numbers;
- \ifl@d@ssub for starting sub-line;
- \ifl@d@elin for ending line;
- \ifl@d@esl for ending sub-line; and
- \ifl@d@dash for the dash between the starting and ending groups.

There 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.

```

\ifl@d@pnum3066 \newif\ifl@d@pnum
\ifl@d@ssub3067 \newif\ifl@d@ssub
\ifl@d@elin3068 \newif\ifl@d@elin
\ifl@d@esl3069 \newif\ifl@d@esl
\ifl@d@dash3070 \newif\ifl@d@dash
\ifl@d@Xtwolines3071 \newif\ifl@d@Xtwolines%
\ifl@d@Xmorethantwolines3072 \newif\ifl@d@Xmorethantwolines%
3073 %

\l@dp@rsefootsspec \l@dp@rsefootsspec parses lines specification and defines macros which hold the nu-
\l@dparsedstartpage \l@dparsedstartpage{#1}%
\l@dparsedstartline \printlines #1 | #2 | #3 | #4 | #5 | #6 | #7
\l@dparsedstartsub \printlines start-page | line | subline | end-page | line | subline | fontflag
\l@dparsedendpage3074 \def\l@dp@rsefootsspec#1|#2|#3|#4|#5|#6|#7|{%
\l@dparsedendline3075 \gdef\l@dparsedstartpage{#1}%
\l@dparsedendsub3076 \gdef\l@dparsedstartline{#2}%
3077 \gdef\l@dparsedstartsub{#3}%
3078 \gdef\l@dparsedendpage{#4}%
3079 \gdef\l@dparsedendline{#5}%
3080 \gdef\l@dparsedendsub{#6}%
3081 }
3082 %

```

Initialise the several number value macros.

```

3083 \def\l@dparsedstartpage{0}%
3084 \def\l@dparsedstartline{0}%
3085 \def\l@dparsedstartsub{0}%
3086 \def\l@dparsedendpage{0}%

```

```

3087 \def\l@dparsedendline{0}%
3088 \def\l@dparsedendsub{0}%
3089
3090 %

```

\setprintlines The macro `\setprintlines` does the work of deciding what numbers should be printed. Its arguments are the same as the first 6 of `\printlines`.

```

3091 \newcommand*\setprintlines}[6]{%
3092   \l@d@pnumfalse \l@d@dashfalse
3093 %

```

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

```

3094   \ifbypage@
3095     \ifnum#4=#1 \else
3096       \l@d@pnumtrue
3097       \l@d@dashtrue
3098     \fi
3099   \fi
3100 %

```

We print the ending line number if: (1) we are printing the ending page number, or (2) it is different from the starting line number.

```

3101   \ifl@d@pnum \l@d@elintrue \else \l@d@elinfalse \fi
3102   \ifnum#2=#5 \else
3103     \l@d@elintrue
3104     \l@d@dashtrue
3105   \fi
3106 %

```

We print the starting sub-line if it is nonzero.

```

3107   \l@d@ssubfalse
3108   \ifnum#3=0 \else
3109     \l@d@ssubtrue
3110   \fi
3111 %

```

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.

```

3112   \l@d@eslfalse
3113   \ifnum#6=0 \else
3114     \ifnum#6=#3
3115       \ifl@d@elin \l@d@esltrue \else \l@d@eslfalse \fi
3116     \else
3117       \l@d@esltrue
3118       \l@d@dashtrue
3119     \fi
3120   \fi%
3121 %

```


However, if the `\Xtwolines` is set for the current series, we do not print the last line number.

```

3122 \ifl@d@dash%
3123 \ifbool{expr{togl{fulllines@} or test{\ifcempty{Xtwolines@
@currentseries}}}}%
3124 {}%
3125 {%
3126 \setistwofollowinglines{#1}{#2}{#4}{#5}%
3127 \ifbool{expr{
3128 (%
3129 togl {Xtwolinesbutnotmore@ \@currentseries}%
3130 and not%
3131 (%
3132 bool {istwofollowinglines@}%
3133 )%
3134 )%
3135 or%
3136 (%
3137 (not test{\ifnumequal{#1}{#4}})%
3138 and togl{Xtwolinesonlyinsamepage@ \@currentseries}%
3139 )%
3140 }%
3141 {}%
3142 {%
3143 \l@d@dashfalse%
3144 \l@d@Xtwolinestrue%
3145 \l@d@elinfalse%
3146 \l@d@eslfalse%
3147 \ifcempty{Xmorethantwolines@ \@currentseries}%
3148 {}%
3149 {\ifistwofollowinglines@ \else%
3150 \l@d@Xmorethantwolinestrue%
3151 \fi%
3152 }%
3153 }%
3154 }%
3155 \fi%
3156 %

```

End of `\setprintlines`.

```

3157 }%
3158 %

```

`\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.

```

3159 \newif\ifistwofollowinglines@%
3160 \newcommand{\setistwofollowinglines}[4]{%
3161   \ifcsdef{lastlinenumberon@#1}%
3162     {\numdef{\tmp}{\csuse{lastlinenumberon@#1}}}%
3163     {\numdef{\tmp}{0}}%
3164   \istwofollowinglines@false%
3165   \ifnumequal{#4-#2}{1}%
3166     {\istwofollowinglines@true}%
3167     {\ifbypage@%
3168       \ifnumequal{#3-#1}{1}%
3169       {%
3170         \ifnumequal{#2}{\tmp}%
3171         {\ifnumequal{#4}{1}{\istwofollowinglines@true}{}}%
3172         {}%
3173       }%
3174       {}%
3175     \fi%
3176   }%
3177 }%
3178 %

```

`\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

```

3179 \def\printlines#1|#2|#3|#4|#5|#6|#7|#8|{%
3180   \begingroup%
3181 %

```

If we use Lua_T_EX, ensure we use good text's direction.

```

3182 \ifluatex%
3183   \edef\@tmp{\the\textdir}%
3184   \ifdefstring{\@tmp}{TLT}{\textdir TLT}%Test in order to prevent
spurious space (bug #397)

```

```

3185 \fi%
3186 %

```

Decide which part of line number components we will print.

```

3187 \setprintlines{#1}{#2}{#3}{#4}{#5}{#6}%
3188 %

```

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.

```

3189 \ifdimequal{\csuse{Xboxstartlinenum@\@currentseries}}{0pt}%
3190 {\bgroup}%
3191 {\leavevmode\hbox to \csuse{Xboxstartlinenum@\@currentseries}\bgroup\
hfill}%
3192 \ifl@d@pnum%
3193 \wrap@edcrossref{\@this@crossref@start}{#1}%
3194 \csuse{Xsublinesep@\@currentseries}%
3195 \fi%
3196 \wrap@edcrossref{\@this@crossref@start}{%
3197 \linenumrep{#2}%
3198 \iftoggle{Xlineflag@\@currentseries}{#8}{}}%
3199 }%
3200 \ifl@d@ssub%
3201 \csuse{Xsublinesep@\@currentseries}%
3202 \wrap@edcrossref{\@this@crossref@start}{\sublinenumrep{#3}}%
3203 \fi
3204 \egroup%
3205 %

```

Then print the dash + end line number, or the range symbol.

```

3206 \ifdimequal{\csuse{Xboxendlinenum@\@currentseries}}{0pt}%
3207 {\bgroup}%
3208 {\hbox to \csuse{Xboxendlinenum@\@currentseries}\bgroup}%
3209 \ifl@d@Xtwolines%
3210 \ifl@d@Xmorethantwolines%
3211 \csuse{Xmorethantwolines@\@currentseries}%
3212 \else%
3213 \csuse{Xtwolines@\@currentseries}%
3214 \fi%
3215 \else%
3216 \ifl@d@dash%
3217 \ifdefined\linerangesep@%
3218 \linerangesep@%
3219 \else%
3220 \csuse{Xlinerangeseparator@\@currentseries}%
3221 \fi%
3222 \fi%
3223 \ifl@d@pnum%

```

```

3224 \wrap@edcrossref{\@this@crossref@end}{#4}%
3225 \csuse{Xsublinesep@\@currentseries}%
3226 \fi%
3227 \ifl@d@elin%
3228 \wrap@edcrossref{\@this@crossref@end}{%
3229 \linenumrep{#5}%
3230 \iftoggle{Xlineflag@\@currentseries}{#8}{}}%
3231 }%
3232 \fi%
3233 \ifl@d@esl%
3234 \ifl@d@elin%
3235 \csuse{Xsublinesep@\@currentseries}%
3236 \fi%
3237 \wrap@edcrossref{\@this@crossref@end}{\sublinenumrep{#6}}%
3238 \fi%
3239 \fi%
3240 \ifdimequal{\csuse{Xboxendlinenum@\@currentseries}}{0pt}%
3241 {}%
3242 {\hfill}%Prevent underfull hbox
3243 \egroup%
3244 \endgroup%
3245 }%
3246 %

```

XIII Familiar footnotes

XIII.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` package and perhaps by other footnote packages. That is why we use `\providecommand` and not `\newcommand`.

`\multfootsep`

```

3247 \providecommand*\multiplefootnotemarker{3sp}
3248 \providecommand*\multfootsep{\textsuperscript{\normalfont,}}
3249
3250 %

```

`\m@mmf@prepare` A pair of self-cancelling kerns. This may have been defined in the `memoir` class.

```

3251 \providecommand*\m@mmf@prepare}{%
3252 \kern-\multiplefootnotemarker
3253 \kern\multiplefootnotemarker\relax}
3254 %

```

\m@mmf@check This may have been defined in the memoir class. If it recognises the last kern as \multiplefootnotemarker it typesets \multfootsep.

```

3255 \providecommand*\m@mmf@check}{%
3256 \ifdim\lastkern=\multiplefootnotemarker\relax
3257 \edef\x@sf{\the\spacefactor}%
3258 \unkern
3259 \multfootsep
3260 \spacefactor\x@sf\relax
3261 \fi}
3262 %
3263 %

```

We have to modify \@footnotetext and \@footnotemark. However, if memoir is used the modifications have already been made.

```

3264 \@ifclassloaded{memoir}{}{%
3265 %

```

\@footnotetext Add \m@mmf@prepare at the end of \@footnotetext.

```

3266 \apptocmd{\@footnotetext}{\m@mmf@prepare}{}{}
3267 %

```

\@footnotemark Modify \@footnotemark to cater for adjacent \footnotes.

```

3268 \patchcmd{\@footnotemark}
3269 {\nobreak}
3270 {\m@mmf@check
3271 \nobreak
3272 }
3273 {}{}
3274 {}{}
3275 \patchcmd{\@footnotemark}
3276 {\@makefnmark}
3277 {\@makefnmark
3278 \m@mmf@prepare
3279 }
3280 {}{}
3281 %

```

Finished the modifications for the non-memoir case.

```

3282 }
3283 %
3284 %

```

XIII.2 Regular footnotes for numbered texts

`\l@doldold@footnotetext` In order to enable the regular `\footnotes` in numbered text we have to play around
`\@footnotetext` with its `\@footnotetext`, using different forms for when in numbered or regular text.

```

3285 \pretocmd{\@footnotetext}{%
3286   \ifnumberedpar@
3287     \edtext{}{\l@dbfnote{#1}}%
3288   \else
3289     {}{}
3290 \apptocmd{\@footnotetext}{\fi}{}{}%
3291 %

```

`\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`

```

3292 \patchcmd%
3293 {\footnote}%
3294 {\stepcounter\@mpfn}%
3295 {%
3296   \ifl@dpairing%
3297     \global\advance\footnote@reading by \@ne%
3298     \get@thisfootnote%
3299     \get@fnmark{\thisfootnote}%
3300     \ifcsdef{footnotereading\the\footnote@reading=typeset}%
3301     {\setcounter{\@mpfn}{\csuse{footnotereading\the\footnote@reading=
3302 typeset}}}%
3303     {\setcounter{\@mpfn}{\footnote@reading}}%
3304   \else%
3305     \stepcounter\@mpfn%
3306   \fi%
3307 }%
3308 {}
3309 {}
3310
3311 \newcommand{\get@thisfootnote}{%
3312   \ifl@dpairing
3313     \protected@xdef\thisfootnote{\the\footnote@reading}%
3314   \else%
3315     \protected@xdef\thisfootnote{\thefootnote}%
3316   \fi%
3317 }%
3318
3319 \newcommand{\l@dbfnote}[1]{%
3320   \get@thisfootnote%
3321   \gdef\tag{#1\relax}%

```

```

3322 \ifledRcol%
3323 \xright@appenditem{%
3324 \ifdefined\Hy@footnote@currentHref%
3325 \noexpand\def\noexpand\Hy@footnote@currentHref{\
Hy@footnote@currentHref}%
3326 \fi%
3327 \noexpand\vl@dbfnote{\expandonce\@tag}{\thisfootnote}%
3328 }%
3329 \to\inserts@listR
3330 \global\advance\insert@countR \@ne%
3331 \else%
3332 \xright@appenditem{%
3333 \ifdefined\Hy@footnote@currentHref%
3334 \noexpand\def\noexpand\Hy@footnote@currentHref{\
Hy@footnote@currentHref}%
3335 \fi%
3336 \noexpand\vl@dbfnote{\expandonce\@tag}{\thisfootnote}%
3337 }%
3338 \to\inserts@list
3339 \global\advance\insert@count \@ne%
3340 \fi
3341 \ignorespaces%
3342 }%
3343
3344 \newcommand{\get@fnmark}[1]{%
3345 \ifboolexpr{bool{l@dpairing} or bool{l@dprintingpages} or bool{
l@dprintingcolumns}}}%
3346 {%
3347 \stepcounter{footnote@typeset}%
3348 \setcounter{footnote}{\c@footnote@typeset}%
3349 \immediate\write\@mainaux{%
3350 \global\csdef{footnotereading#1=typeset}{\the\c@footnote@typeset}
%
3351 }%
3352 \def\@thefnmark{\thefootnote}%
3353 }%
3354 {%
3355 \@namedef{\@thefnmark}{#1}%
3356 }%
3357 }%
3358
3359 \newcommand{\vl@dbfnote}[2]{%
3360 \get@fnmark{#2}%
3361 \@footnotetext{#1}%
3362 }%
3363 %

```

XIII.3 Footnote formats

Some of the code for the various formats is remarkably similar to that in section ??.

The following macros generally set things up for the ‘standard’ footnote format.

`\prebodyfootmark` Two convenience macros for use by `\...@footnotemark...` macros.
`\postbodyfootmark`

```

3364 \newcommand*\prebodyfootmark{%
3365   \leavevmode
3366   \ifhmode
3367     \edef\@x@sf{\the\spacefactor}%
3368     \m@mmf@check
3369     \nobreak
3370   \fi}
3371 \newcommand\postbodyfootmark{%
3372   \m@mmf@prepare
3373   \ifhmode\spacefactor\@x@sf\fi\relax}
3374
3375 %
```

XIII.4 Footnote arrangement

XIII.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.

```

3376 \newcommandx{\arrangementX}[2][1,usedefault]{%
3377   \def\do##1{%
3378     \csname arrangementX@#2\endcsname{##1}%
3379   }%
3380   \ifstrempy{#1}%
3381     {%
3382       \dolistloop{\@series}%
3383     }%
3384     {
3385       \docsvlist{#1}%
3386     }%
3387   }%
3388   %
```

XIII.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.

```

3389 \newcommand*\normal@footnotemarkX[1]{%
3390   \prebodyfootmark
3391   \wrapped@bodyfootmarkX{#1}%
3392   \postbodyfootmark}
```



```
3393
3394 %
```

`\normalbodyfootmarkX` The `\normalbodyfootmarkX{⟨series⟩}` *really* typesets the in-text marker. The style is the normal superscript.

```
3395 \newcommand*{\normalbodyfootmarkX}[1]{%
3396   \hbox{\textsuperscript{\normalfont\@nameuse{@thefnmark#1}}}%
3397 %
```

`\normalvfootnoteX` `\normalvfootnoteX{⟨series⟩}{⟨text⟩}` does the `\insert` for the `⟨series⟩` and calls the series' `\footfmt...` to format the `⟨text⟩`.

```
3398 \notbool{parapparatus@}{\newcommand*{\newcommand}{\normalvfootnoteX}[2]{%
3399   \csuse{beforeinsertingX@#1}%
3400   \insert\@nameuse{footins#1}\bgroup
3401     \hsize=\expandafter\dimexpr\csuse{widthX@#1}\relax%
3402     \noindent\csuse{bhooknoteX@#1}%
3403     \csuse{notefontsizeX@#1}%
3404     \footsplitskips
3405     \ifl@dpairing\ifl@dpadding\else%
3406       \setnoteswidthliketwocolumnsX@{#1}%
3407     \fi\fi%
3408     \setnotesXpositionliketwocolumns@{#1}%
3409     \spaceskip=\z@skip \xspaceskip=\z@skip
3410     \csuse{\csuse{footnote@dir}}\@nameuse{footfmt#1}{#1}{#2}\egroup}
3411 %
3412 %
```

`\mpnormalvfootnoteX` The minipage version.

```
3413 \newcommand*{\mpnormalvfootnoteX}[2]{%
3414   \get@thisfootnoteX{#1}%
3415   \get@fnmarkX{#1}{\thisfootnote}%
3416   \edef\this@footnoteX@reading{\the\csname footnote#1@reading\endcsname}%
3417   \global\setbox\@nameuse{mpfootins#1}\vbox{%
3418     \unvbox\@nameuse{mpfootins#1}
3419     \noindent\csuse{bhooknoteX@#1}%
3420     \csuse{notefontsizeX@#1}%
3421     \hsize\columnwidth
3422     \@parboxrestore
3423     \color@begingroup
3424     \@nameuse{footfmt#1}{#1}{#2}\color@endgroup}}
3425 %
3426 %
```

`\normalfootfmtX` `\normalfootfmtX{⟨series⟩}{⟨text⟩}` typesets the footnote text, prepended by the marker.

```
3427 \notbool{parapparatus@}{\newcommand*{\newcommand}{\normalfootfmtX}[2]{%
3428   \ifluatex%
```

```

3429 \texdir\footnote@luatextextdir%
3430 \pardir\footnote@luatexpardir%
3431 \par%
3432 \fi%
3433 \protected@edef\@currentlabel{%
3434   \@nameuse{@thefnmark#1}%
3435 }%
3436 \ledsetnormalparstuffX{#1}%
3437 \hangindent=\csuse{hangindentX@#1}%
3438 \everypar{\hangindent=\csuse{hangindentX@#1}}%
3439 {{\csuse{notenumberfontX@#1}\wrapped@footfootmarkX{#1}}\strut%
3440   \csuse{wrapcontentX@#1}{#2}%
3441 \strut\par}}
3442
3443 %

```

`\normalfootfootmarkX` `\normalfootfootmarkX{<series>}` is called by `\normalfootfmtX` to typeset the footnote marker in the footer before the footnote text.

```

3444 \newcommand*{\normalfootfootmarkX}[1]{%
3445   \textsuperscript{\@nameuse{@thefnmark#1}}}
3446
3447 %

```

`\normalfootstartX` `\normalfootstartX{<series>}` is the `<series>` footnote starting macro used in the output routine.

```

3448 \newcommand*{\normalfootstartX}[1]{%
3449   \ifdimequal{Opt}{\prenotesX@}{}%
3450   {%
3451     \iftoggle{prenotesX@}{%
3452       \togglefalse{prenotesX@}%
3453       \skip\csname footins#1\endcsname=%
3454       \glueexpr\csuse{prenotesX@}+\csuse{afterruleX@#1}\relax%
3455     }%
3456   }%
3457   }%
3458   \vskip\skip\csname footins#1\endcsname%
3459   \leftskip=\z@
3460   \rightskip=\z@
3461   \ifl@dpairing\else%
3462     \hsize=\old@hsize%
3463   \fi%
3464   \setnoteswidthliketwocolumnsX@{#1}%
3465   \setnotesXpositionliketwocolumns@{#1}%
3466   \print@footnoteXrule{#1}%
3467 }%
3468
3469 %

```

`\normalfootnoteruleX` The rule drawn before the footnote series group.

```
3470 \let\normalfootnoteruleX=\footnoterule
3471
3472 %
```

`\normalfootgroupX` `\normalfootgroupX{<series>}` sends the contents of the `<series>` insert box to the output page without alteration.

```
3473 \newcommand*{\normalfootgroupX}[1]{%
3474   \csuse{bhookgroupX@#1}%
3475   \unvbox\@nameuse{footins#1}%
3476   \hsize=\old@hsize%
3477 }%
3478
3479 %
```

`\mpnormalfootgroupX` The minipage version.

```
3480 \newcommand*{\mpnormalfootgroupX}[1]{%
3481   \vskip\skip\@nameuse{mpfootins#1}
3482   \ifl@dpairing\ifparledgroup%
3483     \leavevmode\marks\parledgroup@{begin}%
3484     \marks\parledgroup@series{#1}%
3485     \marks\parledgroup@type{footnoteX}%
3486     \fi\fi\normalcolor
3487   \ifparledgroup%
3488     \ifl@dpairing%
3489     \else%
3490       \setnoteswidthliketwocolumnsX@{#1}%
3491       \setnotesXpositionliketwocolumns@{#1}%
3492       \print@footnoteXrule{#1}%
3493     \fi%
3494   \else%
3495     \setnoteswidthliketwocolumnsX@{#1}%
3496     \setnotesXpositionliketwocolumns@{#1}%
3497     \print@footnoteXrule{#1}%
3498   \fi%
3499   \csuse{bhookgroupX@#1}%
3500   \unvbox\@nameuse{mpfootins#1}}
3501
3502 %
```

`\normalbfnoteX`

```
3504 \newcommand{\normalbfnoteX}[2]{%
3505   \get@thisfootnoteX{#1}%
3506   \ifledRcol%
3507     \ifluatex
3508       \footnotelang@lua[R]%
```

```

3509 \fi
3510 \@ifundefined{xpg@main@language}%if polyglossia
3511 {}%
3512 {\footnotelang@poly[R]}%
3513 \xright@appenditem{%
3514 \noexpand\led@set@index@fornote{#1}%
3515 \unexpanded{\def\this@footnoteX@reading}{\the\csname footnote#1
@reading\endcsname}%
3516 \noexpand\vbfnoteX{#1}{#2}{\expandonce\thisfootnote}%
3517 \noexpand\led@reinit@index@fornote%
3518 }%
3519 \to\inserts@listR
3520 \global\advance\insert@countR \@ne%
3521 \else%
3522 \ifluatex
3523 \footnotelang@lua%
3524 \fi
3525 \@ifundefined{xpg@main@language}%if polyglossia
3526 {}%
3527 {\footnotelang@poly}%
3528 \xright@appenditem{%
3529 \noexpand\led@set@index@fornote{#1}%
3530 \unexpanded{\def\this@footnoteX@reading}{\the\csname footnote#1
@reading\endcsname}%
3531 \noexpand\vbfnoteX{#1}{#2}{\expandonce\thisfootnote}%
3532 \noexpand\led@reinit@index@fornote%
3533 }%
3534 \to\inserts@list
3535 \global\advance\insert@count \@ne%
3536 \fi
3537 \ignorespaces}
3538
3539 %

```

\get@thisfootnoteX The macro `\get@thisfootnote` command just saves the footnote number in the `\thisfootnote` macro, depending on the use of pairing environments.

```

3540 \newcommand{\get@thisfootnoteX}[1]{%
3541 \ifl@dpairing%
3542 \protected@xdef\thisfootnote{\the\csname footnote#1@reading\endcsname
}%
3543 \else%
3544 \protected@xdef\thisfootnote{\csuse{thefootnote#1}}%
3545 \fi%
3546 }%
3547 %

```

\vbfnoteX This command calls the correct footnote-inserting commands.

```

3548 \newcommand{\vbfnoteX}[3]{%

```

```

3549 \get@fnmarkX{#1}{#3}%
3550 \@nameuse{regvfootnote#1}{#1}{#2}%
3551 }%
3552
3553 %

```

`\get@fnmarkX` This command gets the correct footnote number when typesetting parallel texts.

```

3554 \newcommand{\get@fnmarkX}[2]{%
3555   \ifbool{bool{!@dpairing}} or bool{!@dprintingpages} or bool{
!@dprintingcolumns}}{%
3556     {%
3557       \stepcounter{footnote#1@typeset}%
3558       \setcounter{footnote#1}{\value{footnote#1@typeset}}%
3559       \@namedef{@thefnmark#1}{\csuse{thefootnote#1}}%
3560       \immediate\write\@mainaux{%
3561         \global\csdef{footnote#1reading#2=typeset}{\the\csname c@footnote
#1@typeset\endcsname}%
3562       }%
3563     }%
3564     {%
3565       \@namedef{@thefnmark#1}{#2}%
3566     }%
3567   }
3568   %
3569   %

```

```

\numfootnoteX70 \newcommand{\numfootnoteX}[2]{%
3571   \ifnumberedpar@
3572     \edtext{}{\normalbfnoteX{#1}{#2}}%
3573   \else
3574     \def\this@footnoteX@reading{\the\csname footnote#1@reading\endcsname}%
3575     \@nameuse{regvfootnote#1}{#1}{#2}%
3576   \fi}
3577
3578 %

```

`arrangementX@normal` `\arrangementX@normal{<series>}` initialises the settings for the `<series>` footnotes. This should always be called for each series.

```

3579 \newcommand*{\arrangementX@normal}[1]{%
3580   \csgdef{series@displayX#1}{normal}
3581   \expandafter\let\csname footstart#1\endcsname=\normalfootstartX
3582   \expandafter\newcount\csname prevpage#1@num\endcsname%
3583   \expandafter\newcount\csname prevpage#1@numR\endcsname%
3584   \@namedef{@footnotemark#1}{\normal@footnotemarkX{#1}}
3585   \@namedef{bodyfootmark#1}{\normalbodyfootmarkX{#1}}
3586   \expandafter\let\csname regvfootnote#1\endcsname=\normalvfootnoteX
3587   \expandafter\let\csname vfootnote#1\endcsname=\numfootnoteX

```

```

3588 \expandafter\let\csname footfmt#1\endcsname=\normalfootfmtX
3589 \@namedef{footfootmark#1}{\normalfootfootmarkX{#1}}
3590 \expandafter\let\csname footgroup#1\endcsname=\normalfootgroupX
3591 \expandafter\let\csname footnoterule#1\endcsname=\normalfootnoteruleX
3592 \count\csname footins#1\endcsname=1000
3593 \csxdef{default@footins#1}{1000}%Use to have note only for one side
3594 \dimen\csname footins#1\endcsname=\csuse{maxhnotesX@#1}
3595 \skip\csname footins#1\endcsname=\csuse{beforenotesX@#1}%
3596 \advance\skip\csname footins#1\endcsname by\csuse{afterruleX@#1}%
3597 %

```

Additions for minipages.

```

3598 \ifnoledgroup@else%
3599 \expandafter\let\csname mpvfootnote#1\endcsname=\mpnormalvfootnoteX
3600 \expandafter\let\csname mpfootgroup#1\endcsname=\mpnormalfootgroupX
3601 \count\csname mpfootins#1\endcsname=1000
3602 \dimen\csname mpfootins#1\endcsname=\csuse{maxhnotesX@#1}
3603 \skip\csname mpfootins#1\endcsname=\csuse{beforenotesX@#1}%
3604 \advance\skip\csname mpfootins#1\endcsname by\csuse{afterruleX@#1}%
3605 \fi
3606 }
3607
3608 %

```

XIII.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 359 \newcommand*\arrangementX@twocol}[1]{%
3610 \csgdef{series@displayX#1}{twocol}
3611 \expandafter\let\csname regvfootnote#1\endcsname=\twocolvfootnoteX
3612 \expandafter\let\csname footfmt#1\endcsname=\twocolfootfmtX
3613 \expandafter\let\csname footgroup#1\endcsname=\twocolfootgroupX
3614 \dimen\csname footins#1\endcsname=\csuse{maxhnotesX@#1}%
3615 \skip\csname footins#1\endcsname=\csuse{beforenotesX@#1}%
3616 \advance\skip\csname footins#1\endcsname by \csuse{afterruleX@#1}\relax%
3617 \twocolfootsetupX{#1}
3618 \ifnoledgroup@else%
3619 \expandafter\let\csname mpvfootnote#1\endcsname=\mpnormalvfootnoteX
3620 \expandafter\let\csname mpfootgroup#1\endcsname=\mptwocolfootgroupX
3621 \skip\csname mpfootins#1\endcsname=\csuse{beforenotesX@#1}%
3622 \advance\skip\csname mpfootins#1\endcsname by\csuse{afterruleX@#1}
3623 \mptwocolfootsetupX{#1}
3624 \fi%
3625 }
3626
3627 %

```

```

\twocolfootsetupX \twocolfootsetupX{<series>}
\mptwocolfootsetupX
3628 \newcommand*{\twocolfootsetupX}[1]{%
3629   \count\csname footins#1\endcsname 500
3630   \csxdef{default@footins#1}{500}%Use this to confine the notes to one
side only
3631   \multiply\dimen\csname footins#1\endcsname by \tw@}
3632 \newcommand*{\mptwocolfootsetupX}[1]{%
3633   \count\csname mpfootins#1\endcsname 500
3634   \multiply\dimen\csname mpfootins#1\endcsname by \tw@}
3635
3636   %

\twocolvfootnoteX \twocolvfootnoteX{<series>}

3637 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\twocolvfootnoteX}[2]{%
3638   \csuse{beforeinsertingX@#1}%
3639   \insert\csname footins#1\endcsname\bgroup%
3640     \hsize=\expandafter\dimexpr\csuse{widthX@#1}\relax%
3641     \noindent\csuse{bhooknoteX@#1}%
3642     \csuse{notefontsizeX@#1}%
3643     \footssplitsskip%
3644     \spaceskip=\z@skip \xspaceskip=\z@skip%
3645     \@nameuse{footfmt#1}{#1}{#2}\egroup}
3646
3647   %

\twocolfootfmtX \twocolfootfmtX{<series>}

3648 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\twocolfootfmtX}[2]{%
3649   \protected@edef\@currentlabel{%
3650     \@nameuse{@thefnmark#1}%
3651   }%
3652   \normal@pars%
3653   \hangindent=\csuse{hangindentX@#1}%
3654   \everypar{\hangindent=\csuse{hangindentX@#1}}%
3655   \hsize \csuse{hsizetwocolX@#1}%
3656   \nottoggle{parindentX@#1}{\parindent=\z@}{}%
3657   \tolerance=5000\relax%
3658   \par%
3659   \@tempdima=\parindent%
3660   \csuse{colalignX@#1}%
3661   \parindent=\@tempdima%
3662   {\hspace{\parindent}%
3663     \csuse{notenunfontX@#1}\wrapped@footfootmarkX{#1}\strut%
3664     \csuse{wrapcontentX@#1}{#2}%
3665     \strut\par}%
3666   \allowbreak%
3667 }%
3668
3669   %

```

```

\twocolfootgroupX \twocolfootgroupX{<series>}
\mptwocolfootgroupX
3670 \newcommand*\twocolfootgroupX[1]{\csuse{bhookgroupX@#1}\csuse{
notefontsizeX@#1}
3671 \splittopskip=\ht\strutbox
3672 \expandafter
3673 \rigidbalanceX\csname footins#1\endcsname \tw@ \splittopskip}}
3674
3675 \newcommand*\mptwocolfootgroupX[1]{\%
3676 \vskip\skip\@nameuse{mpfootins#1}
3677 \ifl@dpairing\ifparledgroup%
3678 \leavevmode\marks\parledgroup@{begin}%
3679 \marks\parledgroup@series{#1}%
3680 \marks\parledgroup@type{footnoteX}%
3681 \fi\fi\normalcolor
3682 \ifparledgroup%
3683 \ifl@dpairing%
3684 \else%
3685 \setnoteswidthliketwocolumnsX@{#1}%
3686 \setnotesXpositionliketwocolumns@{#1}%
3687 \print@footnoteXrule{#1}%
3688 \fi%
3689 \else%
3690 \setnoteswidthliketwocolumnsX@{#1}%
3691 \setnotesXpositionliketwocolumns@{#1}%
3692 \print@footnoteXrule{#1}%
3693 \fi%
3694 \csuse{bhookgroupX@#1}%
3695 \splittopskip=\ht\strutbox
3696 \expandafter
3697 \rigidbalanceX\csname mpfootins#1\endcsname \tw@ \splittopskip}}
3698
3699 %

```

XIII.4.4 Three columns footnotes

The following macros set footnotes in three columns. It is assumed that the length of each footnote is less than the column width.

```

\arrangementX@threecol
3700 \newcommand*\arrangementX@threecol[1]{\%
3701 \csgdef{series@displayX#1}{threecol}
3702 \expandafter\let\csname regvfootnote#1\endcsname=\threecolvfootnoteX
3703 \expandafter\let\csname footfmt#1\endcsname=\threecolfootfmtX
3704 \expandafter\let\csname footgroup#1\endcsname=\threecolfootgroupX
3705 \dimen\csname footins#1\endcsname=\csuse{maxhnotesX@#1}%
3706 \skip\csname footins#1\endcsname=\csuse{beforenotesX@#1}%
3707 \advance\skip\csname footins#1\endcsname by \csuse{afterruleX@#1}\relax%
3708 \threecolfootsetupX{#1}
3709 \ifnoledgroup@\else%

```



```

3710 \expandafter\let\csname mpvfootnote#1\endcsname=\mpnormalvfootnoteX
3711 \expandafter\let\csname mpfootgroup#1\endcsname=\mpthreecolfootgroupX
3712 \skip\csname mpfootins#1\endcsname=\csuse{beforenotesX@#1}%
3713 \advance\skip\csname mpfootins#1\endcsname by\csuse{afterruleX@#1}
3714 \mpthreecolfootsetupX{#1}
3715 \fi%
3716 }
3717
3718 %

```

`\threecolfootsetupX` `\threecolfootsetupX{<series>}`

`\mpthreecolfootsetupX`

```

3719 \newcommand*{\threecolfootsetupX}[1]{%
3720 \count\csname footins#1\endcsname 333
3721 \csxdef{default@footins#1}{333}%Use this to confine the notes to one
side only
3722 \multiply\dimen\csname footins#1\endcsname by \thr@@}
3723 \newcommand*{\mpthreecolfootsetupX}[1]{%
3724 \count\csname mpfootins#1\endcsname 333
3725 \multiply\dimen\csname mpfootins#1\endcsname by \thr@@}
3726
3727 %

```

`\threecolvfootnoteX` `\threecolvfootnoteX{<series>}{<text>}`

```

3728 \notbool{parapparatus@}{\newcommand*{\threecolvfootnoteX}[2]{
%
3729 \csuse{beforeinsertingX@#1}%
3730 \insert\csname footins#1\endcsname\bgroup%
3731 \hsize=\expandafter\dimexpr\csuse{widthX@#1}\relax%
3732 \noindent\csuse{bhooknoteX@#1}%
3733 \csuse{notefontsizeX@#1}%
3734 \footplitskips%
3735 \@nameuse{footfmt#1}{#1}{#2}\egroup}
3736
3737 %

```

`\threecolfootfmtX` `\threecolfootfmtX{<series>}`

```

3738 \notbool{parapparatus@}{\newcommand*{\threecolfootfmtX}[2]{%
3739 \protected@edef\@currentlabel{%
3740 \nameuse{@thefnmark#1}%
3741 }%
3742 \hangindent=\csuse{hangindentX@#1}%
3743 \everypar{\hangindent=\csuse{hangindentX@#1}}%
3744 \normal@pars%
3745 \hsize \csuse{hsizethreecolX@#1}%
3746 \nottoggle{parindentX@#1}{\parindent=\z@}{}%
3747 \tolerance=5000\relax%
3748 \@tempdima=\parindent%

```

```

3749 \csuse{colalignX@#1}%
3750 \parindent=\@tempdima%
3751 {\hspace{\parindent}%
3752 \csuse{notenumfontX@#1}\wrapped@footfootmarkX{#1}\strut%
3753 \csuse{wrapcontentX@#1}{#2}%
3754 \strut\par}\allowbreak
3755
3756 %

```

```

\threecolfootgroupX \threecolfootgroupX{<series>}
\mpthreecolfootgroupX
3757 \newcommand*{\threecolfootgroupX}[1]{\{\csuse{bhookgroupX@#1}\csuse{
notefontsizeX@#1}
3758 \splittopskip=\ht\strutbox
3759 \expandafter
3760 \rigidbalanceX\csname footins#1\endcsname \thr@@ \splittopskip}}
3761
3762 \newcommand*{\mpthreecolfootgroupX}[1]{\{
3763 \vskip\skip\@nameuse{mpfootins#1}
3764 \ifl@dpairing\ifparledgroup
3765 \leavevmode\marks\parledgroup@{begin}%
3766 \marks\parledgroup@series{#1}%
3767 \marks\parledgroup@type{footnoteX}%
3768 \fi\fi\normalcolor
3769 \ifparledgroup%
3770 \ifl@dpairing%
3771 \else%
3772 \setnoteswidthliketwocolumnsX@{#1}%
3773 \setnotesXpositionliketwocolumns@{#1}%
3774 \print@footnoteXrule{#1}%
3775 \fi%
3776 \else%
3777 \setnoteswidthliketwocolumnsX@{#1}%
3778 \setnotesXpositionliketwocolumns@{#1}%
3779 \print@footnoteXrule{#1}%
3780 \fi%
3781 \csuse{bhookgroupX@#1}%
3782 \splittopskip=\ht\strutbox
3783 \expandafter
3784 \rigidbalanceX\csname mpfootins#1\endcsname \thr@@ \splittopskip}}
3785
3786 %

```

XIII.4.5 Paragraphed footnotes

The following macros set footnotes as one paragraph.

```
\arrangementX@threecol \footparagraphX{<series>}
```

```

3787 \newcommand*{\arrangementX@paragraph}[1]{%
3788   \csgdef{series@displayX#1}{paragraph}%
3789   \expandafter\newcount\csname prevpage#1@num\endcsname%
3790   \expandafter\newcount\csname prevpage#1@numR\endcsname%
3791   \expandafter\let\csname footstart#1\endcsname=\parafootstartX
3792   \expandafter\let\csname regvfootnote#1\endcsname=\para@vfootnoteX
3793   \expandafter\let\csname footfmt#1\endcsname=\parafootfmtX
3794   \expandafter\let\csname footgroup#1\endcsname=\para@footgroupX
3795   \expandafter\let\csname footnoterule#1\endcsname=\normalfootnoteruleX
3796   \count\csname footins#1\endcsname=1000
3797   \csxdef{default@footins#1}{1000}%Use this to confine the notes to one
side only
3798   \dimen\csname footins#1\endcsname=\csuse{maxhnotesX@#1}
3799   \skip\csname footins#1\endcsname=\csuse{beforenotesX@#1}%
3800   \advance\skip\csname footins#1\endcsname by\csuse{afterruleX@#1}%
3801   \para@footsetupX{#1}
3802   \ifnoledgroup@else
3803     \expandafter\let\csname mpvfootnote#1\endcsname=\mppara@vfootnoteX
3804     \expandafter\let\csname mpfootgroup#1\endcsname=\mppara@footgroupX
3805     \count\csname mpfootins#1\endcsname=1000
3806     \dimen\csname mpfootins#1\endcsname=\csuse{maxhnotesX@#1}
3807     \skip\csname mpfootins#1\endcsname=\csuse{beforenotesX@#1}%
3808     \advance\skip\csname mpfootins#1\endcsname by\csuse{afterruleX@#1}%
3809   \fi
3810 }
3811
3812 %

```

`\para@footsetupX` `\para@footsetupX{<series>}`

```

3813 \newcommand*{\para@footsetupX}[1]{\csuse{bhookgroupX@#1}\csuse{
notefontsizeX@#1}
3814 \setnoteswidthliketwocolumnsX@#1}%
3815 \ifcsempy{widthX@#1}%
3816   {}%
3817   {\columnwidth=\expandafter\dimexpr\csuse{widthX@#1}\relax}%
3818   \dimen0=\baselineskip
3819   \multiply\dimen0 by 1024
3820   \divide\dimen0 by \columnwidth \multiply\dimen0 by \footfudgefiddle\relax
%
3821 \expandafter
3822 \xdef\csname footfudgefactor#1\endcsname{%
3823   \expandafter\strip@pt\dimen0 }}
3824
3825 %

```

`\parafootstartX` `\parafootstartX{<series>}`

```

3826 \newcommand*{\parafootstartX}[1]{%
3827   \ifdimequal{Opt}{\prenotesX@}{}%

```

```

3828 {%
3829 \iftoggle{prenotesX@}{%
3830   \togglefalse{prenotesX@}%
3831   \skip\csname footins#1\endcsname=%
3832   \glueexpr\csuse{prenotesX@}+\csuse{afterruleX@#1}\relax%
3833 }%
3834 {}%
3835 }%
3836 \leftskip=\z@
3837 \rightskip=\z@
3838 \nottoggle{parindentX@#1}{\parindent=\z@}{}%
3839 \vskip\skip\@nameuse{footins#1}%
3840 \setnoteswidthliketwocolumnsX@{#1}%
3841 \setnotesXpositionliketwocolumns@{#1}%
3842 \print@footnoteXrule{#1}%
3843 }
3844
3845 %

```

`\para@vfootnoteX` `\para@vfootnoteX{<series>}{<text>}`
`\mppara@vfootnoteX`

```

3846 \newcommand*{\para@vfootnoteX}[2]{%
3847   \csuse{beforeinsertingX@#1}%
3848   \insert\csname footins#1\endcsname%
3849   \bgroup
3850     \csuse{notefontsizeX@#1}
3851     \footsplitskips
3852     \setbox0=\vbox{\hsize=\maxdimen%
3853       \let\bidir@RTL@everypar\@empty%
3854       \noindent\csuse{bhooknoteX@#1}%
3855       \@nameuse{footfmt#1}{#1}{#2}}%
3856     \setbox0=\hbox{\unvxxX{0}{#1}}%
3857     \dp0=\z@
3858     \ht0=\csname footfudgefactor#1\endcsname\wd0
3859     \box0
3860     \penalty0
3861   \egroup}
3862 \newcommand*{\mppara@vfootnoteX}[2]{%
3863   \get@thisfootnoteX{#1}%
3864   \get@fnmarkX{#1}{\thisfootnote}%
3865   \edef\this@footnoteX@reading{\the\csname footnote#1@reading\endcsname}%
3866   \global\setbox\@nameuse{mpfootins#1}\vbox{%
3867     \unvbox\@nameuse{mpfootins#1}
3868     \csuse{notefontsizeX@#1}
3869     \footsplitskips
3870     \setbox0=\vbox{\hsize=\maxdimen%
3871       \let\bidir@RTL@everypar\@empty%
3872       \noindent\color@begingroup%
3873       \csuse{bhooknoteX@#1}%
3874       \@nameuse{footfmt#1}{#1}{#2}\color@endgroup}%

```

```

3875 \setbox0=\hbox{\unvvhX{0}{#1}}%
3876 \dp0=\z@
3877 \ht0=\csname footfudgefactor#1\endcsname\wd0
3878 \box0
3879 \penalty0}}
3880
3881 %

```

`\unvvhX`₈₂ `\newcommand*{\unvvhX}[2]{%` 2th is optional for retro-compatibility

```

3883 \setbox0=\vbox{\unvbox#1%
3884 \global\setbox1=\lastbox}%
3885 \unhbox1
3886 \unskip           % remove \rightskip,
3887 \unskip           % remove \parfillskip,
3888 \unpenalty        % remove \penalty of 10000,
3889 \hskip\csuse{afternoteX@#2}%
3890 \relax}% but add the glue to go between the notes
3891
3892 %

```

`\parafootfmtX` `\parafootfmtX{<series>}`

```

3893 \newcommand*{\parafootfmtX}[2]{%
3894 \protected@edef\@currentlabel{%
3895 \@nameuse{@thefnmark#1}%
3896 }%
3897 \insertparafootsepX{#1}%
3898 \ledsetnormalparstuff@common%
3899 {\csuse{notenumfontX@#1}%
3900 \csuse{notenumfontX@#1}%
3901 \wrapped@footfootmarkX{#1}%
3902 \strut%
3903 \csuse{wrapcontentX@#1}{#2}%
3904 \penalty-10}}
3905
3906 %

```

`\para@footgroupX` `\para@footgroupX{<series>}`

`\mppara@footgroupX`

```

3907 \newcommand*{\para@footgroupX}[1]{%
3908 \hsize=\expandafter\dimexpr\csuse{widthX@#1}\relax%
3909 \unvbox\csname footins#1\endcsname
3910 \ifcsstring{raggedX@#1}{L}{\RaggedLeft}{}%
3911 \ifcsstring{raggedX@#1}{R}{\RaggedRight}{}%
3912 \makehboxofhboxes
3913 \setbox0=\hbox{\unhbox0 \removehboxes}%
3914 \csuse{bhookgroupX@#1}
3915 \csuse{notefontsizeX@#1}
3916 \unhbox0\par}

```

```

3917
3918 \newcommand*\mppara@footgroupX}[1]{%
3919   \setnoteswidthliketwocolumnsX@{#1}%
3920   \vskip\skip\@nameuse{mpfootins#1}
3921   \ifl@dpairing\ifparledgroup
3922     \leavevmode%
3923     \leavevmode\marks\parledgroup@{begin}%
3924     \marks\parledgroup@series{#1}%
3925     \marks\parledgroup@type{footnoteX}%
3926   \fi\fi\normalcolor
3927   \ifparledgroup%
3928     \ifl@dpairing%
3929     \else%
3930       \setnoteswidthliketwocolumnsX@{#1}%
3931       \setnotesXpositionliketwocolumns@{#1}%
3932       \print@footnoteXrule{#1}%
3933     \fi%
3934   \else%
3935     \setnoteswidthliketwocolumnsX@{#1}%
3936     \setnotesXpositionliketwocolumns@{#1}%
3937     \print@footnoteXrule{#1}%
3938   \fi%
3939   \unvbox\csname mpfootins#1\endcsname
3940   \ifcsstring{raggedX@#1}{L}{\RaggedLeft}{}%
3941   \ifcsstring{raggedX@#1}{R}{\RaggedRight}{}%
3942   \makehboxofhboxes
3943   \setbox0=\hbox{\unhbox0 \removehboxes}%
3944   \csuse{bhookgroupX@#1}%
3945   \csuse{notefontsizeX@#1}%
3946   \nottoggle{parindentX@#1}{\parindent=\z@}{}%
3947   \unhbox0\par}}
3948
3949 %

```

Insertion of the footnotes separator The command `\insertparafootsepX{<series>}` must be called at the beginning of `\parafootftmX`.

```

\insertparafootsepX50 \newcommand{\insertparafootsepX}[1]{%
3951   \ifledRcol@%
3952     \ifnumequal{\csuse{prevpage#1@numR}}{\page@numR}%
3953       {\csuse{Xparafootsep@#1}}%
3954     {}%
3955     \global\csname prevpage#1@numR\endcsname=\page@numR%
3956   \else%
3957     \ifnumequal{\csuse{prevpage#1@num}}{\page@num}%
3958       {\csuse{Xparafootsep@#1}}%
3959     {}%
3960     \global\csname prevpage#1@num\endcsname=\page@num%
3961   \fi%

```

```
3962 }
3963 %
```

XIII.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.

```
3964 \newcommand{\wrapped@footfootmarkX}[1]{%
3965   \ifdefined\hypertarget%
3966     \hyperlink%
3967       {@bodyfootmark#1@\this@footnoteX@reading}%
3968       {@\@nameuse{footfootmark#1}}}%
3969   \Hy@raisedlink{%
3970     \hypertarget%
3971       {@footnotemark#1@\this@footnoteX@reading}%
3972       {}%
3973   }%
3974   \else%
3975     \@nameuse{footfootmark#1}%
3976   \fi%
3977 }%
3978 %
```

\wrapped@bodyfootmarkX \wrapped@bodyfootmarkX prints the footnote mark of the text body, wrapped in hyperref package's commands, if needed.

```
3979 \newcommand{\wrapped@bodyfootmarkX}[1]{%
3980   \ifdefined\hypertarget%
3981     \hyperlink%
3982       {@footnotemark#1\expandafter\the\csname footnote#1@reading\
3983   endcsname}%
3984       {@\@nameuse{bodyfootmark#1}}}%
3985   \Hy@raisedlink{%
3986     \hypertarget%
3987       {@bodyfootmark#1\expandafter\the\csname footnote#1@reading\
3988   endcsname}%
3989       {}%
3990   }%
3991   \else%
3992     \@nameuse{bodyfootmark#1}%
3993   \fi%
3994 }%
3995 %
```

XIV 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 `LuaLaTeX`.

That why we have defined `\ledsetnormalparstuff@common` in order to make this setting for both familiar and critical notes. This command is called by command to make specific setting to critical or familiar footnote.

```

\ledsetnormalparstuff@common% \newcommand*{\ledsetnormalparstuff@common}{%
  \ifluatex%
  \textdir\footnote@luatexttextdir%
  \pardir\footnote@luatexpardir%
  \fi%
  \csuse{\csuse{footnote@dir}}%
  \normal@pars%
  \parfillskip \z@ \@plus 1fil}%
%
\newcommand*{\Xledsetnormalparstuff}[1]{%
  \ledsetnormalparstuff@common%
  \nottoggle{Xparindent@#1}{\parindent=\z@}{\hspace{\parindent}}%
}%
%
\newcommand*{\ledsetnormalparstuffX}[1]{%
  \ledsetnormalparstuff@common%
  \nottoggle{parindentX@#1}{\parindent=\z@}{\hspace{\parindent}}%
}%
%
```

XV 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@`

```

%
\newdimen\old@hsize%
\AtBeginDocument{\old@hsize=\hsize}%
%
\newcommand{\setXnoteswidthliketwocolumns@}[1]{%
```



```

4018 \global\let\hsize@fornote=\hsize%
4019 \global\old@hsize=\hsize%
4020 \let\old@columnwidth=\columnwidth%
4021 \iftoggle{Xnoteswidthliketwocolumns@#1}{%
4022   {%
4023     \csuse{setwidthliketwocolumns@\columns@position}%
4024     \global\let\hsize@fornote=\hsize%
4025   }%
4026   {}%
4027   \let\hsize=\hsize@fornote%
4028   \let\columnwidth=\old@columnwidth%
4029 }%
4030
4031 \newcommand{\setnoteswidthliketwocolumnsX@}[1]{%
4032   \global\let\hsize@fornote=\hsize%
4033   \global\old@hsize=\hsize%
4034   \let\old@columnwidth=\columnwidth%
4035   \iftoggle{noteswidthliketwocolumnsX@#1}{%
4036     {%
4037       \csuse{setwidthliketwocolumns@\columns@position}%
4038       \global\let\hsize@fornote=\hsize%
4039     }%
4040     {}%
4041     \let\hsize=\hsize@fornote%
4042     \let\columnwidth=\old@columnwidth%
4043   }%
4044
4045   %

```

`\setnotespositionliketwocolumns@` 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`.

```

4046 \newcommand{\setXnotespositionliketwocolumns@}[1]{%
4047   \iftoggle{Xnoteswidthliketwocolumns@#1}{%
4048     \csuse{setnotespositionliketwocolumns@\columns@position}%
4049   }{%
4050   }%
4051
4052 \newcommand{\setnotesXpositionliketwocolumns@}[1]{%
4053   \iftoggle{noteswidthliketwocolumnsX@#1}{%
4054     \csuse{setnotespositionliketwocolumns@\columns@position}%
4055   }{%
4056   }%
4057
4058   %

```

XVI Footnotes' order

`\fnpos` The `\fnpos` and `\mpfnpos` simply place their arguments in `\@fnpos` and `\@mpfnpos`, which will be used later in the output routine.

```

4059 \def\@fnpos{familiar-critical}
4060 \def\@mpfnpos{critical-familiar}
4061 \newcommand{\fnpos}[1]{\xdef\@fnpos{#1}}
4062 \newcommand{\mpfnpos}[1]{\xdef\@mpfnpos{#1}}
4063 %

```

XVII 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\newcommand{\print@Xfootnoterule}[1]{%
\print@footnoteXrule\vskip-\csuse{Xafterrule@#1}%Because count in \dimen\csuse{#1footins}
4066 \nointerlineskip%
4067 \moveleft-\leftskip\vbox{\csuse{#1footnoterule}}%
4068 \nointerlineskip%
4069 \vskip\csuse{Xafterrule@#1}%
4070 }%
4071
4072 \newcommand{\print@footnoteXrule}[1]{%
4073 \vskip-\csuse{afterruleX@#1}%Because count in \dimen\csuse{footins#1}
4074 \nointerlineskip%
4075 \moveleft-\leftskip\vbox{\csuse{footnoterule#1}}%
4076 \nointerlineskip%
4077 \vskip\csuse{afterruleX@#1}%
4078 }%
4079
4080 %

```

XVIII Specific skip for first series of footnotes

XVIII.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@preXnotes` 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 !

XVIII.0.2 User level command

`\preXnotes@` If user redefines `\preXnotes@`, via `\preXnotes` to a value greater than 0 pt, this skip will be added before first series notes instead of the notes skip.

```
4081 \newtoggle{preXnotes@}
4082 \toggletrue{preXnotes@}
4083 \newcommand{\preXnotes@}{0pt}
4084 \newcommand*{\preXnotes}[1]{\renewcommand{\preXnotes@}{#1}}
4085 %
```

The same, but for familiar footnotes.

```
\preXnotes@ \newtoggle{prenotesX@}
\preXnotes@ \toggletrue{prenotesX@}
4088 \newcommand{\prenotesX@}{0pt}
4089 \newcommand*{\prenotesX}[1]{\renewcommand{\prenotesX@}{#1}}
4090 %
```

XVIII.0.3 Internal commands

```
firstXseries@ \gdef\firstXseries@{}
prepare@preXnotes@ \newcommand{\prepare@preXnotes}[1]{%
4093 \ifdimequal{0pt}{\preXnotes@}%
4094 {}}%
4095 {}%
4096 \IfStrEq{\firstXseries@}{}{%}
```

```

4097 \global\skip\csuse{#1footins}=\preXnotes@%
4098 \global\advance\skip\csname #1footins\endcsname by\csuse{Xafterrule@
#1}%
4099 \gdef\firstXseries@{#1}%
4100 }%
4101 {%
4102 \ifseriesbefore{#1}{\firstXseries@}%
4103 {%
4104 \global\skip\csuse{#1footins}=\csuse{Xbeforenotes@firstXseries@}%
4105 \global\advance\skip\csname #1footins\endcsname by\csuse{Xafterrule@
#1}%
4106 \gdef\firstXseries@{#1}%
4107 }%
4108 {%}
4109 }%
4110 }%
4111 }
4112 %

```

The same thing is required for familiar notes and `\prenotesX`.

```

firstseriesX@13 \gdef\firstseriesX@{}
prepare@prenotesX14 \newcommand{\prepare@prenotesX}[1]{%
4115 \ifdimequal{0pt}{\prenotesX@}%
4116 {}%
4117 {%
4118 \IfStrEq{\firstseriesX@}{-}{%
4119 \global\skip\csuse{footins#1}=\prenotesX@%
4120 \global\advance\skip\csname footins#1\endcsname by\csuse{afterruleX@
#1}%
4121 \gdef\firstseriesX@{#1}%
4122 }%
4123 {%
4124 \ifseriesbefore{#1}{\firstseriesX@}%
4125 {%
4126 \global\skip\csuse{footins#1}=\csuse{beforenotesX@firstseriesX@}%
4127 \global\advance\skip\csname footins#1\endcsname by\csuse{afterruleX@
#1}%
4128 \gdef\firstXseries@{#1}%
4129 }%
4130 {}%
4131 }%
4132 }%
4133 }
4134 %

```

XIX Endnotes

First, check the noend option.

```
4135 \ifbool{noend@}{}{%Used instead of \ifnoend@ to prevent expansion problem
4136 %
```

`\l@dend@open` `\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.

```
4137 \newcommand{\l@dend@open}[1]{%
4138   \global\booltrue{l@dend@#1}%
4139   \expandafter\immediate%
4140     \expandafter\openout%
4141       \csname l@d@#1end\endcsname%
4142     =\jobname.#1end\relax%
4143 }%
4144 \newcommand{\l@dend@close}[1]{%
4145   \global\boolfalse{l@dend@#1}%
4146   \expandafter\immediate%
4147     \expandafter\closeout\csname l@d@#1end\endcsname%
4148 }%
4149
4150 %
```

`\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.

```
4151 \newcommand{\l@dend@stuff}{%
4152   \def\do##1{%
4153     \ifbool{l@dend@##1}{}%
4154       {\l@dend@open{##1}}%
4155     \expandafter\immediate\expandafter\write\csname l@d@##1end\endcsname{\
string\l@d@section{\the\section@num}}%
4156   }%
4157   \dolistloop{\@series}%
4158 }%
4159
4160 %
```

`\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.

```

4161 \global\newbool{parapparatus@}{\long\def\endprint#1#2#3#4#5#6#7{%
4162   \csuse{Xendhooknote@#4}%
4163   \csuse{Xendnotefontsize@#4}%
4164   \hangindent=\csuse{Xendhangindent@#4}%
4165   \ifXendinsertsep%
4166     \hskip\csuse{Xendafternote@#4}\relax%
4167     \csuse{Xendsep@#4}%
4168   \else%
4169     \iftoggle{Xendparagraph@#4}%
4170       {\global\Xendinsertsep@true}%
4171       {}%
4172   \fi%
4173   \xdef\@currentseries{#4}%
4174   \def\do##1{%
4175     \setkeys[mac]{truefootnoteoption}{##1}%
4176   }%
4177   \notblank{#5}{\docsvlist{#5}}{}%
4178   \IfStrEq{#6}{R}{\ledRcol@true}{}%
4179   \def\@this@crossref@start{#7:start}%
4180   \def\@this@crossref@end{#7:end}%
4181   \printlineendnote{#1}{#4}%
4182   \IfStrEq{#6}{R}{\ledRcol@false}{}%
4183   \undef\@this@crossref@start%
4184   \undef\@this@crossref@end%
4185   \nottoggle{Xendlemmadisablefontselection@#4}%
4186     {\select@lemmafont#1}%
4187     {}%
4188   \bgroup%
4189     \csuse{Xendlemmafont@#4}%
4190     \csuse{Xendwraplemma@#4}{#2}%
4191   \egroup%
4192   \ifboolexpr{
4193     togl {nosep@}%
4194     or test{\ifcempty{Xendlemmaseparator@#4}}%
4195   }%
4196     {\hskip\csuse{Xendinplaceoflemmaseparator@#4}\relax}%
4197     {\nobreak%

```

```

4198 \hskip\csuse{Xendbeforelemmaseparator@#4}%
4199 \csuse{Xendlemmaseparator@#4}%
4200 \hskip\csuse{Xendafterlemmaseparator@#4}%
4201 \relax%
4202 }%
4203 \csuse{Xendwrapcontent@#4}{#3}%
4204 \nottoggle{Xendparagraph@#4}{\par}{}%
4205 \def\do##1{%
4206 \setkeys[mac]{falsefootnoteoption}{##1}%
4207 }%
4208 \notblank{#5}{\docsvlist{#5}}{}%
4209 }}%
4210
4211 \let\l@d@section=\@gobble
4212
4213 %

```

\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.

```

4214 \newcommand{\printlineendnote}[2]{%
4215 \l@dp@rsefootspec#1|%
4216 \iftoggle{Xendnumberonlyfirstintwolines@#2}{%
4217 \edef\lineinfo@{\l@dparsedstartpage - \l@dparsedstartline - \
4218 \l@dparsedstartsub - \l@dparsedendpage - \l@dparsedendline - \
4219 \l@dparsedendsub}%
4220 }%
4221 {%
4222 \edef\lineinfo@{\l@dparsedstartpage - \l@dparsedstartline - \
4223 \l@dparsedstartsub}%
4224 }%
4225 \ifboolexpr{%
4226 togl {nonum@}%
4227 or togl {Xendnonumber@#2}}%
4228 {%
4229 \iftoggle{Xendnumberonlyfirstinline@#2}%
4230 {\ifcsdef{prevendline#2}%
4231 {\ifcsequal{prevendline#2}{\lineinfo@}%
4232 {%
4233 \csuse{Xendbhookinplaceofnumber@#2}%
4234 \ifcsepty{Xendsymlinenum@#2}%
4235 {\hspace{\csuse{Xendinplaceofnumber@#2}}}%
4236 {\printsymlineendnotearea{#2}}}%
4237 \csuse{Xendahookinplaceofnumber@#2}%
4238 }%
4239 {\printlineendnotearea{#1}{#2}}}%
4240 {\printlineendnotearea{#1}{#2}}%

```

```

4240 }%
4241 {\printlineendnotearea{#1}{#2}}%We keep every time line
4242 \csxdef{prevendline#2}{\lineinfo@}%
4243 }%
4244 }%
4245 %

```

```

\printsymlineendnotearea \newcommand{\printsymlineendnotearea}[1]{%
4246
4247 \hspace{\csuse{Xendbeforesymlinenum@#1}}%
4248 \csuse{Xendnotenumfont@#1}%
4249 \ifdimequal{\csuse{Xendboxsymlinenum@#1}}{\z@}%
4250 {\csuse{Xendsymlinenum@#1}}%
4251 {\hbox to \csuse{Xendboxsymlinenum@#1}%
4252 {\csuse{Xendsymlinenum@#1}\hfill}%
4253 }%
4254 \hspace{\csuse{Xendaftersymlinenum@#1}}%
4255 }%
4256 %

```

\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.)

```

4257 \newcommand{\printlineendnotearea}[2]{%
4258 \csuse{Xendbhooklinenumber@#2}%
4259 \hspace{\csuse{Xendbeforenumber@#2}}%
4260 \bgroup%
4261 \csuse{Xendnotenumfont@#2}%
4262 \ifdimequal{\csuse{Xendboxlinenum@#2}}{\Opt}%
4263 {\printendlines#1||\ifledRcol@{\@Rlineflag\fi}%
4264 {\leavevmode%
4265 \hbox to \csuse{Xendboxlinenum@#2}%
4266 {%
4267 \IfSubStr{RC}{\csuse{Xendboxlinenumalign@#2}}{\hfill}{}%
4268 \printendlines#1||\ifledRcol@{\@Rlineflag\fi}%
4269 \IfSubStr{LC}{\csuse{Xendboxlinenumalign@#2}}{\hfill}{}%
4270 }%
4271 \egroup%
4272 \hspace{\csuse{Xendafternumber@#2}}%
4273 \csuse{Xendahooklinenumber@#2}%
4274 }%
4275 %

```

\doendnotes `\doendnotes` is the command you use to print one series of endnotes; it takes one argument: the series letter of the note series you want to print. `\Xendinsertsep@` is set to true at the first note of the series, and to false at the last one.


```

4276 \newif\ifXendinsertsep%
4277 \newcommand*{\doendnotes}[1]{%
4278   \l@dend@close{#1}%
4279   \begingroup
4280     \makeatletter
4281     \expandafter\let\csname #1end\endcsname=\endprint
4282     \input\jobname.#1end%
4283     \global\Xendinsertsep=false%
4284   \endgroup}
4285 %

```

\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.

```

4286 \newcommand*{\doendnotesbysection}[1]{%
4287   \l@dend@close{#1}%
4288   \global\expandafter\advance\csname #1end@bysection\endcsname by 1%
4289   \begingroup%
4290     \makeatletter%
4291     \def\l@d@section##1{%
4292       \ifnumequal{##1}{\csname #1end@bysection\endcsname}%
4293       {\cslet{#1end}{\endprint}}%
4294       {\cslet{#1end}{\@gobblefive}}%
4295     }%
4296     \input\jobname.#1end%
4297     \global\Xendinsertsep=false%
4298   \endgroup%
4299 }%
4300 %

```

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.

```

4301 }%
4302 %

```

\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.

First of all, we print the second page number only if the ending page number is different from the starting page number.

```

4303 \newcommand*{\setprintendlines}[6]{%
4304   \l@dpnumfalse \l@ddashfalse
4305   \ifnum#4=#1 \else
4306     \l@dpnumtrue
4307     \l@ddashtrue
4308   \fi
4309 %

```

We print the ending line number if: (1) we are printing the ending page number, or (2) it is different from the starting line number.

```

4310   \ifl@dpnum \l@d@elintrue \else \l@d@elinfalse \fi
4311   \ifnum#2=#5 \else
4312     \l@d@elintrue
4313     \l@d@dashtrue
4314   \fi
4315 %

```

We print the starting sub-line if it is nonzero.

```

4316   \l@dssubfalse
4317   \ifnum#3=0 \else
4318     \l@dssubtrue
4319   \fi
4320 %

```

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.

```

4321   \l@d@eslfalse
4322   \ifnum#6=0 \else
4323     \ifnum#6=#3
4324       \ifl@d@elin \l@d@esltrue \else \l@d@eslfalse \fi
4325     \else
4326       \l@d@esltrue
4327       \l@d@dashtrue
4328     \fi
4329   \fi%
4330 %

```

```

4331   \ifl@d@dash%
4332     \ifboolexpr{togl{fulllines@} or test{\ifcempty{Xendtwolines@}\@currentseries}}}%
4333     {%
4334     {%
4335       \setistwofollowinglines{#1}{#2}{#4}{#5}%
4336       \ifboolexpr{%
4337         (%
4338           togl {Xendtwolinesbutnotmore@\@currentseries}%
4339           and not%
4340         (%
4341           bool {istwofollowinglines@}%

```

```

4342         )%
4343     )%
4344     or%
4345     (%
4346         (not test{\ifnumequal{#1}{#4}})%
4347         and togl{Xendtwolinesonlyinsamepage@\@currentseries}%
4348     )%
4349 }%
4350 {}%
4351 {%
4352     \l@d@dashfalse%
4353     \l@d@Xtwolinestrue%
4354     \l@d@elinfalse%
4355     \l@d@eslfalse%
4356     \ifcempty{Xendmorethantwolines@\@currentseries}%
4357     {%
4358         {\ifistwofollowinglines@\else%
4359             \l@d@Xmorethantwolinestrue%
4360         }%
4361     }%
4362 }%
4363 }%
4364 \fi%
4365 %

```

End of \setprintendlines.

```

4366 }%
4367 %

```

\printendlines Now we are ready to print it all.

```

4368 \def\printendlines#1|#2|#3|#4|#5|#6|#7|#8|{\begingroup
4369     \setprintendlines{#1}{#2}{#3}{#4}{#5}{#6}%
4370 %

```

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, print the start lines.

```

4371 \ifdimequal{\csuse{Xendboxstartlinenum@\@currentseries}}{0pt}%
4372 {\bgroup}%
4373 {\leavevmode\hbox to \csuse{Xendboxstartlinenum@\@currentseries}\bgroup
\hfill}%
4374 \wrap@edcrossref{\@this@crossref@start}{\printnnum{#1}}%
4375 \ifl@d@dash%
4376 \ifl@d@pnum%
4377     \csuse{Xendlineprefixsingle@\@currentseries}%
4378 \else%

```

```

4379 \ifcsemtyp{Xendlineprefixmore@ \@currentseries}%
4380 {\csuse{Xendlineprefixsingle@ \@currentseries}}
4381 {\csuse{Xendlineprefixmore@ \@currentseries}}%
4382 \fi%
4383 \else%
4384 \csuse{Xendlineprefixsingle@ \@currentseries}%
4385 \fi%
4386 \wrap@edcrossref{\@this@crossref@start}{\linenumrep{#2}}%
4387 \iftoggle{Xendlineflag@ \@currentseries}{\ifledRcol@ \@Rlineflag\fi}{}%
4388 \ifl@d@ssub%
4389 \csuse{Xendsublinesep@ \@currentseries}%
4390 \wrap@edcrossref{\@this@crossref@start}{\sublinenumrep{#3}}%
4391 \fi%
4392 \egroup%
4393 %

```

And now, print the dash + the end line number, or the line number range symbol.

```

4394 \ifdimequal{\csuse{Xendboxendlinenum@ \@currentseries}}{0pt}%
4395 {\bgroup}%
4396 {\hbox to \csuse{Xendboxendlinenum@ \@currentseries}\bgroup}%
4397 \ifl@d@Xtwolines%
4398 \ifl@d@Xmorethantwolines%
4399 \csuse{Xendmorethantwolines@ \@currentseries}%
4400 \else%
4401 \csuse{Xendtwolines@ \@currentseries}%
4402 \fi%
4403 \else%
4404 \ifl@d@dash%
4405 \ifdefined\linrangesep@%
4406 \linrangesep@%
4407 \else%
4408 \csuse{Xendlinrangeseparator@ \@currentseries}%
4409 \fi%
4410 \fi%
4411 \ifl@d@pnnum%
4412 \wrap@edcrossref{\@this@crossref@end}\printnpnum{#4}%
4413 \fi%
4414 \ifl@d@elin%
4415 \ifl@d@pnnum\csuse{Xendlineprefixsingle@ \@currentseries}\fi%
4416 \wrap@edcrossref{\@this@crossref@end}{\linenumrep{#5}}%
4417 \iftoggle{Xendlineflag@ \@currentseries}{\ifledRcol@ \@Rlineflag\fi}{}%
4418 \fi%
4419 \ifl@d@esl%
4420 \ifl@d@elin%
4421 \csuse{Xendsublinesep@ \@currentseries}%
4422 \fi%
4423 \wrap@edcrossref{\@this@crossref@end}{\sublinenumrep{#6}}%
4424 \fi%
4425 \fi%
4426 \ifdimequal{\csuse{Xendboxendlinenum@ \@currentseries}}{0pt}%

```

```

4427     {}%
4428     {\hfill}%Prevent underfull hbox
4429     \egroup%
4430     \endgroup%
4431 }%
4432
4433 %

```

\printnpnum A macro to print a page number in an endnote. Should not be override anymore

```

4434 \newcommand*\printnpnum[1]{\csuse{Xendbeforepagenumber@ \@currentseries
4435 }#1\csuse{Xendafterpagenumber@ \@currentseries}}
4436 %

```

XX Generate series of notes

In this section, X means the name of the series (A, B etc.)

\series `\series` creates one more new series. It is a public command, which just loops on the private command `\newseries@`.

```

4437 \newcommand{\newseries}[1]{%
4438     \def\do##1{\newseries@{##1}}%
4439     \docsvlist{#1}
4440 }
4441 %

```

\@series The `\series@` macro is an etoolbox list, which contains the name of all series.

```

4442 \newcommand{\@series}{}
4443 %

```

The command `\newseries@` creates a new series of the footnote.

```

4444 \newseries@ \newcommand{\newseries@}[1]{
4445 %

```

XX.1 Test if series is still existing

```

4446 \xifinlist{#1}{\@series}{\led@warn@SeriesStillExist{#1}}%
4447 {%
4448 %

```

XX.2 Init specific to reledpar

When calling `\newseries@` after having loaded `reledpar`, we need to load specific setting.

```

4449 \ifdefined\newseries@par%
4450 \newseries@par{#1}%
4451 \fi%
4452 %

```

XX.3 For critical footnotes

Critical footnotes are those which start with letters. We look for the `\nocritical` option of `reledmac`.

```

4453 \unless\ifnocritical@
4454 %

```

XX.3.1 Options

```

4455 \newtoggle{Xlineflag@#1}
4456 \newtoggle{Xparindent@#1}
4457 \newtoggle{Xlemmadisablefontselection@#1}
4458 \csgdef{Xwrapcontent@#1}{}%
4459 \csgdef{Xbeforeinserting@#1}{}%
4460 \csgdef{Xhangindent@#1}{Opt}%
4461 \csgdef{Xragged@#1}{}%
4462 \csgdef{Xhsizetwocol@#1}{0.45 \hsize}%
4463 \csgdef{Xhsizethreecol@#1}{.3 \hsize}%
4464 \csgdef{Xcolalign@#1}{\raggedright}%
4465 \csgdef{Xnotenunfont@#1}{\normalfont}%
4466 \csgdef{Xnotefontsize@#1}{\footnotesize}%
4467 \csgdef{Xbhooknote@#1}{}%
4468 \csgdef{Xbhookgroup@#1}{}%
4469
4470 \csgdef{Xboxlinenum@#1}{Opt}%
4471 \csgdef{Xboxlinenumalign@#1}{L}%
4472
4473 \csgdef{Xboxstartlinenum@#1}{Opt}%
4474 \csgdef{Xboxendlinenum@#1}{Opt}%
4475
4476 \csgdef{Xboxsymlinenum@#1}{Opt}%
4477 \newtoggle{Xnumberonlyfirstinline@#1}%
4478 \newtoggle{Xnumberonlyfirstintwolines@#1}%
4479 \csgdef{Xtwolines@#1}{}%
4480 \csgdef{Xmorethantwolines@#1}{}%
4481 \csgdef{Xsublinesep@#1}{\fullstop}%
4482 \newtoggle{Xtwolinesbutnotmore@#1}%
4483 \newtoggle{Xtwolinesonlyinsamepage@#1}%
4484 \newtoggle{Xonlypstart@#1}%
4485 \newtoggle{Xpstarteverytime@#1}%
4486 \newtoggle{Xpstart@#1}%
4487 \newtoggle{Xstanza@#1}%

```

```

4488 \csgdef{Xstanzaseparator@#1}{}%
4489 \csgdef{Xsymlinenum@#1}{}%
4490 \newtoggle{Xnonumber@#1}%
4491 \csgdef{Xbeforenumber@#1}{0pt}%
4492 \csgdef{Xtxtbeforenumber@#1}{}%
4493 \csgdef{Xafternumber@#1}{0.5em}%
4494 \newtoggle{Xnonbreakableafternumber@#1}%
4495 \csgdef{Xbeforesymlinenum@#1}{\csuse{Xbeforenumber@#1}}%
4496 \csgdef{Xaftersymlinenum@#1}{\csuse{Xafternumber@#1}}%
4497 \csgdef{Xinplaceofnumber@#1}{1em}%
4498 \global\cslet{Xlemmaseparator@#1}{\rbracket}%
4499 \csgdef{Xbeforelemmaseparator@#1}{0em}%
4500 \csgdef{Xafterlemmaseparator@#1}{0.5em}%
4501 \csgdef{Xinplaceoflemmaseparator@#1}{1em}%
4502 \csgdef{Xbeforenotes@#1}{1.2em \@plus .6em \@minus .6em}%
4503 \csgdef{Xafterrule@#1}{0pt}%
4504
4505 \csgdef{Xtxtbeforenotes@#1}{}%
4506 \newtoggle{Xtxtbeforenotes@#1@typeset}%Not directly used by user,
but internal
4507
4508 \csgdef{Xmaxhnotes@#1}{0.8\vsizel}%
4509 \newtoggle{Xnoteswidthliketwocolumns@#1}%
4510 \csgdef{Xparafootsep@#1}{}%
4511 \csgdef{Xafternote@#1}{1em plus.4em minus.4em}%
4512 \csgdef{Xlinrangeseparator@#1}{\endashchar}%
4513
4514 \csgdef{Xlemmafont@#1}{}%
4515 \csgdef{Xwraplemma@#1}{}%
4516 \csgdef{Xwidth@#1}{\hsizel}%
4517 %

```

XX.3.2 Create inserts, needed to add notes in foot

As regards inserts, see chapter 15 of *The TeXbook* by D. Knuth.

```

4518 \expandafter\newinsert\csname #1footins\endcsname%
4519 \unless\ifnoledgroup%
4520 \expandafter\newinsert\csname mp#1footins\endcsname%
4521 \fi%
4522 %

```

XX.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.

```

4523 \global\newcommand\parapparatus@{\expandafter\newcommand\expandafter
*}{\expandafter\newcommand}\csname #1footnote\endcsname[2][]{%
4524 \ifnum\@edtext@level>0%
4525 \begingroup%

```

```

4526 \newcommand{\content}{##2}%
4527 \ifnumberedpar%
4528   \ifledRcol%
4529     \ifluatex%
4530       \footnotelang@lua[R]%
4531     \fi%
4532   \@ifundefined{xpg@main@language}%if polyglossia
4533     {}%
4534     {\footnotelang@poly[R]}%
4535   \footnoteoptions@{R}{##1}{true}%
4536   \xright@appenditem{%
4537     \ifbool{indtl@innote}%
4538       {\unexpanded{\let\index\nindex}}%
4539     {}%
4540     \ifbool{indtl@notenumber}%
4541       {\unexpanded{\let\index\nindex}}%There is no note
...number so
4542     {}%
4543     \noexpand\Xnote@true%
4544     \noexpand\prepare@preXnotes{#1}%
4545     \noexpand\prepare@edindex@fornote{\l@d@nums}%
4546     \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.
4547     \noexpand\setcounter{stanzaR}{\the\c@stanzaR}%Save
stanzaR counter for footnote
4548     \unexpanded{\def\@this@crossref@start}{\theedtext:
start}%
4549     \unexpanded{\def\@this@crossref@end}{\theedtext:end}%
4550     \noexpand\csuse{v#1footnote}{#1}%
4551     {\l@d@nums}{\expandonce\@tag}{\expandonce\content}}
%
4552     \noexpand\Xnote@false%
4553     \unexpanded{\undef\@this@crossref@start}%
4554     \unexpanded{\undef\@this@crossref@end}%
4555     \ifbool{indtl@innote}%
4556       {\unexpanded{\let\index\orig@index}}%
4557     {}%
4558     \ifbool{indtl@notenumber}%
4559       {\unexpanded{\let\index\orig@index}}%
4560     {}%
4561     }\to\inserts@listR
4562     \footnoteoptions@{R}{##1}{false}%
4563     \global\advance\insert@countR \c@ne%
4564   \else%
4565     \ifluatex%
4566       \footnotelang@lua%
4567     \fi%
4568     \@ifundefined{xpg@main@language}%if polyglossia

```



```

4569         {}%
4570         {\footnotelang@poly}%
4571 \footnoteoptions@{L}{##1}{true}%
4572 \xright@appenditem{%
4573   \ifbool{indtl@innote}%
4574     {\unexpanded{\let\index\nindex}}%
4575     {}%
4576   \ifbool{indtl@notenumber}%
4577     {\unexpanded{\let\index\nindex}}%There is no note
...number so
4578   {}%
4579   \noexpand\Xnote@true%
4580   \noexpand\prepare@preXnotes{#1}%
4581   \noexpand\prepare@edindex@fornote{\ld@nums}%
4582   \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.
4583   \ifl@dpairing%
4584   \noexpand\setcounter{stanzaL}{\the\c@stanzaL}%Save
stanzaR counter for footnote
4585   \fi%
4586   \unexpanded{\def\@this@crossref@start{\theedtext:
start}}%
4587   \unexpanded{\def\@this@crossref@end{\theedtext:end}}%
4588   \noexpand\csuse{v#1footnote}%
4589   {#1}%
4590   {\ld@nums}\expandonce\@tag\expandonce\content
}}%
4591   \unexpanded{\undef\@this@crossref@start}%
4592   \unexpanded{\undef\@this@crossref@end}%
4593   \noexpand\Xnote@false%
4594   \ifbool{indtl@innote}%
4595     {\unexpanded{\let\index\orig@index}}%
4596     {}%
4597   \ifbool{indtl@notenumber}%
4598     {\unexpanded{\let\index\orig@index}}%
4599     {}%
4600   }\to\inserts@list
4601   \global\advance\insert@count \@ne%
4602   \footnoteoptions@{L}{##1}{false}%
4603   \fi
4604   \else
4605     \csuse{v#1footnote}{#1}{0|0|0|0|0|0|0}{##1}%
4606   \fi%
4607   \endgroup%
4608   \else%
4609     \led@err@FootnoteWithoutEdtext%
4610   \fi%
4611   \ignorespaces%

```

```

4612     }
4613 %

```

We need to be able to modify reledmac's footnote macros and restore their

```

4614     \global\csletcs{#1@@footnote}{#1footnote}
4615 %

```

XX.3.4 Set standard display

```

4616     \Xarrangement@normal{#1}%
4617 %

```

End of for critical footnotes.

```

4618     \fi
4619 %

```

XX.4 For familiar footnotes

Familiar footnotes are those which end with letters. We look for the `nofamiliar` option of reledmac.

```

4620     \unless\ifnofamiliar@
4621 %

```

XX.4.1 Options

```

4622     \newtoggle{parindentX@#1}
4623     \csgdef{wrapcontentX@#1}{}%
4624     \csgdef{hangindentX@#1}{Opt}%
4625     \csgdef{beforeinsertingX@#1}{}%
4626     \csgdef{raggedX@#1}{}%
4627     \csgdef{hsizetwocolX@#1}{0.45 \hsize}%
4628     \csgdef{hsizethreecolX@#1}{.3 \hsize}%
4629     \csgdef{colalignX@#1}{\raggedright}%
4630     \csgdef{notenumfontX@#1}{\normalfont}%
4631     \csgdef{notefontsizeX@#1}{\footnotesize}%
4632     \csgdef{bhooknoteX@#1}{}%
4633     \csgdef{bhookgroupX@#1}{}%
4634     \csgdef{afterruleX@#1}{Opt}
4635     \csgdef{beforenotesX@#1}{1.2em \@plus .6em \@minus .6em}
4636     \csgdef{maxhnotesX@#1}{0.8\vsizex}%
4637     \newtoggle{noteswidthliketwocolumnsX@#1}%
4638     \csgdef{parafootsepX@#1}{}%
4639     \csgdef{afternoteX@#1}{1em plus.4em minus.4em}
4640     \csgdef{widthX@#1}{\hsizex}%
4641 % End of for familiar footnotes.
4642 % \subsubsection{Create inserts, needed to add notes in foot}
4643 % As regards inserts, see chapter 15 of the TeXBook by D. Knuth.

```

```

4644 % \begin{macrocode}
4645 \expandafter\newinsert\csname footins#1\endcsname%
4646 \unless\ifnoledgroup%
4647 \expandafter\newinsert\csname mpfootins#1\endcsname%
4648 \fi%
4649 %

```

XX.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.

```

4650 \global\expandafter\newcommand\csname footnote#1\endcsname[1]{%
4651 \begin{group}%
4652 \prepare@prenotesX{#1}%
4653 \newcommand{\content}{##1}%
4654 %
4655 %

```

If we are preparing parallel typesetting, we cannot just increase the footnote counter. Read `reledpar`'s handbook about that (V.2.1 p. 45).

```

4656 \global\expandafter\advance\csname footnote#1@reading\
4657 endcsname by \@ne%
4658 \ifl@dpairing%
4659 \ifcsdef{footnote#1@reading\the\csname footnote#1@reading\
4660 endcsname=typeset}%
4661 {\setcounter{footnote#1}{\csuse{footnote#1@reading\the\
4662 csname footnote#1@reading\endcsname=typeset}}}%
4663 {\setcounter{footnote#1}{\the\csname footnote#1@reading\
4664 endcsname}}}%
4665 \else%
4666 \stepcounter{footnote#1}%
4667 \fi%
4668 %

```

And now, the feature not depending of whether we are preparing parallel typesetting

```

4665 \protected@csxdef{@thefnmark#1}{\csuse{thefootnote#1}}%
4666 \nottoggle{nomk@}%Nomk is set to true when using \
4667 footnoteXnomk with \parpackage
4668 {\csuse{@footnotemark#1}}%
4669 {}%
4670 \ifluatex%
4671 \xdef\footnote@luatextextdir{\the\textdir}%
4672 \xdef\footnote@luatexpardir{\the\pardir}%
4673 \fi%
4674 \if@ledgroup%
4675 \led@set@index@fornote{#1}%
4676 \fi%
4677 \csuse{vfootnote#1}{#1}{\expandonce\content}\m@mmf@prepare%
4678 \ifbool{indtl@innote}%
4679 {\let\index\orig@index}%

```

```

4679         {}%
4680         \ifbool{indtl@notenumber}%
4681         {\let\index\orig@@index}%
4682         {}%
4683     \endgroup%
4684 }
4685 %

```

Then define the counters. The \TeX counter `footnoteX` is the only one manipulated by the user. This is this the one which is printed. The \TeX counter `\footnoteX@reading` is increased at each footnote. It is used for hyperlinks, for using `hyperlink` package, and for getting the correct footnote number when using parallel typesetting (V.2.1 p. 45).

```

4686     \newcounter{footnote#1}
4687     \global\expandafter\renewcommand\csname thefootnote#1\endcsname{\
arabic{footnote#1}}
4688     \expandafter\newcount\csname footnote#1@reading\endcsname%
4689 %

```

Do not forget to initialize series

```

4690     \arrangementX@normal{#1}%
4691     \fi
4692 %

```

XX.5 The endnotes

Endnotes are commands like `\Xendnote`, where `X` is a series letter. First, we check for the `noend` options.

```

4693     \unless\ifnoend@
4694 %

```

XX.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`

```

4695     \expandafter\newwrite\csname l@d@#1end\endcsname%
4696     \expandafter\newif\csname ifl@dend@#1\endcsname%
4697 %

```

XX.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.

```

4698      \global\expandafter\newcommandx\csname #1endnote\endcsname[2][1,
4699      usedefault]{%
4700          \bgroup%
4701          \newlinechar='40%
4702          \global\@noneed@Footnotetrue%
4703          \newcommand{\content}{##2}%
4704          \stepcounter{labidx}%
4705          \expandafter\immediate\expandafter\write\csname l@d@#1end\
endcsname{%
4706              \expandafter\string\csname #1end\endcsname%
4707              {\ifnumberedpar@l@d@nums\fi}%
4708              {\ifnumberedpar@\expandonce\@tag\fi}%
4709              {\expandonce\content}%
4710              {#1}%
4711              {\unexpanded{##1}}%
4712              {\ifledRcol R\else L\fi}%
4713              {\theedtext}%
4714              \@percentchar%
4715          }%
4716          \egroup%
4717          \ignorespaces%
4718      }%
4719  %

```

\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 \@gobblefive, which just skips over its five arguments.

```

4720      \global\cslet{#1end}{\@gobblefive}
4721  %
4722 %

```

We need to store the number of times \doendnotesbysection is called for one series.

```

4723      \global\expandafter\newcount\csname #1endbysection\endcsname%
4724  %

```

XX.5.3 The options

```

4725      \csgdef{Xendwraplemma@#1}{%
4726      \csgdef{Xendwrapcontent@#1}{}%
4727      \csgdef{Xendtwolines@#1}{}%
4728      \csgdef{Xendmorethantwolines@#1}{}%
4729      \newtoggle{Xendtwolinesbutnotmore@#1}{}%
4730      \newtoggle{Xendtwolinesonlyinsamepage@#1}{}%
4731      \newtoggle{Xendlemmadisablefontselection@#1}%
4732      \csgdef{Xendnotenumfont@#1}{\normalfont}%

```

```

4733 \csgdef{Xendnotefontsize@#1}{\footnotesize}%
4734 \csgdef{Xendbhooknote@#1}{}%
4735
4736 \csgdef{Xendsublinesep@#1}{\fullstop}%
4737
4738 \csgdef{Xendbeforenumber@#1}{Opt}%
4739 \csgdef{Xendafternumber@#1}{0.5em}%
4740
4741 \csgdef{Xendboxlinenum@#1}{Opt}%
4742 \csgdef{Xendboxlinenumalign@#1}{L}%
4743
4744 \csgdef{Xendboxstartlinenum@#1}{Opt}%
4745 \csgdef{Xendboxendlinenum@#1}{Opt}%
4746
4747 \csgdef{Xendlemmaseparator@#1}{}%
4748 \csgdef{Xendbeforelemmaseparator@#1}{0em}%
4749 \csgdef{Xendafterlemmaseparator@#1}{0.5em}%
4750 \csgdef{Xendinplaceoflemmaseparator@#1}{0.5em}%
4751
4752 \newtoggle{Xendparagraph@#1}%
4753 \csgdef{Xendafternote@#1}{1em plus.4em minus.4em}%
4754 \csgdef{Xendsep@#1}{}%
4755
4756 \csgdef{Xendinplaceofnumber@#1}{Opt}%
4757 \newtoggle{Xendnonnumber@#1}%
4758
4759 \csgdef{Xendhangindent@#1}{Opt}%
4760 \newtoggle{Xendnumberonlyfirstinline@#1}%
4761 \newtoggle{Xendnumberonlyfirstintwolines@#1}%
4762
4763 \csgdef{Xendbeforesymlinenum@#1}{\csuse{Xendbeforenumber@#1}}%
4764 \csgdef{Xendaftersymlinenum@#1}{\csuse{Xendafternumber@#1}}%
4765 \csgdef{Xendsymlinenum@#1}{}%
4766 \csgdef{Xendboxsymlinenum@#1}{Opt}%
4767
4768 \csgdef{Xendbhooklinenumber@#1}{}%
4769 \csgdef{Xendhooklinenumber@#1}{}%
4770 \csgdef{Xendbhookinplaceofnumber@#1}{}%
4771 \csgdef{Xendhookinplaceofnumber@#1}{}%
4772
4773 \csgdef{Xendlinerangeseparator@#1}{\endashchar}%
4774
4775 \csgdef{Xendbeforepagenumber@#1}{p.}%
4776 \csgdef{Xendafterpagenumber@#1}{) }%
4777 \csgdef{Xendlineprefixsingle@#1}{}%
4778 \csgdef{Xendlineprefixmore@#1}{}%
4779
4780 \newtoggle{Xendlineflag@#1}
4781
4782 \csgdef{Xendlemmafont@#1}{}%

```

```

4783 %
End of endnotes declaration
4784 \fi%
4785 %
Dump series in \@series
4786 \listxadd{\@series}{#1}
4787 }
4788 }% End of \newseries
4789 %

```

XX.6 Init standards series (A,B,C,D,E)

```

4790 \expandafter\newseries\expandafter{\default@series}
4791 %

```

XXI Setting series display

XXI.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.

```

4792 \newcommand{\seriesatbegin}[1]{%
4793   \StrDel{\@series}{#1}[\@series]%
4794   \edef\@new{}%
4795   \listead{\@new}{#1}%
4796   \listead{\@new}{\@series}%
4797   \xdef\@series{\@new}%
4798 }
4799 %

```

\seriesatend And `\seriesatend` moves the series to the end of the list.

```

4800 \newcommand{\seriesatend}[1]{%
4801   \StrDel{\@series}{#1}[\@series]%
4802   \edef\@new{}%
4803   \listead{\@new}{\@series}%
4804   \listead{\@new}{#1}%
4805   \xdef\@series{\@new}%
4806 }
4807 %

```

XXI.2 Test series order

\ifseriesbefore `\ifseriesbefore{<seriesA>}{<seriesB>}{<true>}{<false>}` expands `<true>` if `<seriesA>` is printed before `<seriesB>`, expands `<false>` otherwise.

```

4808 \newcommand{\ifseriesbefore}[4]{%
4809   \StrPosition{\@series}{#1}[\@first]%
4810   \StrPosition{\@series}{#2}[\@second]%
4811   \ifnumgreater{\@second}{\@first}{#3}{#4}%
4812 }
4813 %

```

XXI.2.1 Get the first series

In some specific case, we need to know the first series of the list of series.

```

\@getfirstseries\newcommand{\@getfirstseries}{%
4815   \ifdefempty{\@series}%
4816     {\xdef\@firstseries{}}%
4817     {\StrChar{\@series}{1}[\@firstseries]}%
4818 }%
4819 %

```

XXI.3 Series setting

XXI.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.

XXI.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`.

```

4820 \newcommandx{\settoggle@series}[5][4,5,usedefault]{%
4821   \def\do##1{%
4822     \global\settoggle{#2@##1}{#3}%
4823     \ifstrequal{#4}{critical}{
4824       \csuse{Xarrangement@}\csuse{series@display##1}{##1}%
4825     }{}
4826     \ifstrequal{#4}{familiar}{
4827       \csuse{arrangementX@}\csuse{series@displayX##1}{##1}%
4828     }{}
4829   }%
4830   \ifstreempty{#1}{%
4831     \dolistloop{\@series}%
4832     \ifstreempty{#5}{%
4833       \docsvlist{#5}%
4834     }
4835   }%
4836   {%
4837     \docsvlist{#1}%
4838   }%
4839 }
4840 %

```

`\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`.

```

4841 \newcommandx{\setcommand@series}[5][4,5,usedefault]{%
4842   \def\do##1{
4843     \csgdef{#2@##1}{#3}
4844     \ifstrequal{#4}{critical}{%
4845       \csuse{Xarrangement@}\csuse{series@display##1}}{##1}%
4846     }{}
4847     \ifstrequal{#4}{familiar}{%
4848       \csuse{arrangementX@}\csuse{series@displayX##1}}{##1}%
4849     }{}%
4850   }%
4851   \ifstreempty{#1}{%
4852     \dolistloop{\@series}%
4853     \ifstreempty{#5}{}%
4854     \docsvlist{#5}
4855   }
4856   }%
4857   {%
4858     \docsvlist{#1}%
4859   }%
4860 }%
4861 %

```

XXI.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`.

```

4862 \newcommandx{\newhookcommand@series}[2][2,usedefault]{%
4863   \global\expandafter\newcommand\expandafter*\csname #1\endcsname[2][]{%
4864     \setcommand@series{##1}{#1}{##2}[][#2]%
4865   }%
4866   \ifstreempty{#2}{}%
4867   \def\do##1{%
4868     \global\expandafter\newcommand\expandafter*\csname #1##1\endcsname
4869     [1]{%
4870       \csuse{#1}[##1]{####1}%
4871     }%
4872     \docsvlist{#2}%
4873   }%
4874 }
4875 %

```

\newhooktoggle@series `\newhooktoggle@series\command` names is a generic command to add new commands for a new toggle hook, like `\Xnumberonlyfirstinline`. The second argu-

ment is also used to create commands named `\<hookname><pseudoseris>`, like `\Xtwolinesbutnotmoreappref`.

```

4876 \newcommand{\newhooktoggle@series}[2][2,usedefault]{%
4877   \global\expandafter\newcommand\expandafter*\csname #1\endcsname[2][1,2={
true},usedefault]{%
4878     \settoggle@series{##1}{#1}{##2}[][#2]%
4879   }%
4880   \ifstrempy{#2}{}{%
4881     \def\do##1{%
4882       \global\expandafter\newcommand\expandafter*\csname #1##1\endcsname{%
4883         \csuse{#1}[##1]%
4884       }%
4885     }%
4886     \docsvlist{#2}%
4887   }%
4888 }
4889 %

```

hooktoggle@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

```

4890 \newcommand{\newhooktoggle@series@reload}[2]{%
4891   \global\expandafter\newcommand\expandafter*\csname #1\endcsname[2][1,2={
true},usedefault]{%
4892     \settoggle@series{##1}{#1}{##2}[#2]%
4893   }%
4894 }%
4895 %

```

hookcommand@series@reload `\newhookcommand@series@reload` does the same thing as `\newhookcommand@series` but the commands created by this macro also reload the series' arrangement.

```

4896 \newcommand{\newhookcommand@series@reload}[2]{%
4897   \global\expandafter\newcommand\expandafter*\csname #1\endcsname[2][]{%
4898     \setcommand@series{##1}{#1}{##2}[#2]%
4899   }%
4900 }
4901 %

```

XXI.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.

```

4902 \unless\ifnocritical@
4903   \newhookcommand@series{Xwrapcontent}%
4904   \newhookcommand@series{Xbeforeinserting}%

```

```

4905 \newhookcommand@series{Xlemmafont}%
4906 \newhookcommand@series{Xwraplemma}%
4907 \newhooktoggle@series{Xparindent}
4908 \newhookcommand@series{Xhangindent}
4909 \newhookcommand@series{Xragged}
4910 \newhookcommand@series{Xhsizetwocol}
4911 \newhookcommand@series{Xhsizethreecol}
4912 \newhookcommand@series{Xcolalign}%
4913 \newhookcommand@series{Xnotenumfont}
4914 \newhookcommand@series{Xbhooknote}
4915 \newhookcommand@series@reload{Xbhookgroup}{critical}
4916 \newhookcommand@series{Xboxsymlinenum}%
4917 \newhookcommand@series{Xsymlinenum}
4918 \newhookcommand@series{Xbeforenumber}
4919 \newhookcommand@series{Xtxtbeforenumber}
4920 \newhookcommand@series{Xafternumber}
4921 \newhookcommand@series{Xbeforesymlinenum}
4922 \newhookcommand@series{Xaftersymlinenum}
4923 \newhookcommand@series{Xinplaceofnumber}
4924 \newhookcommand@series{Xlemmaseparator}
4925 \newhookcommand@series{Xbeforelemmaseparator}
4926 \newhookcommand@series{Xafterlemmaseparator}
4927 \newhookcommand@series{Xinplaceoflemmaseparator}
4928 \newhookcommand@series{Xtxtbeforenotes}
4929 \newhookcommand@series@reload{Xafterrule}{critical}
4930 \newhooktoggle@series{Xnumberonlyfirstinline}
4931 \newhooktoggle@series{Xnumberonlyfirstintwolines}
4932 \newhooktoggle@series{Xnonumber}
4933 \newhooktoggle@series{Xpstart}
4934 \newhooktoggle@series{Xpstarteverytime}%
4935
4936 \newhooktoggle@series{Xstanza}%
4937 \newhookcommand@series{Xstanzaseparator}%
4938
4939 \newhooktoggle@series{Xonlypstart}
4940 \newhooktoggle@series{Xnonbreakableafternumber}
4941 \newhooktoggle@series{Xlemmadisablefontselection}
4942 \newhookcommand@series@reload{Xmaxhnotes}{critical}
4943 \newhookcommand@series@reload{Xbeforenotes}{critical}
4944 \newhooktoggle@series@reload{Xnoteswidthliketwocolumns}{critical}%
4945 \newhookcommand@series@reload{Xnotefontsize}{critical}
4946
4947 \newhookcommand@series{Xboxlinenum}%
4948 \newhookcommand@series{Xboxlinenumalign}%
4949
4950 \newhookcommand@series{Xboxstartlinenum}%
4951 \newhookcommand@series{Xboxendlinenum}%
4952
4953 \newhookcommand@series{Xafternote}%
4954 \newhookcommand@series{Xparafootsep}

```

```

4955 \newhookcommand@series@reload{Xwidth}{critical}%
4956
4957 \ifundef{\Xhsize}%
4958 {%
4959   \newcommandx{\Xhsize}[2][1,usedefault]{%
4960     \led@warning@Xhsize@deprecated%
4961     \Xwidth[#1]{#2}%
4962   }%
4963 }%
4964 {}%
4965 \fi
4966 \newhooktoggle@series{Xlineflag}[appref,Seref]
4967 \newhookcommand@series{Xtwolines}[appref,Seref]
4968 \newhookcommand@series{Xmorethantwolines}[appref,Seref]
4969 \newhookcommand@series{Xsublinesep}[appref,Seref,side]
4970 \newhooktoggle@series{Xtwolinesbutnotmore}[appref,Seref]
4971 \newhooktoggle@series{Xtwolinesonlyinsamepage}[appref,Seref]
4972 \newhookcommand@series{Xlinerrangeseparator}[appref,Seref]
4973 %
4974 %

```

XXI.3.5 Options for familiar notes

Before generating the optional commands for familiar notes, we check the `\nofamiliar` option.

```

4975 \unless\ifnofamiliar@
4976 \newhookcommand@series{wrapcontentX}%
4977 \newhookcommand@series{beforeinsertingX}%
4978 \newhooktoggle@series{parindentX}
4979 \newhookcommand@series{hangindentX}
4980 \newhookcommand@series{raggedX}
4981 \newhookcommand@series{hsizetwocolX}
4982 \newhookcommand@series{hsizethreecolX}
4983 \newhookcommand@series{colalignX}%
4984 \newhookcommand@series{notenunfontX}
4985 \newhookcommand@series{bhooknoteX}
4986 \newhookcommand@series@reload{bhookgroupX}{familiar}
4987 \newhookcommand@series@reload{beforenotesX}{familiar}
4988 \newhookcommand@series@reload{maxhnotesX}{familiar}
4989 \newhooktoggle@series@reload{noteswidthliketwocolumnsX}{familiar}%
4990 \newhookcommand@series@reload{afterruleX}{familiar}
4991 \newhookcommand@series@reload{notefontsizeX}{familiar}
4992 \newhookcommand@series{afternoteX}
4993 \newhookcommand@series{parafootsepX}
4994 \newhookcommand@series@reload{widthX}{familiar}%
4995 \ifundef{\hsizeX}%
4996 {%
4997   \newcommandx{\hsizeX}[2][1,usedefault]{%
4998     \led@warning@hsizeX@deprecated%

```

```

4999     \widthX[#1]{#2}%
5000   }%
5001 }%
5002 {}%
5003 \fi
5004 %

```

XXI.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.

```

5005 \unless\ifnoend@
5006   \newhookcommand@series{Xendwraplemma}
5007   \newhookcommand@series{Xendwrapcontent}
5008   \newhookcommand@series{Xendnotenumfont}
5009   \newhookcommand@series{Xendlemmafont}%
5010   \newhookcommand@series{Xendbhooknote}
5011
5012   \newhookcommand@series{Xendboxlinenum}%
5013   \newhookcommand@series{Xendboxlinenumalign}%
5014
5015   \newhookcommand@series{Xendboxstartlinenum}%
5016   \newhookcommand@series{Xendboxendlinenum}%
5017
5018   \newhookcommand@series{Xendnotefontsize}
5019   \newhooktoggle@series{Xendlemmadisablefontselection}
5020   \newhookcommand@series{Xendlemmaseparator}
5021   \newhookcommand@series{Xendbeforelemmaseparator}
5022   \newhookcommand@series{Xendafterlemmaseparator}
5023   \newhookcommand@series{Xendinplaceoflemmaseparator}
5024
5025   \newhookcommand@series{Xendbeforenumber}%
5026   \newhookcommand@series{Xendafternumber}%
5027
5028   \newhooktoggle@series{Xendparagraph}
5029   \newhookcommand@series{Xendafternote}
5030   \newhookcommand@series{Xendsep}
5031
5032   \newhookcommand@series{Xendinplaceofnumber}%
5033   \newhooktoggle@series{Xendnonumber}%
5034
5035   \newhooktoggle@series{Xendnumberonlyfirstinline}%
5036   \newhooktoggle@series{Xendnumberonlyfirstintwolines}%
5037
5038   \newhookcommand@series{Xendsymlinenum}%
5039   \newhookcommand@series{Xendbeforesymlinenum}%
5040   \newhookcommand@series{Xendaftersymlinenum}%
5041   \newhookcommand@series{Xendboxsymlinenum}%
5042

```

```

5043 \newhookcommand@series{Xendbhooklinenumber}%
5044 \newhookcommand@series{Xendahooklinenumber}%
5045 \newhookcommand@series{Xendbhookinplaceofnumber}%
5046 \newhookcommand@series{Xendahookinplaceofnumber}%
5047
5048 \newhookcommand@series{Xendhangindent}%
5049
5050
5051 \fi
5052 \newhooktoggle@series{Xendlineflag}[apprefwithpage,SErefwithpage]
5053 \newhookcommand@series{Xendtwolines}[apprefwithpage,SErefwithpage]
5054 \newhookcommand@series{Xendmoreethantwolines}[apprefwithpage,SErefwithpage]
5055 \newhooktoggle@series{Xendtwolinesbutnotmore}[apprefwithpage,SErefwithpage]
5056 \newhooktoggle@series{Xendtwolinesonlyinsamepage}[apprefwithpage,
SErefwithpage]
5057 \newhookcommand@series{Xendlinerangeseparator}[apprefwithpage,SErefwithpage
]
5058 \newhookcommand@series{Xendbeforepagenumber}[apprefwithpage,SErefwithpage,
SErefonlypage]
5059 \newhookcommand@series{Xendafterpagenumber}[apprefwithpage,SErefwithpage]
5060 \newhookcommand@series{Xendlineprefixsingle}[apprefwithpage,SErefwithpage]
5061 \newhookcommand@series{Xendlineprefixmore}[apprefwithpage,SErefwithpage]
5062 \newhookcommand@series{Xendsublinesep}[apprefwithpage,SErefwithpage]
5063
5064 %

```

XXI.4 Hooks for a particular footnote

\newhooktoggle@specific \newhooktoggle@specific is a generic command to create boolean hook specific to a note.

```

5065 \newcommand{\newhooktoggle@specific}[1]{%
5066 \newtoggle{#1}%
5067 \define@key{mac}{truefootnoteoption}{#1}[]{\global\settoggle{#1}{true}}%
When enabling footnote option
5068 \define@key{mac}{falsefootnoteoption}{#1}[]{\global\settoggle{#1}{false
}}
5069 }
5070 %

```

\newhookarg@specific \newhookarg@specific is a generic command to create argumen hook specific to a note.

```

5071 \newcommand{\newhookarg@specific}[1]{%
5072 \define@key{mac}{truefootnoteoption}{#1}{\global\def\linrangesep@{##1}}%
When enabling footnote option
5073 \define@key{mac}{falsefootnoteoption}{#1}{\global\undef\linrangesep@}%
When
5074 }
5075 %

```

And now, we define some hooks specific to a note.

```
5076 \newhooktoggle@specific{fulllines}%
5077 \newhooktoggle@specific{nonum}
5078 \newhooktoggle@specific{nosep}
5079 \newhookarg@specific{linrangesep}
5080 %
```

linrangesep@ \linrangesep@ is defined by the option `linrangesep` of critical notes to change temporarily the line range separator for a specific line. As we have to define it before typesetting the line and undefine it after, we use the family of `xkeyval` package's key.

```
5081 %
```

\nomk@ \nomk@ toggle is used by `reledpar` to remove the footnote mark in the text when using `\footnoteXmk`. Read `reledpar` handbook.

```
5082 \newtoggle{nomk}%
5083 %
```

XXI.5 Alias

\Xnolemmaseparator \Xnolemmaseparator[*series*] is just an alias for `\Xlemmaseparator[series]{}`.

```
5084 \newcommandx*{\Xnolemmaseparator}[1][1]{\Xlemmaseparator[#1]}
5085 %
```

XXII Output routine

Now we begin the output routine and associated things.

XXII.1 Page number management

\pageno \pageno is a page number, starting at 1, and `\advancepageno` increments the number.
\advancepageno

```
5086 \countdef\pageno=0 \pageno=1
5087 \newcommand*{\advancepageno}{\ifnum\pageno<\z@ \global\advance\pageno\m@ne
5088 \else\global\advance\pageno\@ne\fi}
5089
5090 %
```

XXII.2 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 \LaTeX are first, then familiar familiar footnotes and finally the critical footnotes.

```

5091 \newcommand*{\l@ddoxtrafeet}{%
5092   \IfStrEq{familiar-critical}{\@fnpos}
5093     {\do@feetX\Xdo@feet}%
5094     {%
5095       \IfStrEq{critical-familiar}{\@fnpos}%
5096         {\Xdo@feet\do@feetX}%
5097         {\do@feetX\Xdo@feet}%
5098     }%
5099 }%
5100
5101 %

```

`\Xdo@feet` `\Xdo@feet` is the code extending `\@makecol` to cater for the extra critical feet.

```

5102 \newcommand*{\Xdo@feet}{%
5103   \setbox\@outputbox \vbox{%
5104     \unvbox\@outputbox
5105     \@opXfeet}}
5106 %

```

`\@opXfeet` The extra critical feet to be added to the output. The normal way to add one series, `\print@Xnotes` `\print@Xnotes`, is replaced by `reledpar` when using `\Pages`.

```

5107 \newcommand\print@Xnotes[1]{%
5108   \xdef\@currentseries{#1}%
5109   \csuse{#1footstart}{#1}%
5110   \csuse{#1footgroup}{#1}%
5111 }%
5112 %

```

We print all series of notes by looping on them. We check before printing them that they are not voided.

```

5113 \newcommand*{\@opXfeet}{%
5114   \unless\ifnocritical%
5115     \gdef\firstXseries{}%
5116     \def\do##1{%
5117       \ifvoid\csuse{##1footins}\else%
5118         \global\skip\csuse{##1footins}=\csuse{Xbeforenotes@##1}%
5119         \global\advance\skip\csuse{##1footins} by\csuse{Xafterrule@##1}%
5120         \print@Xnotes{##1}%
5121       \fi%
5122     }%
5123     \dolistloop{\@series}%
5124   \fi%
5125 }%
5126 %

```

`\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`.

```

5127 \newcommand*{\l@ddodoreintrafeet}{%
5128   \IfStrEq{familiar-critical}{\@fnpos}
5129   {\@doreinfeetX\X@doreinfeet}%
5130   {%
5131     \IfStrEq{critical-familiar}{\@fnpos}%
5132     {\X@doreinfeet\@doreinfeetX}%
5133     {\@doreinfeetX\X@doreinfeet}%
5134   }%
5135 }
5136
5137 %

```

`\X@doreinfeet` `\X@doreinfeet` is the code for catering for the extra critical footnotes within `\@reinserts`.

```

5138 \newcommand*{\X@doreinfeet}{%
5139   \unless\ifnocritical@%
5140   \def\do##1{%
5141     \ifvoid\csuse{##1footins}\else%
5142     \insert\csuse{##1footins}{\unvbox\csuse{##1footins}}%
5143     \fi}%
5144   \dolistloop{\@series}
5145   \fi%
5146 }
5147
5148 %

```

`\print@notesX` We have to add all the new kinds of familiar footnotes to the output routine. The normal way to add one series. `\print@Xnotes` is replaced by `reledpar` when using `\Pages`.

`\do@feetX`

`\@doreinfeetX`

```

5149 \newcommand\print@notesX[1]{%
5150   \xdef\@currentseries{#1}%
5151   \csuse{footstart#1}{#1}%
5152   \csuse{footgroup#1}{#1}%
5153 }%
5154 %

```

We print all the series of notes by looping on them. We check before printing them that they are not voided.

```

5155 \newcommand*{\do@feetX}{%
5156   \unless\ifnofamiliar@%
5157   \gdef\firstseriesX@{}%
5158   \setbox\@outputbox \vbox{%
5159     \unvbox\@outputbox%
5160     \def\do##1{%
5161       \ifvoid\csuse{footins##1}\else%
5162       \global\skip\csuse{footins##1}=\csuse{beforenotesX@##1}%
5163       \global\advance\skip\csuse{footins##1} by\csuse{afterruleX@##1}%

```

```

5164         \print@notesX{##1}%
5165     \fi%
5166 }%
5167 \dolistloop{\@series}}%
5168 \fi%
5169 }%
5170
5171 \newcommand{\@doreinfeetX}{%
5172     \unless\ifnofamiliar%
5173     \def\do##1{%
5174         \ifvoid\csuse{footins##1}\else
5175         \insert%
5176             \csuse{footins##1}
5177             {\unvbox\csuse{footins##1}}%
5178         \fi%
5179     }%
5180     \dolistloop{\@series}%
5181 \fi%
5182 }%
5183
5184 %

```

XXII.3 Patching standard output of 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.)

```

5185 \@ifclassloaded{memoir}{%
5186 %

```

memoir is loaded so we use memoir’s built in hooks.

```

5187 \g@addto@macro{\m@mdoextrafeet}{\l@ddoxtrafeet}%
5188 \g@addto@macro{\m@mdodoreinextrafeet}{\l@ddodoreinextrafeet}%
5189 }{%
5190 %

```

memoir has not been loaded, so patch `\@makecol` and `\@reinserts`.

```

5191 \@ifpackageloaded{fancyhdr}{%
5192 \patchcmd%
5193     {\latex@makecol}%
5194     {\xdef\@freelist{\@freelist\@midlist}}%
5195     {\xdef\@freelist{\@freelist\@midlist}\l@ddoxtrafeet}%
5196     {}%
5197     {\led@error@fail@patch@makecol}%
5198 }{%
5199 \patchcmd%
5200     {\@makecol}%

```

```

5201     {\xdef\@freelist{\@freelist\@midlist}}}%
5202     {\xdef\@freelist{\@freelist\@midlist}\l@ddoxtrafeet}%
5203     }%
5204     {\led@error@fail@patch@@makecol}%
5205   }%
5206
5207   \patchcmd%
5208     {\@reinserts}%
5209     {\ifvbox}%
5210     {\l@ddodoreinextrafeet\ifvbox}%
5211     {}%
5212     {\led@error@fail@patch@@reinserts}}%
5213 }
5214
5215 %

```

It turns out that `\@doclearpage` also needs modifying.

`\if@led@nofoot` We have to check if there are any leftover feet.

```

5216 \newif\if@led@nofoot
5217
5218 %

```



```

5219 \@ifclassloaded{memoir}{%
5220 %

```

If the memoir class is loaded we hook into its modified `\@doclearpage`.

```

\@mem@extranofeet 5221 \g@addto@macro{\@mem@extranofeet}{%%
5222   \def\do#1{%
5223     \unless\ifnocritical@
5224       \ifvoid\csuse{#1footins}\else\@mem@nofootfalse\fi%
5225     \fi%
5226     \unless\ifnofamiliar@
5227       \ifvoid\csuse{footins#1}\else\@mem@nofootfalse\fi%
5228     \fi%
5229   }
5230   \dolistloop{\@series}%
5231 }%
5232 }{%
5233 %

```

As memoir is not loaded we have patch `\@doclearpage`.

```

\@led@testifnofoot 5234 \newcommand*{\@led@testifnofoot}{%
\@doclearpage 5235   \@led@nofoottrue%
5236   \ifvoid\footins\else%
5237     \@led@nofootfalse%
5238   \fi%

```

```

5239 \def\do##1{%
5240   \unless\ifnocritical@%
5241     \ifvoid\csuse{##1footins}\else%
5242       \@led@nofootfalse%
5243     \fi%
5244   \fi%
5245   \unless\ifnofamiliar@%
5246     \ifvoid\csuse{footins##1}\else%
5247       \@led@nofootfalse%
5248     \fi%
5249   \fi%
5250 }%
5251 \dolistloop{\@series}%
5252 }%
5253
5254 \pretocmd%
5255 {\@docclearpage}%
5256 {\@led@testifnofoot}%
5257 {}%
5258 {\led@error@fail@patch@@docclearpage}%
5259
5260 \patchcmd%
5261 {\@docclearpage}%
5262 {\ifvoid\footins}%
5263 {\if@led@nofoot}%
5264 {}%
5265 {\led@error@fail@patch@@docclearpage}%
5266
5267 }
5268
5269 %

```

XXIII 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.

```

5270 \list@create{\labelref@list}

```

```
5271 %
```

\zz@@@ A convenience macro to zero two labeling counters in one go.

```
5272 \newcommand*{\zz@@@}{000|000} % set two counters to zero in one go
5273
5274 %
```

\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.³¹

This version of the original `edmac \label` uses `\@bsphack` and `\@esphack` to eliminate extra space problems and also use the \TeX write methods for the `.aux` file.

Jesse Billett³² found that the original code could be off by several pages. This version, hopefully cures that, and also allows for non-arabic page numbering.

```
5275 \newcommand*{\edlabel}[1]{%
5276   \ifl@dpairing\ifautopar%
5277     \strut%
5278     \fi\fi%
5279     \@bsphack%
5280     \ifboolexpr{bool{ledRcol} or bool{ledRcol}}{%
5281       \ifXnote%
5282         \protected@write\@auxout{%
5283           {\string\l@dmake@labelsR\space\thepage|\l@dparsedstartline|\l@dparsedstartsub|\the\c@pstartR|{#1}}%
5284           \ifdef{\hypertarget}%
5285             {\Hy@raisedlink{\hypertarget{#1}{}}}%
5286             {}%
5287         \else%
5288           \write\linenum@outR{\string\@lab}%
5289           \ifx\labelref@listR\empty%
5290             \xdef\label@refs{\zz@@@}%
5291           \else%
5292             \gl@p\labelref@listR\to\label@refs%
5293           \fi%
5294           \ifvmode%
5295             \advancelabel@refs%
5296           \fi%
5297     %
```

Use code from the kernel `\label` command to write the correct page number. Also define an `hypertarget` if `hyperref` package is loaded.

³¹The remaining macros in this section were kindly revised by Wayne Sullivan, who substantially improved their efficiency and flexibility.

³²(jdb43@cam.ac.uk) via the `ctt` thread ‘`ledmac cross referencing`’, 25 August 2003.

```

5298 \protected@write\@auxout{}\%
5299 {\string\l@dmake@labelsR\space\thepage|\label@refs|\the\c@pstartR
|{#1}}}%
5300 \ifdef\hypertarget{%
5301 {\Hy@raisedlink{\hypertarget{#1}{}}}%
5302 {}}%
5303 \fi%
5304 }{%
5305 \ifXnote@%
5306 \protected@write\@auxout{}\%
5307 {\string\l@dmake@labels\space\thepage|\l@dparsedstartline|\
l@dparsedstartsub|\the\c@pstart|{#1}}}%
5308 \ifdef\hypertarget{%
5309 {\Hy@raisedlink{\hypertarget{#1}{}}}%
5310 {}}%
5311 \else%
5312 \write\linenum@out{\string\@lab}%
5313 \ifx\labelref@list\empty%
5314 \xdef\label@refs{\zz@@@}%
5315 \else%
5316 \gl@p\labelref@list\to\label@refs%
5317 \fi%
5318 \ifvmode%
5319 \advancelabel@refs%
5320 \fi%
5321 \protected@write\@auxout{}\%
5322 {\string\l@dmake@labels\space\thepage|\label@refs|\the\c@pstart
|{#1}}}%
5323 \ifdef\hypertarget{%
5324 {\Hy@raisedlink{\hypertarget{#1}{}}}%
5325 {}}%
5326 \fi%
5327 }%
5328 \@esphack}%
5329
5330 %

```

`\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.

`\labelrefsparseline` Hence, we need to test `\edlabel` if it occurs at the start of a paragraph. To do so, we

`\labelrefsparsesubline` 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.

```

5331 \newcounter{line}%
5332 \newcounter{subline}%
5333 \newcommand{\advancelabel@refs}{%
5334 \setcounter{line}{\expandafter\labelrefsparseline\label@refs}%
5335 \stepcounter{line}%
5336 \ifsublines@%

```

```

5337 \setcounter{subline}{\expandafter\labelrefsparsesubline\label@refs}
5338 %
5339 \stepcounter{subline}{1}%
5340 \def\label@refs{\theline|\thesubline}%
5341 \else%
5342 \def\label@refs{\theline|0}%
5343 \fi%
5344 }
5345 \def\labelrefsparseline#1|#2{#1}
5346 \def\labelrefsparsesubline#1|#2{#2}
5347 %

```

\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 pstart number, #5 label.

```

5347 \newcommand*\l@dmake@labels{}
5348 \def\l@dmake@labels#1|#2|#3|#4|#5{%
5349 \expandafter\ifx\csname the@label\csuse{XR@prefix}#5\endcsname \relax\
5350 else
5351 \led@warn@DuplicateLabel{\csuse{XR@prefix}#5}%
5352 \fi
5353 \expandafter\gdef\csname the@label\csuse{XR@prefix}#5\endcsname
5354 {#1|#2|#3|#4|\relax}%
5355 \ignorespaces}
5356 %

```

ℒ_{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.

```

5356 \AtBeginDocument{%
5357 \def\l@dmake@labels#1|#2|#3|#4|#5{%
5358 }
5359 }
5360 %

```

\@lab The `\@lab` command, which appears in the `\linenum@out` file, appends the current values of page, line and sub-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.

ℒ_{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`.


```

5361 \newcommand*{\@lab}{%
5362   \ifledRcol
5363     \xright@appenditem{\linenumr@p{\line@numR}}{%
5364       \ifsublines@ \sublinenumr@p{\subline@numR}\else 0\fi}%
5365     \to\labelref@listR
5366   \else
5367     \xright@appenditem{\linenumr@p{\line@num}}{%
5368       \ifsublines@ \sublinenumr@p{\subline@num}\else 0\fi}%
5369     \to\labelref@list
5370   \fi}
5371 \fi}
5372 %

```

`\applabel` `\applabel`, if called in `\edtext` will insert automatically both a start and an end label for the current edtext lines.

```

5373 \newcommand*{\applabel}[1]{%
5374   \ifnum\@edtext@level>0%
5375   %

```

Label should not be already defined.

```

5376   \ifcsundef{the@label#1}{%
5377     \csdef{the@label#1}{\applabel}%
5378   }%
5379   {%
5380     \led@warn@DuplicateLabel{#1 (\applabel)}%
5381   }%
5382 %

```

Parse the `\edtext` line numbers.

```

5383   \expandafter\l@dp@rsefootspec\l@d@nums|%
5384 %

```

Use the \TeX standard hack for label.

```

5385   \@bsphack%
5386 %

```

And now, write the data in the auxiliary file.

```

5387   \ifledRcol%
5388     \protected@write\@auxout{}%
5389     {\string\l@dmake@labelsR\space\l@d@p@rsestartpage|\
5390 \l@d@p@rsestartline|\l@d@p@rsestartsub|\the\c@pstartR|{#1:start}}%
5391     \ifdef{\hypertarget}%
5392     {\Hy@raisedlink{\hypertarget{#1:start}}{}}%
5393     {}%
5394     \protected@write\@auxout{}%
5395     {\string\l@dmake@labelsR\space\l@d@p@rseendpage|\l@d@p@rseendline
5396 |\l@d@p@rseendsub|\the\c@pstartR|{#1:end}}%
5397   \else%

```

```

5396     \protected@write\@auxout{}\%
5397     {\string\l@dmake@labels\space\l@dparsedstartpage|\
l@dparsedstartline|\l@dparsedstartsub|\the\c@pstart|{#1:start}}}%
5398     \ifdef{\hypertarget}%
5399     {\Hy@raisedlink{\hypertarget{#1:start}}}%
5400     }%
5401     \protected@write\@auxout{}\%
5402     {\string\l@dmake@labels\space\l@dparsedendpage|\l@dparsedendline
|\l@dparsedendsub|\the\c@pstart|{#1:end}}}%
5403     \fi%
5404     %

```

Use the \LaTeX standard hack for label.

```

5405     \@esphack%
5406     %

```

Warning if `\applabel` is called outside of `\edtext`.

```

5407     \else%
5408     \led@warn@AppLabelOutEdtext{#1}%
5409     \fi%
5410     %

```

End of `\applabel`

```

5411 }%
5412 %

```

`\edlabels` `\edlabels` and `\edlabelE` are just used to mark the beginning and the end of a passage.

```

\edlabelE
\edlabelSE
5413 \newcommand{\edlabels}[1]{%
5414     \edlabel{#1:start}%
5415 }
5416 \newcommand{\edlabelE}[1]{%
5417     \edlabel{#1:end}%
5418 }
5419 \newcommand{\edlabelSE}[1]{%
5420     \edlabels{#1}%
5421     \edlabelE{#1}%
5422 }
5423 %

```

`\wrap@edcrossref` `\wrap@edcrossref` is called around all `reledmac` crossref commands, except those which start with `x`. It adds the hyperlink.

```

5424 \newrobustcmd{\wrap@edcrossref}[2]{%
5425     \ifdef{\hyperlink}%
5426     {\hyperlink{#1}{#2}}%
5427     {#2}%
5428 }
5429 %

```

\edpageref If the specified label exists, \edpageref gives its page number.

\xpageref For this reference command, as for the other two, a special version with prefix x is provided for use in places where the command is to be scanned as a number, as in \linenum. These special versions have two limitations: they do not print error messages if the reference is unknown, and they can't appear as the first label or reference command in the file; you must ensure that a \edlabel or a normal reference command appears first, or these x-commands will always return zeros.

TeX already defines a \pageref, so changing the name to \edpageref.

```
5430 \newcommand*\edpageref}[1]{\l@dref@undefined{#1}\wrap@edcrossref{#1}{\l@dgetref@num{1}{#1}}}%
5431 \newcommand*\xpageref}[1]{\l@dgetref@num{1}{#1}}%
5432
5433 %
```

\edlineref If the specified label exists, \lineref gives its line number.

\xlineref

```
5434 \newcommand*\edlineref}[1]{\l@dref@undefined{#1}\wrap@edcrossref{#1}{\l@dgetref@num{2}{#1}\xflagref{#1}}}%
5435 \newcommand*\xlineref}[1]{\l@dgetref@num{2}{#1}}%
5436
5437 %
```

\sublineref If the specified label exists, \sublineref gives its sub-line number.

\xsublineref

```
5438 \newcommand*\sublineref}[1]{\l@dref@undefined{#1}\wrap@edcrossref{#1}{\l@dgetref@num{3}{#1}}}%
5439 \newcommand*\xsublineref}[1]{\l@dgetref@num{3}{#1}}%
5440
5441 %
```

\pstarteref If the specified label exists, \pstartref gives its pstart number.

\xpstartref

```
5442 \newcommand*\pstartref}[1]{\l@dref@undefined{#1}\wrap@edcrossref{#1}{\l@dgetref@num{4}{#1}}}%
5443 \newcommand*\xpstartref}[1]{\l@dgetref@num{4}{#1}}%
5444
5445 %
```

\xflagref \xflagref finds the side flag of any ref defined with \edlabel.

```
5446 \newcommand*\xflagref}[1]{\l@dgetref@num{5}{#1}}%
5447 %
```

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.

```

5448 \newcommand*{\l@dref@undefined}[1]{%
5449   \expandafter\ifx\csname the@label#1\endcsname\relax
5450     \led@warn@RefUndefined{#1}%
5451   \fi}
5452
5453 %

```

\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.

```

5454 \newcommand*{\l@dgetref@num}[2]{%
5455   \expandafter
5456   \ifx\csname the@label#2\endcsname \relax
5457     000%
5458   \else
5459     \expandafter\expandafter\expandafter
5460     \l@dlabel@parse\csname the@label#2\endcsname|#1%
5461   \fi}
5462
5463 %

```

\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 or 4) which defines which of the earlier five numbers to pick out. (It was earlier given as the first argument of `\l@dgetref@num`.)

```

5464 \newcommand*{\l@dlabel@parse}{%
5465   \def\l@dlabel@parse#1|#2|#3|#4|#5|#6{%
5466     \ifcase #6%
5467       \or #1%
5468       \or #2%
5469       \or #3%
5470       \or #4%
5471       \or #5%
5472     \fi}
5473 %

```

\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.

```

5474 \newcommand*{\xxref}[2]{%
5475   {%
5476     \expandafter\ifx\csname the@label#1\endcsname \relax%
5477       \expandafter\let\csname the@@label#1\endcsname\zz@@@%
5478     \else%
5479       \expandafter\def\csname the@@label#1\endcsname{\l@dgetref@num
5480 {1}{#1}|\l@dgetref@num{2}{#1}|\l@dgetref@num{3}{#1}}%
5481     \fi%
5482     \expandafter\ifx\csname the@label#2\endcsname \relax%
5483       \expandafter\let\csname the@@label#2\endcsname\zz@@@%
5484     \else%
5485       \expandafter\def\csname the@@label#2\endcsname{\l@dgetref@num
5486 {1}{#2}|\l@dgetref@num{2}{#2}|\l@dgetref@num{3}{#2}}%
5487     \fi%
5488     \letcs{\@tempa}{the@@label#1}%
5489     \letcs{\@tempb}{the@@label#2}%
5490     \linenum{\@tempa|
5491     \@tempb}}}%

```

`\appref` `\appref`, `\Seref`, `\apprefwithpage`, `\Serefwithpage` and `\SEonlypage` print cross-ref to some start / end lines defined by specific commands. It prints the lines as they should be printed in the apparatus (critical notes for not suffixed versions, endnotes for suffixed versions).

`\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`.

```

5492 \def\Xtwolines@appref{%
5493 \def\Xtwolines@Seref{%
5494
5495 \def\Xmorethantwolines@appref{%
5496 \def\Xmorethantwolines@Seref{%
5497
5498 \def\Xlinerangeseparator@appref{\endashchar}%
5499 \def\Xlinerangeseparator@Seref{\endashchar}%
5500
5501 \def\Xsublinesep@appref{\fullstop}%
5502 \def\Xsublinesep@Seref{\fullstop}%
5503
5504 \newtoggle{Xtwolinesbutnotmore@appref}%

```

```

5505 \newtoggle{Xtwolinesbutnotmore@SEref}%
5506
5507 \newtoggle{Xtwolinesonlyinsamepage@appref}%
5508
5509 \newtoggle{Xtwolinesonlyinsamepage@SEref}%
5510
5511 \newtoggle{Xlineflag@appref}%
5512 \toggletrue{Xlineflag@appref}%Here exception
5513 \newtoggle{Xlineflag@SEref}%
5514 \toggletrue{Xlineflag@SEref}%Here exception
5515
5516 \def\Xendtwolines@apprefwithpage{ }%
5517 \def\Xendtwolines@SErefwithpage{ }%
5518
5519 \def\Xendmoreethantwolines@apprefwithpage{ }%
5520 \def\Xendmoreethantwolines@SErefwithpage{ }%
5521
5522 \def\Xendlinerangeseparator@apprefwithpage{\endashchar}
5523 \def\Xendlinerangeseparator@SErefwithpage{\endashchar}
5524 \def\Xendlinerangeseparator@SErefonlypage{\endashchar}
5525
5526 \def\Xendbeforepagenumber@apprefwithpage{p.}%
5527 \def\Xendbeforepagenumber@SErefwithpage{p.}%
5528 \def\Xendbeforepagenumber@SEonlypage{p.}%
5529
5530 \def\Xendafterpagenumber@apprefwithpage{ )}%
5531 \def\Xendafterpagenumber@SErefwithpage{ )}%
5532
5533
5534 \def\Xendlineprefixsingle@apprefwithpage{ }%
5535 \def\Xendlineprefixsingle@SErefwithpage{ }%
5536
5537 \def\Xendlineprefixmore@apprefwithpage{ }%
5538 \def\Xendlineprefixmore@SErefwithpage{ }%
5539
5540 \newtoggle{Xendtwolinesbutnotmore@apprefwithpage}%
5541 \newtoggle{Xendtwolinesbutnotmore@SErefwithpage}%
5542
5543 \def\Xendsublinesep@apprefwithpage{\fullstop}%
5544 \def\Xendsublinesep@SErefwithpage{\fullstop}%
5545
5546 \newtoggle{Xendtwolinesonlyinsamepage@apprefwithpage}%
5547 \newtoggle{Xendtwolinesonlyinsamepage@SErefwithpage}%
5548
5549 \newtoggle{Xendlineflag@apprefwithpage}
5550 \toggletrue{Xendlineflag@apprefwithpage}%Here, exception
5551 \newtoggle{Xendlineflag@SErefwithpage}
5552 \toggletrue{Xendlineflag@SErefwithpage}%Here, exception
5553
5554 %

```

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.

```

5555
5556 \xdef\Xboxstartlinenum@appref{Opt}
5557 \xdef\Xboxstartlinenum@SEref{Opt}
5558
5559 \xdef\Xboxendlinenum@appref{Opt}
5560 \xdef\Xboxendlinenum@SEref{Opt}
5561
5562 \xdef\Xendboxstartlinenum@apprefwithpage{Opt}
5563 \xdef\Xendboxstartlinenum@SErefwithpage{Opt}
5564
5565 \xdef\Xendboxendlinenum@apprefwithpage{Opt}
5566 \xdef\Xendboxendlinenum@SErefwithpage{Opt}
5567
5568 %

```

Now, declare the default values of `\@apprefprefixsingle` and `\@apprefprefixmore`, `\@SErefprefix`, `\@SErefprefixmore` and the commands which defines them.

```

5569 \newcommand\@apprefprefixsingle{}%
5570 \newcommand\@SErefprefixsingle{}%
5571
5572 \newcommand\@apprefprefixmore{}%
5573 \newcommand\@SErefprefixmore{}%
5574
5575 \newcommand{\setapprefprefixsingle}[1]{%
5576   \gdef\@apprefprefixsingle{#1}%
5577 }
5578 \newcommand{\setSErefprefixsingle}[1]{%
5579   \gdef\@SErefprefixsingle{#1}%
5580 }
5581
5582 \newcommand{\setapprefprefixmore}[1]{%
5583   \gdef\@apprefprefixmore{#1}%
5584 }
5585 \newcommand{\setSErefprefixmore}[1]{%
5586   \gdef\@SErefprefixmore{#1}%
5587 }
5588
5589 %

```

And not `\setSErefonlypageprefixsingle` and `\setSErefonlypageprefixmore`.

```

5590 \let\setSErefonlypageprefixsingle\XendbeforepagenumberSErefonlypage%
5591 \newcommand{\setSErefonlypageprefixmore}[1]{%
5592   \gdef\SErefonlypage@prefixmore{#1}%
5593 }%
5594 %

```

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.

```

5595
5596 \newcommandx{\appref}[2][1,usedefault]{\reformatted@{#1}{#2}{appref}}
5597 \newcommandx{\Seref}[2][1,usedefault]{\reformatted@{#1}{#2}{Seref}}
5598
5599 \newcommandx{\apprefwithpage}[2][1,usedefault]{\reformattedwithpage@
5600 {#1}{#2}{appref}}
5601 \newcommandx{\Serefwithpage}[2][1,usedefault]{\reformattedwithpage@
5602 {#1}{#2}{Seref}}
5603
5604 \newcommand{\reformatted@}[3]{%
5605   \def\do##1{%
5606     \setkeys[mac]{truefootnoteoption}{##1}%
5607   }%
5608   \notblank{#1}{\docsvlist{#1}}{}%
5609   \xdef\@currentseries{#3}%
5610   \ifcempty{@#3prefixmore}%
5611     {\@apprefprefixsingle}%
5612     {%
5613       \IfEq{\xlineref{#2:start}}{\xlineref{#2:end}}%
5614       {\csuse{@#3prefixsingle}}%
5615       {\csuse{@#3prefixmore}}%
5616     }%
5617   \ifboolexpr{%
5618     test{\ifcsundef{the@label#2:start}}%
5619     or test{\ifcsundef{the@label#2:end}}%
5620   }%
5621     {\led@warn@pairRefUndefined{#2}\nfss@text{\reset@font\bfseries ??}}%
5622     {%
5623       \def\@this@crossref@start{#2:start}%
5624       \def\@this@crossref@end{#2:end}%
5625       \printlines\xpageref{#2:start}|\xlineref{#2:start}|\xsublineref{#2:
5626 start}|\xpageref{#2:end}|\xlineref{#2:end}|\xsublineref{#2:end}|\relax|
5627 xflagref{#2:start}}|%
5628       \undef\@this@crossref@end%
5629       \undef\@this@crossref@start%
5630     }%
5631   \def\do##1{%
5632     \setkeys[mac]{falsefootnoteoption}{##1}%
5633   }%
5634   \notblank{#1}{\docsvlist{#1}}{}%
5635 }%

```



```

5635 \newcommand{\reformattedwithpage@}[3]{%
5636   \def\do##1{%
5637     \setkeys[mac]{truefootnoteoption}{##1}%
5638   }%
5639   \notblank{#1}{\docsvlist{#1}}{ }%
5640   \xdef\@currentseries{#3withpage}%
5641   \ifboolexpr{%
5642     test{\ifcsundef{the@label#2:start}}%
5643     or test{\ifcsundef{the@label#2:end}}%
5644   }%
5645   {\led@warn@pairRefUndefined{#2}\nfss@text{\reset@font\bfseries ??}}%
5646   {%
5647     \def\@this@crossref@start{#2:start}%
5648     \def\@this@crossref@end{#2:end}%
5649     \printendlines\xpageref{#2:start}|\xlineref{#2:start}|\xsublineref{#2:
start}|\xpageref{#2:end}|\xlineref{#2:end}|\xsublineref{#2:end}|\relax|\
xflagref{#2:start}|}%
5650     \undef\@this@crossref@end%
5651     \undef\@this@crossref@start%
5652   }%
5653   \def\do##1{%
5654     \setkeys[mac]{falsefootnoteoption}{##1}%
5655   }%
5656   \notblank{#1}{\docsvlist{#1}}{ }%
5657 }%
5658
5659 \newcommand{\reformattedonlypage@}[3]{%
5660   \def\do##1{%
5661     \setkeys[mac]{truefootnoteoption}{##1}%
5662   }%
5663   \notblank{#1}{\docsvlist{#1}}{ }%
5664   \xdef\@currentseries{#3onlypage}%
5665   \ifboolexpr{%
5666     test{\ifcsundef{the@label#2:start}}%
5667     or test{\ifcsundef{the@label#2:end}}%
5668   }%
5669   {\led@warn@pairRefUndefined{#2}\nfss@text{\reset@font\bfseries ??}}%
5670   {\ifnumequal{\xpageref{#2:end}}{\xpageref{#2:start}}%
5671     {%
5672       \printnpnum{%
5673         \wrap@edcrossref{#2:start}{\xpageref{#2:start}}}%
5674       }%
5675     }%
5676   {%
5677     \ifcsvoid{#3onlypage@prefixmore}%
5678     {}%
5679     {\csletcs{Xendbeforepagenumber@#3onlypage}{#3onlypage@prefixmore}}%
5680     \ifdefined\linerrangesep%
5681       \printnpnum{%
5682         \wrap@edcrossref{#2:start}{\xpageref{#2:start}}}%

```

```

5683 \linangesep@%
5684 \wrap@edcrossref{#2:end}{\xpageref{#2:end}}%
5685 }%
5686 \else%
5687 \printnpnum{%
5688 \wrap@edcrossref{#2:start}{\xpageref{#2:start}}%
5689 \csuse{Xendlinangeseparator@{\currentseries}}%
5690 \wrap@edcrossref{#2:end}{\xpageref{#2:end}}%
5691 }%
5692 \fi%
5693 }%
5694 }%
5695 \def\do##1{%
5696 \setkeys[mac]{falsefootnoteoption}{##1}%
5697 }%
5698 \notblank{#1}{\docsvlist{#1}}}%
5699 }%
5700 %

```

`\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`.

```

5701 \newcommand*\edmakelabel}[2]{\expandafter\xdef\csname the@label#1\
5702 endcsname{#2}}
5703 %

```

(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. 121 and V.9 p. 90), since `\xxref` makes a call to `\linenum` in order to do its work.)

XXIII.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`.

```

5704 \pretocmd{\XR@test}%
5705 {\XR@test@mac+++#1#2#3#4+++}%
5706 {}%
5707 {}%
5708 %

```

`\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`.

```
5709 \long\def\xR@test@mac+++#1+++{\XR@test@mac@test#1}
5710 %
```

`\XR@test@mac@test` And finally, `\XR@test@mac@test` does the job. This code is based on the `\XR@test` macro of the `xr` package. However, note that the `\XR@prefix` is not called here, but it is integrated directly in `\l@dmake@labels` and `\l@dmake@labelsR`.

```
5711 \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
5712 \ifx#1\l@dmake@labels%
5713 \l@dmake@labels#2%
5714 \else
5715 \ifx#1\l@dmake@labelsR%
5716 \l@dmake@labelsR #2%
5717 \fi%
5718 \fi%
5719 }%
5720 %
```

XXIV Side notes

Regular `\marginpar`s 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 `\marginpar`s in numbered text do not disturb the flow.

```
5721 \pretocmd{\@xympar}%
5722 {\ifnumberedpar@
5723 \led@warn@NoMarginpars
5724 \esphack
5725 \else}%
5726 {}%
5727 {}%
5728
5729 \apptocmd{\@xympar}%
5730 {\fi}%
5731 {}
5732 {}
5733
5734 %
```

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 `\l@dgetsidenote@margin`

numbers). `\l@dgetsidenote@margin` returns the number associated to side note margin:

left : 0

right : 1

outer : 2

inner : 3

```

5735 \newcount\sidenote@margin
5736 \newcommand*{\sidenotemargin}[1]{%
5737   \l@dgetsidenote@margin{#1}%
5738   \ifnum\@l@tempcntb>\m@ne
5739     \ifledRcol
5740       \global\sidenote@marginR=\@l@tempcntb
5741     \else
5742       \global\sidenote@margin=\@l@tempcntb
5743     \fi
5744   \fi}}
5745 \newcommand*{\l@dgetsidenote@margin}[1]{%
5746   \def\@tempa{#1}\def\@tempb{left}%
5747   \ifx\@tempa\@tempb
5748     \@l@tempcntb \z@
5749   \else
5750     \def\@tempb{right}%
5751     \ifx\@tempa\@tempb
5752       \@l@tempcntb \@ne
5753     \else
5754       \def\@tempb{outer}%
5755       \ifx\@tempa\@tempb
5756         \@l@tempcntb \tw@
5757       \else
5758         \def\@tempb{inner}%
5759         \ifx\@tempa\@tempb
5760           \@l@tempcntb \thr@@
5761         \else
5762           \led@warn@BadSidenotemargin
5763           \@l@tempcntb \m@ne
5764         \fi
5765       \fi
5766     \fi
5767   \fi}
5768 \sidenotemargin{right}
5769
5770 %

```

`\l@dlp@rbox` We need two boxes to store sidenote texts.
`\l@drp@rbox`

```

5771 \newbox\l@dlp@rbox
5772 \newbox\l@drp@rbox
5773
5774 %

```

`\ledlsnotewidth` These specify the width of the left/right boxes (initialised to `\marginparwidth`), their
`\ledrsnotewidth` distance from the text (initialised to `\linenumsep`), and the fonts used.

```

\ledlsnotesep
\ledrsnotesep
5775 \newdimen\ledlsnotewidth \ledlsnotewidth=\marginparwidth
5776 \newdimen\ledrsnotewidth \ledrsnotewidth=\marginparwidth
\ledlsnotefontsetup
5777 \newdimen\ledlsnotesep \ledlsnotesep=\linenumsep
\ledrsnotefontsetup
5778 \newdimen\ledrsnotesep \ledrsnotesep=\linenumsep
5779 \newcommand*\ledlsnotefontsetup{\raggedleft\footnotesize}
5780 \newcommand*\ledrsnotefontsetup{\raggedright\footnotesize}
5781
5782 %

```

`\ledleftnote` `\ledleftnote`, `\ledrightnote`, `\ledinnernote`, `\ledouternote` are the user com-
`\ledrightnote` mands for left, right, inner and outer sidenotes. The two last one are just alias for the
`\ledinnernote` two first one, depending of the page number. `\ledsidenote{<text>}` is the command
`\ledouternote` for a moveable sidenote.

```

\ledsidenote
5783 \newcommand*\ledleftnote[1]{\edtext{}\l@dlsnote{#1}}
5784 \newcommand*\ledrightnote[1]{\edtext{}\l@drsnote{#1}}
5785
5786 \newcommand*\ledinnernote[1]{%
5787   \ifodd\c@page% Do not use \page@num, because it is not yet calculated
when command is called
5788   \ledleftnote{#1}%
5789   \else%
5790   \ledrightnote{#1}%
5791   \fi%
5792 }
5793
5794 \newcommand*\ledouternote[1]{%
5795   \ifodd\c@page% Do not use \page@num, because it is not yet calculated
when command is called
5796   \ledrightnote{#1}%
5797   \else%
5798   \ledleftnote{#1}%
5799   \fi%
5800 }
5801
5802 \newcommand*\ledsidenote[1]{\edtext{}\l@dcsnote{#1}}
5803 %

```

`\l@dlsnote` . The ‘footnotes’ for left, right, and moveable sidenotes. The whole scheme is reminiscent
`\l@drsnote` of the critical footnotes code.

`\l@dcsnote`

```

5804 \newif\ifrightnoteup
5805 \rightnoteuptrue
5806
5807 \newcommand*{\l@dlsnote}[1]{%
5808 \begingroup%
5809 \newcommand{\content}{#1}%
5810 \ifnumberedpar@
5811 \ifledRcol%
5812 \xright@appenditem{\noexpand\vl@dlsnote{\expandonce\content}}{%
5813 \to\inserts@listR
5814 \global\advance\insert@countR \@ne%
5815 \else%
5816 \xright@appenditem{\noexpand\vl@dlsnote{\expandonce\content}}{%
5817 \to\inserts@list
5818 \global\advance\insert@count \@ne%
5819 \fi
5820 \fi\ignorespaces\endgroup}
5821
5822 \newcommand*{\l@drsnote}[1]{%
5823 \begingroup%
5824 \newcommand{\content}{#1}%
5825 \ifnumberedpar@
5826 \ifledRcol%
5827 \xright@appenditem{\noexpand\vl@drsnote{\expandonce\content}}{%
5828 \to\inserts@listR
5829 \global\advance\insert@countR \@ne%
5830 \else%
5831 \xright@appenditem{\noexpand\vl@drsnote{\expandonce\content}}{%
5832 \to\inserts@list
5833 \global\advance\insert@count \@ne%
5834 \fi
5835 \fi\ignorespaces\endgroup}
5836
5837 \newcommand*{\l@dcsnote}[1]{%
5838 \begingroup%
5839 \newcommand{\content}{#1}%
5840 \ifnumberedpar@
5841 \ifledRcol%
5842 \xright@appenditem{\noexpand\vl@dcsnote{\expandonce\content}}{%
5843 \to\inserts@listR
5844 \global\advance\insert@countR \@ne%
5845 \else%
5846 \xright@appenditem{\noexpand\vl@dcsnote{\expandonce\content}}{%
5847 \to\inserts@list
5848 \global\advance\insert@count \@ne%
5849 \fi
5850 \fi\ignorespaces\endgroup}
5851
5852 %

```

`\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 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.

```

5853 \newcommand*{\vl@dlsnote}[1]{%
5854   \ifledRcol{%
5855     \@l@tempcntb=\sidenote@marginR%
5856     \ifnum\@l@tempcntb>\@ne%
5857       \advance\@l@tempcntb by\page@numR%
5858     \fi%
5859   \else%
5860     \@l@tempcntb=\sidenote@margin%
5861     \ifnum\@l@tempcntb>\@ne%
5862       \advance\@l@tempcntb by\page@num%
5863     \fi%
5864   \fi%
5865   \ifodd\@l@tempcntb%
5866     \listgadd{\l@dcsnotetext@l}{#1}%
5867   \else%
5868     \listgadd{\l@dcsnotetext}{#1}%
5869   \fi
5870 }
5871 \newcommand*{\vl@drsnote}[1]{%
5872   \ifledRcol{%
5873     \@l@tempcntb=\sidenote@marginR%
5874     \ifnum\@l@tempcntb>\@ne%
5875       \advance\@l@tempcntb by\page@numR%
5876     \fi%
5877   \else%
5878     \@l@tempcntb=\sidenote@margin%
5879     \ifnum\@l@tempcntb>\@ne%
5880       \advance\@l@tempcntb by\page@num%
5881     \fi%
5882   \fi%
5883   \ifodd\@l@tempcntb%

```

```

5884 \listgadd{\l@dcstetext}{#1}%
5885 \else%
5886 \listgadd{\l@dcstetext@r}{#1}%
5887 \fi%
5888 }
5889 \newcommand*{\vl@dcstetext}[1]{\listgadd{\l@dcstetext}{#1}}
5890
5891 %

```

`\setl@dlp@rbox` `\setl@dlprbox{<lednums>}{<tag>}{<text>}` puts `<text>` into the `\l@dlp@rbox` box. And similarly for the right side box. It is these boxes that finally get displayed in the margins.

```

5892 \newcommand*{\setl@dlp@rbox}[1]{%
5893 \begingroup%
5894 \parindent\z@\hspace=\ledlsnotewidth%
5895 \ledlsnotefontsetup%We kept it outside of the vbox, because can affect
the ragging
5896 \global\setbox\l@dlp@rbox%
5897 \ifleftnoteup%
5898 =\vbox to\z@{\ledlsnotefontsetup\vss #1}}%We put \
ledlsnotefontsetup inside footnote because required for color command. Note
the {} to keep setting local.
5899 \else%
5900 =\vbox to 0.70\baselineskip{\ledlsnotefontsetup\strut#1\vss}}%
5901 \fi%
5902 \endgroup%
5903 }
5904
5905 \newcommand*{\setl@drp@rbox}[1]{%
5906 \begingroup%
5907 \parindent\z@\hspace=\ledrsnotewidth%
5908 \ledrsnotefontsetup%We kept it outside of the vbox, because can affect
the ragging
5909 \global\setbox\l@drp@rbox%
5910 \ifrightnoteup%
5911 =\vbox to\z@{\ledrsnotefontsetup\vss#1}}%We put \ledrsnotefontsetup
inside footnote because required for color command. Note the {} to keep
setting local.
5912 \else%
5913 =\vbox to0.7\baselineskip{\ledrsnotefontsetup\strut#1\vss}}%
5914 \fi%
5915 \endgroup%
5916 }%
5917 \newif\ifleftnoteup
5918 \leftnoteuptrue
5919 %

```

`\@sidenotesep` This macro is used to separate sidenotes of the same line.


```

5920 \newcommand{\setsidenotesep}[1]{\gdef\@sidenotesep{#1}}
5921 \newcommand{\@sidenotesep}{, }
5922 %

```

\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.

```

5923 \newcommand*{\affixside@note}{%
5924   \def\sidenotecontent@{}%
5925   \numgdef{\itemcount@}{0}%
5926   \def\do##1{%
5927     \ifnumequal{\itemcount@}{0}%
5928       {%
5929         \appto\sidenotecontent@{##1}}% Not print not separator before
the 1st note
5930       {\appto\sidenotecontent@{\@sidenotesep ##1}%
5931       }%
5932       \numgdef{\itemcount@}{\itemcount@+\@ne}%
5933   }%
5934   \dolistloop{\l@dcnotesetext}%
5935   \ifnumgreater{\itemcount@}{1}{\led@err@ManySidenotes}{}%
5936   %

```

And we do the same for left and right notes (not movable).

```

5937 \gdef\@templ@d{%
5938 \gdef\@templ@n{\l@dcnotesetext\l@dcnotesetext@l\l@dcnotesetext@r}%
5939 \ifx\@templ@d\@templ@n \else%
5940   \if@twocolumn%
5941     \if@firstcolumn%
5942       \setl@dlp@rbox{##1}{\sidenotecontent@}%
5943     \else%
5944       \setl@drp@rbox{\sidenotecontent@}%
5945     \fi%
5946   \else%
5947     \l@dttempcntb=\sidenote@margin%
5948     \ifnum\l@dttempcntb>\@ne%
5949       \advance\l@dttempcntb by\page@num%
5950     \fi%
5951     \ifodd\l@dttempcntb%
5952       \setl@drp@rbox{\sidenotecontent@}%
5953       \gdef\sidenotecontent@{}%
5954       \numgdef{\itemcount@}{0}%
5955       \dolistloop{\l@dcnotesetext@l}%
5956       \ifnumgreater{\itemcount@}{1}{\led@err@ManyLeftnotes}{}%
5957       \setl@dlp@rbox{\sidenotecontent@}%
5958     \else%

```

```

5959     \setl@dlp@rbox{\sidenotecontent@}%
5960     \gdef\sidenotecontent@{}%
5961     \numgdef{\itemcount@}{0}%
5962     \dolistloop{\l@dcstotetext@r}%
5963     \ifnumgreater{\itemcount@}{1}{\led@err@ManyRightnotes}{}%
5964     \setl@drp@rbox{\sidenotecontent@}%
5965     \fi%
5966   \fi%
5967 \fi%
5968 }
5969 %

```

XXV Minipages and such

We can put footnotes into minipages. The preparatory code has been set up earlier, all that remains is to ensure that it is available inside a minipage box. This requires some alteration to the kernel code, specifically the `\@iiminipage` and `\endminipage` macros. We will arrange this so that additional series can be easily added.

`\l@dfetbeginmini` These will be the hooks in `\@iiminipage` and `\endminipage`.
`\l@dfetendmini` They can be extended to handle other things if necessary.

```

5970 \ifnoledgroup@else%
5971 \newcommand*{\l@dfetbeginmini}{\@ledgrouptrue\l@dedbeginmini\l@dfambeginmini}
5972 \newcommand*{\l@dfetendmini}{%
5973   \IfStrEq{critical-familiar}{\@mpfnpos}%
5974   {\l@dedendmini\l@dfamendmini}%
5975   {%
5976     \IfStrEq{familiar-critical}{\@mpfnpos}%
5977     {\l@dfamendmini\l@dedendmini}%
5978     {\l@dedendmini\l@dfamendmini}%
5979   }%
5980 }%
5981 %

```

`\l@dedbeginmini` These handle the initiation and closure of critical footnotes in a minipage environment.

```

\l@dedendmini
5982 \newcommand*{\l@dedbeginmini}{%
5983   \unless\ifnocritical@%
5984   \def\do##1{\csletcs{v##1footnote}{mpv##1footnote}}%
5985   \dolistloop{\@series}%
5986   \fi%
5987 }
5988 \newcommand*{\l@dedendmini}{%
5989   \unless\ifnocritical@%
5990   \ifl@dpairing%
5991     \ifledRcol%
5992       \flush@notesR%

```

```

5993     \else%
5994         \flush@notes%
5995     \fi%
5996 \fi
5997 \def\do##1{%
5998     \ifvoid\csuse{mp##1footins}\else%
5999         \ifl@dpairing\ifparledgroup%
6000             \ifledRcol%
6001                 \dingdef{\parledgroup@beforenotesR}{\parledgroup@beforenotesR+\
skip\@nameuse{mp##1footins}}}%
6002             \else%
6003                 \dingdef{\parledgroup@beforenotesL}{\parledgroup@beforenotesL
+\skip\@nameuse{mp##1footins}}}%
6004             \fi%
6005             \fi\fi%
6006             \csuse{mp##1footgroup}{##1}%
6007         \fi}%
6008     \dolistloop{\@series}%
6009 \fi%
6010 }%
6011 %
6012 %

```

`\l@dfambeginmini` These handle the initiation and closure of familiar footnotes in a minipage environment.

```

\l@dfamendmini
6013 \newcommand*\l@dfambeginmini{%
6014     \unless\ifnofamiliar%
6015     \def\do##1{\csletcs{vfootnote##1}{mpvfootnote##1}}%
6016     \dolistloop{\@series}%
6017     \fi%
6018 }%
6019
6020 \newcommand*\l@dfamendmini{%
6021     \unless\ifnofamiliar%
6022     \def\do##1{%
6023         \ifvoid\csuse{mpfootins##1}\else%
6024             \csuse{mpfootgroup##1}{##1}%
6025         \fi}%
6026     \dolistloop{\@series}%
6027     \fi%
6028 }%
6029 %

```

`\@iiiminipage` This is our extended form of the kernel `\@iiiminipage` defined in `ltboxes.dtx`.

```

6030 \patchcmd%
6031 {\@iiiminipage}%
6032 {\let\@footnotetext\@mpfootnotetext}%
6033 {\let\@footnotetext\@mpfootnotetext\l@dfetbeginmini}%
6034 {}%

```

```

6035 {\led@error@fail@patch@iiiiminipage}%
6036 %

```

`\endminipage` This is our extended form of the kernel `\endminipage` defined in `ltboxes.dtx`.

```

6037 \patchcmd%
6038 {\endminipage}%
6039 {\footnoterule}%
6040 {\footnoterule\l@advance@parledgroup@beforenormalnotes}%
6041 {}%
6042 {\led@error@fail@patch@endminipage}
6043
6044 \patchcmd%
6045 {\endminipage}%
6046 {\@minipagefalse}%
6047 {\l@dfeetendmini\@minipagefalse}%
6048 {}%
6049 {\led@error@fail@patch@endminipage}
6050
6051 %

```

`\l@dunboxmpfoot` `\@ldunboxmpfoot` insert normal footnotes for ledgroup.

`\l@advance@parledgroup@beforenormalnotes`

```

6052 \newcommand*{\l@dunboxmpfoot}{%
6053   \vskip\skip\@mpfootins
6054   \normalcolor
6055   \footnoterule
6056   \l@advance@parledgroup@beforenormalnotes
6057   \unvbox\@mpfootins%
6058 }
6059 %

```

When using parallel ledgroup, we need to store the vertical space added before footnote, in order to compensate them between left and right pages.

```

6060 \newcommand{\l@advance@parledgroup@beforenormalnotes}{%
6061   \ifparledgroup
6062     \ifl@pairing
6063       \ifledRcol
6064         \dimgdef{\parledgroup@beforenotesR}{\parledgroup@beforenotesR+\
skip\@mpfootins}
6065       \else
6066         \dimgdef{\parledgroup@beforenotesL}{\parledgroup@beforenotesL+\
skip\@mpfootins}
6067       \fi
6068     \fi
6069   \fi
6070 }
6071 %

```

`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.

```

6072 \newenvironment{ledgroup}{%
6073   \resetprevpage@num%
6074   \def\@mpfn{mpfootnote}\def\thempfn{\thempfootnote}\c@mpfootnote\z@%
6075   \let\@footnotetext\@mpfootnotetext
6076   \l@dfetbeginmini%
6077 }{%
6078   \par
6079   \unskip
6080   \ifvoid\@mpfootins\else
6081     \l@dunboxmpfoot
6082   \fi
6083   \l@dfetendmini%
6084   \@ledgroupfalse%
6085 }
6086
6087
6088
6089 %

```

`ledgroupsize` `\begin{ledgroupsize}[\langle pos \rangle]{\langle width \rangle}`

This environment puts footnotes at the end, even if that happens to be in the middle of a page, or crossing a page boundary. It is a sort of unboxed, variable `\langle width \rangle` minipage. The optional `\langle pos \rangle` controls the sideways position of numbered text.

```

6090 \newenvironment{ledgroupsize}[2][l]{%
6091 %

```

Set the various text measures.

```

6092 \hspace #2\relax
6093 %

```

Initialize fills for centering.

```

6094 \let\ledllfill\hfil
6095 \let\ledrlfill\hfil
6096 \def\@tempa{#1}\def\@tempb{1}%
6097 %

```

Left adjusted numbered lines

```

6098   \ifx\@tempa\@tempb
6099     \let\ledllfill\relax
6100   \else
6101     \def\@tempb{r}%
6102     \ifx\@tempa\@tempb
6103 %

```

Right adjusted numbered lines

```

6104     \let\ledrlfill\relax
6105     \fi
6106     \fi
6107 %

Set up the footnoting.

6108     \def\@mpfn{mpfootnote}\def\thempfn{\thempfootnote}\c@mpfootnote\z@
6109     \let\@footnotetext\@mpfootnotetext
6110     \l@dfetbeginmini%
6111 }{
6112     \par
6113     \unskip
6114     \ifvoid\@mpfootins\else
6115         \l@duboxmpfoot
6116     \fi
6117     \l@dfetendmini%
6118 }
6119
6120 %

Close the \ifnoledgroup\else.

6121 \fi%
6122 %

```

`\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`.

```

6123 \newif\ifledgroupnotesL@
6124 \newif\ifledgroupnotesR@
6125 %

```

XXVI Indexing

Here is some code for indexing using page and line numbers.

XXVI.1 Looking on package order

First, ensure that `imakeidx` or `indextools` is loaded *before* `eledmac`.

```

6126 \AtBeginDocument{%
6127     \unless\ifl@imakeidx%
6128         \@ifpackageloaded{imakeidx}{\led@error@ImakeidxAfterEledmac}{}%
6129     \fi%
6130     \unless\ifl@indextools%
6131         \@ifpackageloaded{indextools}{\led@error@indextoolsAfterEledmac}{}%
6132     \fi%
6133 }
6134 %

```

XXVI.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
6135 \newcommand{\pagelinesep}{-}
6136 \newcommand{\edindexlab}{${&}}
6137 \newcounter{labidx}
6138 \setcounter{labidx}{0}
6139
6140 %
```

`\doedindexlabel` This macro sets an `\edlabel`.

```
6141 \newcommand{\doedindexlabel}{%
6142   \stepcounter{labidx}%
6143   \edlabel{\edindexlab\thelabidx}%
6144 }
6145
6146 %
```

`\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`.

```
6147 \newcounter{pageline}%
6148 \renewcommand{\thepageline}{%
6149   \thepage%
6150   \pagelinesep%
6151   \xlineref{\edindexlab\thelabidx}%
6152 }
6153 %
```

`\thestartpageline` These macros make up the page/line start/end number when the `\edindex` command
`\theendpageline` is called in critical notes.

```
6154 \newcommand{\thestartpageline}{%
6155   \l@dparsedstartpage%
6156   \pagelinesep%
6157   \l@dparsedstartline%
6158 }
6159 \newcommand{\theendpageline}{%
6160   \l@dparsedendpage%
6161   \pagelinesep%
6162   \l@dparsedendline%
6163 }
6164 %
```

XXVI.3 Code specific to `\edindex` in critical footnotes

`\if@edindex@fornote@true` This boolean test is switching at the beginning of each critical note, to allow index referring to this note.

```
6165 \newif\if@edindex@fornote@
6166 %
```

`\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 |.

```
6167 \newcommand{\prepare@edindex@fornote}[1]{%
6168   \l@dp@rsefootspec#1|}%
6169   \@edindex@fornote@true%
6170 }
6171 %
```

`\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.

```
6172 \newcommand{\get@edindex@ledinnote@command}{%
6173   \ifxindy%
6174     \gdef\@ledinnote@command{%
6175       ledinnote\thelabidx%
6176     }%
6177   \ifxindyhyperref%
6178     \immediate\write\eledmac@xindy@out{%
6179       (define-attributes ("ledinnote\thelabidx"))^^J
6180       \space\space(markup-locref^^J
6181       \eledmacmarkuplocdepth^^J
6182       :open "\string\ledinnote[\edindexlab\thelabidx]{\@index@command
6183       }{"^^J
6184       :close "}"^^J
6185       :attr "ledinnote\thelabidx"^^J
6186     )
6187   }%
6188   \else%
6189     \immediate\write\eledmac@xindy@out{%
6190       (define-attributes ("ledinnote\thelabidx"))^^J
6191       \space\space(markup-locref^^J
6192       \eledmacmarkuplocdepth^^J
6193       :open "\string\ledinnote{\@index@command}{"^^J
6194       :close "}"^^J
6195       :attr "ledinnote\thelabidx"^^J
6196     )
6197   }%
```



```

6197 \fi%
6198 %

```

If we do not use xindy option, `\@ledinnote@command` will produce something like `ledinnote{formattingcommand}`.

```

6199 \else%
6200 \gdef\@ledinnote@command{%
6201 ledinnote[\edindexlab\thelabidx]{\@index@command}%
6202 }%
6203 \fi%
6204 }
6205 %

```

XXVI.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 `|`.

```

6206 \def\get@index@command#1|#2+{%
6207 \gdef\@index@txt{#1}%
6208 \gdef\@index@command{#2}%
6209 \xdef\@index@parenthesis{}%
6210 \IfBeginWith{\@index@command}{(}{%
6211 \StrGobbleLeft{\@index@command}{1}{\@index@command}%
6212 \global\let\@index@command\@index@command%
6213 \xdef\@index@parenthesis{(%
6214 }%
6215 \IfBeginWith{\@index@command}{)}{%
6216 \StrGobbleLeft{\@index@command}{1}{\@index@command}%
6217 \global\let\@index@command\@index@command%
6218 \xdef\@index@parenthesis{)}%
6219 }%
6220 }
6221 %

```

XXVI.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.

```

6222 \newcommandx{\ledinnote}[3][1,usedefault]{%
6223 \ifboolexpr{%
6224 test{\ifdefequal{\iftrue}{\ifHy@hyperindex}}%
6225 or%
6226 bool {xindyhyperref@}%
6227 }%
6228 {%
6229 \csuse{#2}{\hyperlink{#1}{\ledinnotemark{#3}}}%
6230 }%

```

```

6231   {%
6232   \csuse{#2}{\ledinnotemark{#3}}%
6233   }%
6234 }%
6235 \newcommand{\ledinnotehyperpage}[2]{\csuse{#1}{\ledinnotemark{\hyperpage
6236 {#2}}}}%
6237 \newcommand{\ledinnotemark}[1]{#1\emph{n}}%
6238 %

```

XXVI.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
6238 \newcommandx{\@wredindex}[2][1=\expandonce\jobname,usedefault]{%#1 = the
index name, #2 = the text
6239 \ifl@imakeidx%
6240 \if@edindex@fornote%
6241 \IfSubStr[1]{#2}{|}{\get@index@command#2+}{\get@index@command#2|+}%
6242 \get@edindex@ledinnote@command%
6243 \expandafter\imki@wrindexentry{#1}{\@index@txt|(\@ledinnote@command
}{\thestartpageline}%
6244 \expandafter\imki@wrindexentry{#1}{\@index@txt|)\@ledinnote@command
}{\theendpageline}%
6245 \else%
6246 \get@edindex@hyperref{#2}%
6247 \imki@wrindexentry{#1}{\@index@txt\@edindex@hyperref}{\thepageline}%
6248 \fi%
6249 \else%
6250 \if@edindex@fornote%
6251 \IfSubStr[1]{#2}{|}{\get@index@command#2+}{\get@index@command#2|+}%
6252 \get@edindex@ledinnote@command%
6253 \expandafter\protected@write\@indexfile{}%
6254 {\string\indexentry{\@index@txt|(\@ledinnote@command){\thestartpageline}
}%
6255 }%
6256 \expandafter\protected@write\@indexfile{}%
6257 {\string\indexentry{\@index@txt|)\@ledinnote@command}{\theendpageline}
}%
6258 }%
6259 \else%
6260 \protected@write\@indexfile{}%
6261 {\string\indexentry{#2}{\thepageline}

```

```

6262 }%
6263 \fi%
6264 \fi%
6265 \endgroup
6266 \@esphack%
6267 }
6268 %

```

Need to add the definition of `\edindex` to `\makeindex`, and initialise `\edindex` to do nothing.

```

6269 \pretocmd{\makeindex}{%
6270 \def\edindex{\@bsphack
6271 \doedindexlabel
6272 \begingroup
6273 \@sanitize
6274 \@wredindex}}{}{}
6275 \newcommand{\edindex}[1]{\@bsphack\@esphack}
6276 %

```

XXVI.7 **Hyperlink**

`\hyperlinkformat` `\hyperlinkformat` command is to be used to have both a internal hyperlink and a format, when indexing.

```

6277 \newcommand{\hyperlinkformat}[3]{%
6278 \ifstrempy{#1}%
6279 {\hyperlink{#2}{#3}}%
6280 {\csuse{#1}{\hyperlink{#2}{#3}}}%
6281 }}
6282 %

```

`\hyperlinkR` `\hyperlinkR` command is to be used to create a internal hyperlink and `\ledRflag`, when indexing.

```

6283 \newcommand{\hyperlinkR}[2]{%
6284 \hyperlink{#1}{#2\@Rlineflag}%
6285 }%
6286 %
6287 %

```

`\hyperlinkformatR` `\hyperlinkformatR` command is to be used to create a internal hyperlink, a format and a `\@Rlineflag`, when indexing.

```

6288 \newcommand{\hyperlinkformatR}[3]{%
6289 \hyperlinkformat{#1}{#2}{#3\@Rlineflag}%
6290 }%
6291 %
6292 %

```

`\get@edindex@hyperref` `\@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`).

```
6293 \newcommand{\get@edindex@hyperref}[1]{%
6294 %
```

We have to disable temporary spaces to work through a xstring bug (or feature?)

```
6295 \edef\temp@{%
6296 \catcode`\ =9 %space need for catcode
6297 \detokenize{#1}%For active character in unicode
6298 \catcode`\ =10 % space need for catcode
6299 }%
6300 %
```

Now, we define `\@edindex@hyperref` if the hyperindex of `hyperref` is enabled.

```
6301 \ifdefequal{\iftrue}{\ifHy@hyperindex}{%
6302 \IfSubStr{\temp@}{|}%
6303 {\get@index@command#1+%
6304 \ifledRcol%
6305 \gdef\@edindex@hyperref{|\@index@parenthesis %space kept
6306 hyperlinkformatR{\@index@command}%
6307 {\edindexlab\thelabidx}}%
6308 \else%
6309 \gdef\@edindex@hyperref{|\@index@parenthesis %space kept
6310 hyperlinkformat{\@index@command}%
6311 {\edindexlab\thelabidx}}%
6312 \fi%
6313 }%
6314 {\get@index@command#1+%
6315 \ifledRcol%
6316 \gdef\@edindex@hyperref{hyperlinkR{\edindexlab\thelabidx}}%
6317 \else%
6318 \gdef\@edindex@hyperref{hyperlink{\edindexlab\thelabidx}}%
6319 \fi%
6320 }%
6321 }%
6322 %
```

```
6323 % If we use both xindy and hyperref, first get the \protect\cs{
index@command} command.
6324 % Then define \protect\cs{@edindex@hyperref} in the form \verb+eledmacXXX+
6325 % \begin{macrocode}
6326 {\ifxindyhyperref%
6327 \IfSubStr{\temp@}{|}%
6328 {\get@index@command#1+%
6329 {\get@index@command#1+%
6330 \gdef\@edindex@hyperref{eledmac\thelabidx}%
6331 %
```

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`.

```

6332 \IfStrEq{\@index@parenthesis}{(}{%
6333     {%
6334     \csxdef{xindy@parenthesis@\@index@txt}{\thelabidx}%
6335     \gdef\@edindex@hyperref{|\@edmac\thelabidx}%
6336     }%
6337     {}%
6338 %

```

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`.

```

6339 \IfStrEq{\@index@parenthesis}{)}{%
6340     {%
6341     \xdef\@edindex@hyperref{|\@edmac\csuse{xindy@parenthesis@\@index@txt}}%
6342     \global\csundef{xindy@parenthesis@\@index@txt}%
6343     }%
6344 %

```

Write in the `.xdy` file the attributes of the location.

```

6345     {%
6346     \immediate\write\@edmac\xindy@out{%
6347     (define-attributes ("eledmac\thelabidx"))^^J
6348     \space\space(markup-locref^^J
6349     \eledmacmarkuplocdepth^^J
6350     :open "\string\hyperlink%
6351     \ifledRcol R\fi%
6352     {\edindexlab\thelabidx}%
6353     {\ifdefempty{\@index@command}%
6354     }%
6355     {\@backslashchar\@index@command}%
6356     {"^^J
6357     :close "}"^^J
6358     :attr "eledmac\thelabidx"^^J
6359     )
6360     }%
6361     }%
6362 %

```

And now, in any other case.

```

6363 \else%
6364 \gdef\@index@txt{#1}%
6365 \gdef\@edindex@hyperref{}%
6366 \fi%
6367 }%
6368 }
6369 %

```

XXVI.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.

```

6370 \newcommand{\led@set@index@fornote}[1]{%
6371   \ifbool{indtl@innote}%
6372     {\let\index\nindex}%
6373     {}%
6374   \ifbool{indtl@notenumber}%
6375     {%
6376       \renewcommand{\index}[2][\indtl@jobname]{%
6377         \orig@@index[##1]{%
6378           ##2|innotenumber{\csuse{thefootnote#1}}}%
6379         }%
6380       }%
6381     }%
6382   {}%
6383 }%
6384 %

```

`\led@reinit@index@fornote` The `\led@reinit@index@fornote` just reset the default value of `\index`.

```

6385 \newcommand{\led@reinit@index@fornote}{%
6386   \ifbool{indtl@innote}%
6387     {\let\index\orig@@index}%
6388     {}%
6389   \ifbool{indtl@notenumber}%
6390     {\let\index\orig@@index}%
6391     {}%
6392 }%
6393 %

```

XXVII Glossaries

Here, we define the `\gls`-like commands prefixed by `ed`, only if the package `glossaries` is loaded.

```

6394 \AtBeginDocument{%
6395   \ifpackageloaded{glossaries}{%
6396     %

```

First those which arguments are `[<options>]{<label>}[<insert>]`.

```

6397   \renewcommand{\do}[1]{%
6398     \expandafter\DeclareRobustCommand\csname ed#1\endcsname[3][1,3,
usedefault]{%
6399       \doedindexlabel%
6400       \csname#1\endcsname[counter=pageline,##1]{##2}[##3]%

```

```

6401 }%
6402 \expandafter\WithSuffix\expandafter\DeclareRobustCommandx\csname ed
#1\endcsname*[3][1,3,usedefault]{%
6403 \doedindexlabel%
6404 \csname#1\endcsname*[counter=pageline,##1]{##2}[##3]%
6405 }%
6406 }%
6407 \docsvlist{%
6408 gls,%
6409 GLs,%
6410 GLS,%
6411 glspl,%
6412 Glspl,%
6413 GLSpl,%
6414 glstext,%
6415 Glstext,%
6416 GLStext,%
6417 Glsfirst,%
6418 GLSfirst,%
6419 glsplural,%
6420 Glsplural,%
6421 GLSplural,%
6422 glsfirstplural,%
6423 Glsfirstplural,%
6424 GLSfirstplural,%
6425 glsname,%
6426 Glname,%
6427 GLSname,%
6428 glssymbol,%
6429 Glssymbol,%
6430 GLSsymbol,%
6431 glsdesc,%
6432 Glsdesc,%
6433 GLSdesc,%
6434 glsuseri,%
6435 Glsuseri,%
6436 GLSuseri,%
6437 glsuserii,%
6438 Glsuserii,%
6439 GLSuserii,%
6440 glsuseriii,%
6441 Glsuseriii,%
6442 GLSuseriii,%
6443 glsuseriv,%
6444 Glsuseriv,%
6445 GLSuseriv,%
6446 glsuserv,%
6447 Glsuserv,%
6448 GLSuserv,%
6449 glsuservi,%

```

```

6450     Glsuservi,%
6451     GLSuservi%
6452     }%
6453 %

```

First those which arguments are [*options*]{*label*}{*link text*}.

```

6454     \renewcommand{\do}[1]{%
6455     \expandafter\DeclareRobustCommand\csname ed#1\endcsname[3][1,
usedefault]{%
6456     \doedindexlabel%
6457     \csname#1\endcsname[counter=pageline,##1]{##2}{##3}%
6458     }%
6459     \expandafter\WithSuffix\expandafter\DeclareRobustCommand\csname ed
#1\endcsname*[3][1,usedefault]{%
6460     \doedindexlabel%
6461     \csname#1\endcsname*[counter=pageline,##1]{##2}{##3}%
6462     }%
6463     }%
6464     \docsvlist{glsdisp,glslink}%
6465     }{}%
6466 }%
6467 %

```

XXVIII 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`.

XXVIII.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 typographie 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.

```

6468 \def\@hangingsymbol{}
6469 \newcommand*\sethangingsymbol[1]{%
6470     \gdef\@hangingsymbol{#1}%
6471 }%
6472 \newif\ifinstanza
6473 %

```

`\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.

```

6474 \newif\ifinserthangingsymbol
6475 \newcommand{\inserthangingsymbol}{%
6476 \ifinserthangingsymbol%
6477   \ifinstanza%
6478     \@hangingsymbol%
6479   \fi%
6480 \fi%
6481 }
6482 %

```

XXVIII.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.

```

6483 \newcommand*{\ampersand}{\char`\&}
6484
6485 %

```

XXVIII.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.

```

6486 \chardef\body=\catcode`\@
6487 \catcode`\@=11
6488 \chardef\next=\catcode`\&
6489 \catcode`\&=\active
6490
6491 %

```

XXVIII.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.

```

6492 \newcount\stanza@count
6493 \newlength{\stanzaindentbase}
6494 \setlength{\stanzaindentbase}{20pt}
6495
6496 %

```

`\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.

```

6497 \def\strip@szacnt#1,#2|{\def\@tempb{#1}\def\@tempa{#2|}}
6498 \newcommand*\setstanzavalues[2]{\def\@tempa{#2,,|}%
6499   \stanza@count\z@
6500   \def\next{\expandafter\strip@szacnt\@tempa
6501     \ifx\@tempb\empty\let\next\relax\else
6502     \expandafter\mathchardef\csname #1\@number\stanza@count
6503     \@endcsname\@tempb\relax
6504     \advance\stanza@count\@ne\fi\next}%
6505   \next}
6506
6507 %

```

`\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).

```

6508 \newcommand*\setstanzaindents[1]{\setstanzavalues{sza}{#1}}
6509 \newcommand*\setstanzapenalties[1]{\setstanzavalues{szp}{#1}}
6510 %
6511 %

```

`\managestanza@modulo` Since version 0.13, the `stanzaindentsrepetition` counter can be used when the indentation 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.

```

6512 \newcounter{stanzaindentsrepetition}
6513 \newcount\stanza@modulo
6514
6515 \newcommand*\managestanza@modulo[0]{%
6516   \advance\stanza@modulo\@ne%
6517   \ifnum\stanza@modulo>\value{stanzaindentsrepetition}%
6518   \stanza@modulo\@ne%
6519   \fi%
6520 }
6521 %

```

`\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.

```

6522 \newcommand{\stanzaindent}[1]{%
6523   \hspace{\dimexpr#1\stanzaindentbase-\parindent\relax}%
6524   \ignorespaces%
6525 }%
6526 \WithSuffix\newcommand\stanzaindent*[1]{%
6527   \stanzaindent{#1}%
6528   \global\advance\stanza@modulo-\@ne%
6529   \ifnum\stanza@modulo=0%
6530     \global\stanza@modulo=\value{stanzaindentrepetition}%
6531   \fi%
6532   \ignorespaces%
6533 }%
6534 %

```

XXVIII.5 Numbering stanza

Here, macro for numbering stanza. First, the stanza counter.

```

\thestanza \newcounter{stanza}
6536 \renewcommand{\thestanza}{%
6537   \textbf{\arabic{stanza}}%
6538 }
6539 %

```

`\ifnumberstanza` Then, macro to activate automatically numbering of stanza.

```

6540 \newif\ifnumberstanza%
6541 %

```

`\@insertstanzanumber` Now, macro called at the first line of of verse of a stanza.

```

6542 \newcommand{\@insertstanzanumber}[0]{%
6543   \ifnumberstanza%
6544     \ifl@dpairing%
6545       \ifledRcol%
6546         \stanzanumwrapper{\thestanzaR}%
6547       \else%
6548         \stanzanumwrapper{\thestanzaL}%
6549       \fi%
6550     \else%
6551       \stanzanumwrapper{\thestanza}%
6552     \fi%
6553     \setline{1}%
6554   \fi%
6555 }%
6556 %

```

`\@advancestanzanumber` Also a command to advance the counter of stanza.

```

6557 \newcommand{\@advancestanzanumber}[0]{%
6558   \ifnumberstanza%
6559   \ifl@dpairing%
6560   \ifledRcol%
6561     \addtocounter{stanzaR}{1}%
6562   \else%
6563     \addtocounter{stanzaL}{1}%
6564   \fi%
6565 \else%
6566   \addtocounter{stanza}{1}%
6567 \fi%
6568 \fi%
6569 }%
6570 %

```

`\stanzanumwrapper` And finally, the wrapper for stanza number

```

6571 \newcommand{\stanzanumwrapper}[1]{%
6572   \flagstanza{#1}%
6573 }%
6574 %

```

XXVIII.6 Stanza number in note

Here, the command called when printing stanza number in notes.

```

6575 \newcommand{\printstanza}[0]{%
6576   \ifbool{expr{bool{l@dpairing} or bool{l@dprintingpages} or bool{
l@dprintingcolumns}}}{%
6577     \ifledRcol{%
6578       \thestanzaR%
6579     \else%
6580       \thestanzaL%
6581     \fi%
6582   }{%
6583     \thestanza%
6584   }%
6585 }
6586 %

```

XXVIII.7 Main work

`\stanza@line` Now we arrive at the main works. `\stanza@line` sets the indentation for the line and starts a numbered paragraph—each line is treated as a paragraph. `\stanza@hang` sets the hanging indentation to be used if the stanza line requires more than one print line.

If it is known that each stanza line will fit on one print line, it is advisable to set the hanging indentation to zero. `\sza@penalty` places the specified penalty following each stanza line. By default, this facility is turned off so that no penalty is included. However,

the user may initiate these penalties to indicate good and bad places in the stanza for page breaking.

```

6587 \newcommandx{\stanza@line}[1][1]{
6588   \ifnum\value{stanzaindentrepetition}=0
6589     \parindent=\csname sza@\number\stanza@count
6590       @\endcsname\stanzaindentbase
6591   \else
6592     \parindent=\csname sza@\number\stanza@modulo
6593       @\endcsname\stanzaindentbase
6594     \managestanza@modulo
6595   \fi
6596   \pstart[#1]\stanza@hang\ignorespaces}
6597 \xdef\stanza@hang{\noexpand\leavevmode\noexpand\startlock
6598   \hangindent\expandafter
6599   \noexpand\csname sza@0@\endcsname\stanzaindentbase
6600   \hangafter\@ne}
6601 \def\sza@penalty{\count@\csname szp@\number\stanza@count @\endcsname
6602   \ifnum\count@>\@M\advance\count@-\@M\penalty-\else
6603   \penalty\fi\count@}
6604 %

```

\@startstanza Now we have the components of the \stanza macro, which appears at the start of a
 \stanza group of lines. This macro initializes the count and checks to see if hanging indentation
 \@stopstanza and penalties are to be included. Hanging indentation suspends the line count, so that
 \@AtEveryStopStanza the enumeration is by verse line rather than by print line. If the print line count is
 \AtEveryStanza desired, invoke \let\startlock\relax and do the same for \endlock. Here and
 \newverse 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 \&.

```

6605 \xdef\@startstanza[#1]{%
6606   \noexpand\instanzatrue\expandafter
6607   \begingroup%
6608   \catcode`\noexpand\&\active%
6609   \global\stanza@count\@ne\stanza@modulo\@ne
6610   \noexpand\ifnum\expandafter\noexpand
6611   \csname sza@0@\endcsname=\z@\let\noexpand\stanza@hang\relax
6612   \let\noexpand\endlock\relax\noexpand\else\interlinepenalty
6613   \@M\rightskip\z@ plus 1fil\relax\noexpand\fi\noexpand\ifnum
6614   \expandafter\noexpand\csname szp@0@\endcsname=\z@
6615   \let\noexpand\sza@penalty\relax\noexpand\fi%
6616   \def\noexpand&{%
6617     \noexpand\newverse[] []}%
6618   \def\noexpand\&{\noexpand\@stopstanza}%
6619   \noexpand\@advancestanza@number%
6620   \noexpand\stanza@line[#1]%
6621   \noexpand\@insertstanza@number%
6622   \let\par\relax\ignorespaces%No paragraph in verses
6623 }

```

```

6624 \newcommandx{\stanza}[1][1,usedefault]{%
6625   \ifboolexpr{not test{\ifdefvoid{\at@every@stanza}} and test{\ifstrempy
6626     {#1}}}%
6627     {\@startstanza[\at@every@stanza]}%
6628     {\@startstanza[#1]}%
6629   }%
6630
6631 \newcommandx{\@stopstanza}[1][1,usedefault]{%
6632   \unskip%
6633   \endlock%
6634   \ifboolexpr{not test{\ifdefvoid{\at@every@stop@stanza}} and test{\
6635     ifstrempy{#1}}}%
6636     {\pend[\at@every@stop@stanza]}%
6637     {\pend[#1]}%
6638   \endgroup%
6639   \instanzafalse%
6640 }
6641
6642 \newcommand{\AtEveryStopStanza}[1]{%
6643   \ifstrempy{#1}%
6644     {\xdef\at@every@stop@stanza{}}%
6645     {\gdef\at@every@stop@stanza{#1}}%
6646   }%
6647   \def\at@every@stop@stanza{}%
6648
6649 \newcommand{\AtEveryStanza}[1]{%
6650   \ifstrempy{#1}%
6651     {\xdef\at@every@stanza{}}%
6652     {\gdef\at@every@stanza{#1}}%
6653   }%
6654   \def\at@every@stanza{}%
6655
6656 \newcommandx*{\newverse}[2][1,2,usedefault]{%
6657   \unskip%
6658   \endlock\pend[#1]\sza@penalty\global%
6659   \advance\stanza@count\@ne\stanza@line[#2]%
6660 }
6661
6662 %

```

\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`.

```

6663 \newcommand*{\flagstanza}[2][\stanzaindentbase]{%
6664   \hskip -#1\llap{#2}\hskip #1\ignorespaces}
6665
6666 %

```

XXVIII.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.

```

6667 \catcode`\&=\next
6668 \catcode`\@=\body
6669 \setstanzavalues{szp}{0}
6670
6671 %

```

XXIX Apparatus of Manuscripts

XXIX.1 User level macro

\msdata The user level \msdata command only writes the manuscripts data in numbered auxiliary file.

```

6672 \newcommand{\msdata}[1]{%
6673   \leavevmode%
6674   \unless\ifstopmsdata@inserted@%
6675     \stopmsdata%
6676     \led@warning@msdatawithoutstop%
6677   \fi%
6678   \global\stopmsdata@inserted@false%
6679   \unless\ifledRcol%
6680     \protected@write\linenum@out{}{%
6681       \string\@msd{#1}%
6682     }%
6683   \else%
6684     \protected@write\linenum@outR{}{%
6685       \string\@msd{#1}%
6686     }%
6687   \fi%
6688 }%
6689 %

```

\stopmsdata The user level \stopmsdata command only writes information about the end of manuscripts data in numbered auxiliary file.

```

6690 \newcommand{\stopmsdata}[0]{%
6691   \leavevmode%
6692   \unless\ifledRcol%
6693     \protected@write\linenum@out{}{%
6694       \string\@stopmsd%

```

```

6695 }%
6696 \else%
6697   \protected@write\linenum@outR{}{-%
6698     \string\@stopmsd%
6699   }%
6700 \fi%
6701 \global\stopmsdata@inserted@true%
6702 }%
6703 %

```

\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`

```

6704 \newif\ifstopmsdata@inserted@%
6705 %

```

XXIX.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?

```

6706 \newcommand{\setmsdataseries}[1]{%
6707   \gdef\@msdata@series{#1}%
6708 }%
6709 \def\@msdata@series{A}%
6710 %

```

\setmsdatalabel The label for the manuscripts data.

```

6711 \def\ms@data@label{Ms.}%
6712 \newcommand{\setmsdatalabel}[1]{%
6713   \gdef\ms@data@label{#1}%
6714 }%
6715 %

```

XXIX.3 Counters and lists

\@msd@c `\@msd@c` is a counter incremented at each `\@msd` read in auxiliary file.

```

6716 \numdef{\@msd@c}{0}
6717 \numdef{\@msd@cR}{0}
6718 %

```

\@msd@c `\add@msd@c` is a counter incremented at each `\add@msd@c`, that is at each time we prepare the insertion of manuscripts data footnote.


```

6719 \numdef{\add@msd@c}{0}%
6720 \numdef{\add@msd@cR}{0}%
6721 %

```

\@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.

```

6722 \def\@msdata@list{}%
6723 %

```

XXIX.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.

```

6724 \newcommand{\@msd}[1]{%
6725   \unless\ifledRcol%
6726     \numdef{\@msd@c}{\@msd@c+\@ne}%
6727     \csgdef{\@msdata@\@msd@c @data}{#1}%
6728     \csxdef{\@msdata@\@msd@c @linenumber}{\the\line@num}%
6729     \csxdef{\@msdata@\@msd@c @abslinenumber}{\the\absline@num}%
6730     \xright@appenditem{\the\absline@num}\to\actionlines@list%
6731     \xright@appenditem{-1010}\to\actions@list%
6732   \else%
6733     \numdef{\@msd@cR}{\@msd@cR+\@ne}%
6734     \csgdef{\@msdata@\@msd@cR @dataR}{#1}%
6735     \csxdef{\@msdata@\@msd@cR @linenumberR}{\the\line@numR}%
6736     \csxdef{\@msdata@\@msd@cR @abslinenumberR}{\the\absline@numR}%
6737     \xright@appenditem{\the\absline@numR}\to\actionlines@listR%
6738     \xright@appenditem{-1010}\to\actions@listR%
6739   \fi%
6740 }%
6741 %

```

\@endmsd 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.

```

6742 \newcommand{\@stopmsd}[0]{%
6743   \unless\ifledRcol%
6744     \ifcsundef{\@msdata@\@msd@c @stoplinenumber}{%
6745       \csxdef{\@msdata@\@msd@c @stopabslinenumber}{\the\absline@num}%
6746       \csxdef{\@msdata@\@msd@c @stoplinenumber}{\the\line@num}%
6747     }%
6748   \else%
6749     \ifcsundef{\@msdata@\@msd@cR @stoplinenumberR}{%
6750       \csxdef{\@msdata@\@msd@cR @stopabslinenumberR}{\the\absline@numR}%

```

```

6751 \csxdef{@msdata@ \@msd@cR @stoplinenumberR}{\the\line@numR}%
6752 }%
6753 {}%
6754 \fi%
6755 }%
6756 %

```

XXIX.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 store the manuscript data footnote to \@msdata@list.

```

6757 \newcommand{\add@msdata}{%
6758 \bgroup%
6759 \normalfont%
6760 \unless\ifledRcol@%
6761 \numgdef{\add@msd@c}{\add@msd@c+\@ne}%
6762 \ifcsdef{@msdata@ \@add@msd@c @data}{%
6763 \letcs{\@data}{@msdata@ \@add@msd@c @data}%
6764 \edef\l@d@nums{%
6765 000| % Start page = we don't print it
6766 \csuse{@msdata@ \@add@msd@c @linenumber}| % Start line number
6767 000| % Start subline number, for now, not used
6768 000| % End page number, we don't print it
6769 \ifnumless{\csuse{@msdata@ \@add@msd@c @stopabslinenumber}}{\csuse{
@lastabsline@forpage@ \the\page@num}}%
6770 {\csuse{@msdata@ \@add@msd@c @stoplinenumber}} % End line number if
in the same page
6771 {\csuse{@lastline@forpage@ \the\page@num}} % Otherwiser, last
number of the page
6772 |%
6773 000| % End sub line number, for now, not used
6774 \edfont@info % Font
6775 }%
6776 \@msd@options@fullpagefalse%
6777 \if@firstlineofpage % Try if the data are for the full page. If yes
, will add options to the list.
6778 \unless\if@msdata@insertedfrompreviouspage%
6779 \ifnumless{\csuse{@lastabsline@forpage@ \the\page@num}}{\csuse{
@msdata@ \@add@msd@c @stopabslinenumber}+\@ne}%
6780 {%
6781 \numdef{\@tmp}{\add@msd@c+\@ne}%
6782 \ifcsdef{@msdata@ \@tmp @abslinenumber}%
6783 {\ifnumequal{\csuse{@msdata@ \@tmp @abslinenumber}}{\csuse{
@lastabsline@forpage@ \the\page@num}}%
6784 {}%
6785 {\@msd@options@fullpagetrue}%

```

```

6786         }%
6787         {\msd@options@fullpagetrue}%
6788     }%
6789     {}%
6790     \fi%
6791 \fi%
6792 \listxadd{\msd@list}{%
6793     \msd@options@iffullpage%
6794     \noexpand\csuse{v\msd@data@series footnote}{\msd@data@series}{\
expandonce\l@d@nums}{\msd@data@label}{\expandonce\data}}%
6795     \reset@msd@options@iffullpage%
6796 }%
6797 }%
6798 {}%
6799 \else%
6800 \numgdef{\add@msd@cR}{\add@msd@cR+\@ne}%
6801 \ifcsdef{\msd@data@\add@msd@cR @dataR}{%
6802     \letcs{\@data}{\msd@data@\add@msd@cR @dataR}%
6803     \edef\l@d@nums{%
6804         000| % Start page = we don't print it
6805         \csuse{\msd@data@\add@msd@cR @linenumberR}| % Start line number
6806         000| % Start subline number, for now, not used
6807         000| % End page number, we don't print it
6808         \ifnumless{\csuse{\msd@data@\add@msd@cR @stopabslinenumberR}}{\
csuse{@lastline@forpageR@the\page@numR}}}%
6809         {\csuse{\msd@data@\add@msd@cR @stoplinenumberR}}}% End line number
if in the same page
6810         {\csuse{@lastline@forpageR@the\page@numR}}}% Otherwiser, last
number of the page
6811         |%
6812         000| % End sub line number, for now, not used
6813         \edef\info{Font
6814         }%
6815         \msd@options@fullpagefalse%
6816         \if@firstlineofpageR%
6817             \unless\if@msd@insertedfrompreviouspage%
6818                 \ifnumless{\csuse{@lastabsline@forpageR@the\page@numR}}{\
csuse{\msd@data@\add@msd@c @stopabslinenumberR}+\@ne}%
6819                 {%
6820                     \numdef{\@tmp}{\add@msd@cR+\@ne}%
6821                     \ifcsdef{\msd@data@\@tmp @abslinenumberR}%
6822                         {\ifnumequal{\csuse{\msd@data@\@tmp @abslinenumberR}}{\csuse{
@lastabsline@forpageR@the\page@numR}}}%
6823                         {}%
6824                         {\msd@options@fullpagetrue}%
6825                     }%
6826                     {\msd@options@fullpagetrue}%
6827                 }%
6828                 {}%
6829             \fi%

```

```

6830         \fi%
6831         \listxadd{\@msdata@list}{%
6832             \@msd@options@iffullpage%
6833             \noexpand\csuse{v\@msdata@series footnote}{\@msdata@series}{\@
expandonce\l@d@nums}{\ms@data@label}{\expandonce\data}}%
6834             \reset@msd@options@iffullpage%
6835         }%
6836     }%
6837 }%
6838 \fi%
6839 \egroup%
6840 }%
6841 %

```

`\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.

```

6842 \newif\if@msdata@insertedfrompreviouspage%
6843 %

```

`\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.

```

6844 \newcommand{\add@msdata@firstlineofpage}{%
6845     \bgroup%
6846     \normalfont%
6847     \unless\ifledRcol@%
6848         \ifcsdef{\@msdata@add@msd@c @data}{%
6849             \ifnumless{\the\absline@num-\@ne}{\csuse{\@msdata@\add@msd@c
@stopabslinenumber}}}%
6850             {%
6851                 \global\@msdata@insertedfrompreviouspagetrue%
6852                 \letcs{\@data}{\@msdata@\add@msd@c @data}%
6853                 \edef\l@d@nums{%
6854                     000| % Start page = we don't print it
6855                     \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
6856                     000| % Start subline number, for now, not used
6857                     000| % End page number, we don't print it
6858                     \ifnumless{\csuse{\@msdata@\add@msd@c @stopabslinenumber}}{\csuse{
@lastabsline@forpage@\the\page@num}}%
6859                     {\csuse{\@msdata@\add@msd@c @stoplinenumber}}%End line number if
in the same page
6860                     {\csuse{@lastline@forpage@\the\page@num}}%Otherwise, last
number of the page
6861                     |%
6862                     000| % End sub line number, for now, not used
6863                     \edef\info{Font
6864                     }%

```

```

6865 \msd@options@fullpagefalse%
6866 \ifnumless{\csuse{@lastabsline@forpage@the\page@num}}{\csuse{
\msd@data@add@msd@c @stopabslinenumber}+\@one}%We will test if the ms data is
for the full page
6867 {%
6868 \numdef{\@tmp}{\add@msd@c+\@one}%
6869 \ifcsdef{\msd@data@\@tmp @abslinenumber}%
6870 {\ifnumequal{\csuse{\msd@data@\@tmp @abslinenumber}}{\csuse{
@lastabsline@forpage@the\page@num}}}%
6871 {}%
6872 {\msd@options@fullpagetrue}%
6873 }%
6874 {\msd@options@fullpagetrue}%
6875 }%
6876 {}%
6877 \listxadd{\msd@data@list}{%
6878 \msd@options@iffullpage%
6879 \noexpand\csuse{v\msd@data@series footnote}{\msd@data@series}{\{
expandonce\l@d@nums}{\msd@data@label}{\expandonce\data}}}%
6880 \reset@msd@options@iffullpage%
6881 }%
6882 }%
6883 {\global\msd@data@insertedfrompreviouspagefalse}%
6884 }{%}%
6885 \else%
6886 \ifcsdef{\msd@data@add@msd@cR @dataR}{%
6887 \ifnumless{\the\absline@numR-\@one}{\csuse{\msd@data@add@msd@cR
@stopabslinenumberR}}}%
6888 {%
6889 \global\msd@data@insertedfrompreviouspagetrue%
6890 \letcs{\data}{\msd@data@add@msd@cR @dataR}%
6891 \edef\l@d@nums{%
6892 000}% Start page = we don't print it
6893 \numexpr\the\line@numR+\@one\relax}% Start line number = first
line of the page. As \add@msd@firstlineofpage is called before line
number has been incremented, we increment it for printing
6894 000}% Start subline number, for now, not used
6895 000}% End page number, we don't print it
6896 \ifnumless{\csuse{\msd@data@add@msd@cR @stopabslinenumberR}}{\{
\csuse{@lastline@forpageR@the\page@numR}}}%
6897 {\csuse{\msd@data@add@msd@cR @stoplinenumberR}}}%End line number
if in the same page
6898 {\csuse{@lastline@forpageR@the\page@numR}}}%Otherwise, last
number of the page
6899 |%
6900 000}% End sub line number, for now, not used
6901 \edef@info{Font
6902 }%
6903 \msd@options@fullpagefalse%
6904 \ifnumless{\csuse{@lastabsline@forpageR@the\page@numR}}{\csuse{

```

```

@msdata@\add@msd@cR @stopabslinenumberR}+\@ne}%
6905 { %
6906 \numdef{\@tmp}{\add@msd@cR+\@ne}%
6907 \ifcsdef{@msdata@\@tmp @abslinenumberR}%
6908 {\ifnumequal{\csuse{@msdata@\@tmp @abslinenumberR}}{\csuse{
@lastabsline@forpageR@\the\page@numR}}}%
6909 } %
6910 {\@msd@options@fullpagetrue}%
6911 } %
6912 {\@msd@options@fullpagetrue}%
6913 } %
6914 {} %
6915 \listxadd{\@msdata@list}{ %
6916 \@msd@options@iffullpage%
6917 \noexpand\csuse{v\@msdata@series footnote}{\@msdata@series}{\
expandonce\l@d@nums}{\@msdata@label}{\expandonce\data}}%
6918 \reset@msd@options@iffullpage%
6919 } %
6920 } %
6921 {\global\@msdata@insertedfrompreviouspagefalse}%
6922 }{} %
6923 \fi %
6924 \egroup %
6925 } %
6926 %

```

XXIX.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 27 \newcommand{\insert@msdata}{ %
6928 \def\do##1{##1}%
6929 \dolistloop{\@msdata@list}%
6930 \global\let\@msdata@list\relax%
6931 } %
6932 %

```

XXIX.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.

```

6933 \newif\if@msd@options@fullpage%
6934 \newcommand{\@msd@options@iffullpage}[0]{ %
6935 \if@msd@options@fullpage%
6936 \noexpand\toggletrue{nonum@}%

```

```

6937 \ifdefvoid{\ms@data@label}%
6938 {\noexpand\toggletrue{nosep@}}%
6939 {}%
6940 \fi%
6941 }%
6942 \newcommand{\reset@msd@options@iffullpage}[0]{%
6943 \noexpand\togglefalse{nonum@}%
6944 \noexpand\togglefalse{nosep@}%
6945 }%
6946 %

```

XXX Arrays and tables

XXX.1 Preamble: macro as environment

The following is borrowed, and renamed, from the `amsmath` package. See also the CTT thread ‘`eeq` and `amstex`’, 1995/08/31, started by Keith Reckdahl and ended definitively by David M. Jones.

Several of the `[math]` macros scan their body twice. This means we must collect all text in the body of an environment form before calling the macro.

`\@emptytoks` This is actually defined in the `amsgen` package.

```

6947 \newtoks\@emptytoks
6948
6949 %

```

The rest is from `amsmath`.

`\l@denbody` A token register to contain the body.

```

6950 \newtoks\l@denbody
6951
6952 %

```

`\addtol@denbody` `\addtol@denbody{arg}` adds `arg` to the token register `\l@denbody`.

```

6953 \newcommand{\addtol@denbody}[1]{%
6954 \global\l@denbody\expandafter{\the\l@denbody#1}}
6955
6956 %

```

`\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`.

```

6957 \newcommand{\l@dcollect@body}[1]{%
6958   \l@denbody{\expandafter#1\expandafter{\the\l@denbody}}%
6959   \edef\processl@denbody{\the\l@denbody\noexpand\end{\@currenvir}}%
6960   \l@denbody\@emptytoks \def\l@dbegin@stack{b}%
6961   \begingroup
6962     \expandafter\let\csname\@currenvir\endcsname\l@dcollect@@body
6963     \edef\processl@denbody{\expandafter\noexpand\csname\@currenvir\endcsname}%
6964     \processl@denbody%
6965   }%
6966
6967 %

```

\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.

```

6968 \def\l@dpush@begins#1\begin#2{%
6969   \ifx\end#2\else b\expandafter\l@dpush@begins\fi}
6970
6971 %

```

\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.

```

6972 \def\l@dcollect@@body#1\end#2{%
6973   \edef\l@dbegin@stack{\l@dpush@begins#1\begin\end
6974     \expandafter\gobble\l@dbegin@stack}%
6975   \ifx\@empty\l@dbegin@stack
6976     \endgroup
6977     \@checkend{#2}%
6978     \addtol@denbody{#1}%
6979   \else
6980     \addtol@denbody{#1\end{#2}}%
6981   \fi
6982   \processl@denbody % A little tricky! Note the grouping
6983 }
6984
6985 %

```

There was a question on CTT about how to use \collect@body for a macro taking an argument. The following is part of that thread.

From: Heiko Oberdiek <oberdiek@uni-freiburg.de>
 Newsgroups: comp.text.tex
 Subject: Re: Using \collect@body with commands that take >1 argument

Date: Fri, 08 Aug 2003 09:03:20 +0200

eed132@psu.edu (Evan) wrote:

```
> I'm trying to make a new Latex environment that acts like the>
> \colorbox command that is part of the color package. I looked through
> the FAQ and ran across this bit about using the \collect@body command
> that is part of AMSLaTeX:
> http://www.tex.ac.uk/cgi-bin/texfaq2html?label=cmdasenv
>
> It almost works. If I do something like the following:
> \newcommand{\redbox}[1]{\colorbox{red}{#1}}
>
> \makeatletter
> \newenvironment{redbox}{\collect@body \redbox}{}
```

You will get an error message: Command \redbox already defined.
Thus you must rename either the command \redbox or the environment name.

```
> \begin{coloredbox}{blue}
> Yadda yadda yadda... this is on a blue background...
> \end{coloredbox}
> and can't figure out how to make the \collect@body take this.

> \collect@body \colorbox{red}
> \collect@body {\colorbox{red}}
```

The argument of \collect@body has to be one token exactly.

```
\documentclass{article}
\usepackage{color}
\usepackage{amsmath}

\newcommand{\redbox}[1]{\colorbox{red}{#1}}
\makeatletter
\newenvironment{coloredbox}[1]{%
  \def\next@{\colorbox{#1}}%
  \collect@body\next@
}{%

% ignore spaces at begin and end of environment
\newenvironment{coloredboxII}[1]{%
  \def\next@{\mycoloredbox{#1}}%
  \collect@body\next@
}{%
\newcommand{\mycoloredbox}[2]{%
  \colorbox{#1}{\ignorespaces#2\unskip}%
}

% support of optional color model argument
```

```

\newcommand\coloredboxIII\endcsname{}
\def\coloredboxIII#1#{%
  \@coloredboxIII{#1}%
}
\def\@coloredboxIII#1#2{%
  \def\next@{\mycoloredboxIII{#1}{#2}}%
  \collect@body\next@
}
\newcommand{\mycoloredboxIII}[3]{%
  \colorbox#1{#2}{\ignorespaces#3\unskip}%
}

\makeatother

\begin{document}
  Black text before
  \begin{coloredbox}{blue}
    Hello World
  \end{coloredbox}
  Black text after

  Black text before
  \begin{coloredboxII}{blue}
    Hello World
  \end{coloredboxII}
  Black text after

  Black text before
  \begin{coloredboxIII}[rgb]{0,0,1}
    Hello World
  \end{coloredboxIII}
  Black text after

\end{document}

Yours sincerely
Heiko <oberdiek@uni-freiburg.de>

```

XXX.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`.

XXX.2.1 Disabling and restoring commands

`\l@dtabnoexpands` More no expansion for critical and familiar footnotes in tabular environment.

```

6986 \newcommand*{\l@dtabnoexpands}{%
6987   \let\rtab=0%
6988   \let\ctab=0%
6989   \let\ltab=0%
6990   \let\rtabtext=0%
6991   \let\ltabtext=0%
6992   \let\ctabtext=0%
6993   \let\edbeforetab=0%
6994   \let\edaftertab=0%
6995   \let\edatleft=0%
6996   \let\edatright=0%
6997   \let\edvertline=0%
6998   \let\edvertdots=0%
6999   \let\edrowfill=0%
7000 }
7001
7002 %

```

`\disable@familiarnotes` Macros to disable and restore familiar notes, to prevent them from printing multiple times in edtabularx and edarrayx environments.

`\restore@familiarnotes`

```

7003 \newcommand{\disable@familiarnotes}{%
7004   \unless\ifnofamiliar%
7005     \def\do##1{%
7006       \csletcs{footnote@@##1}{footnote##1}%
7007       \expandafter\renewcommand \csname footnote##1\endcsname[1]{%
7008         \protected@csxdef{@thefnmark##1}{\csuse{thefootnote##1}}%
7009         \csuse{@footnotemark##1}%
7010       }%
7011     }%
7012     \dolistloop{\@series}%
7013   \fi%
7014 }%
7015 \newcommand{\restore@familiarnotes}{%
7016   \unless\ifnofamiliar%
7017     \def\do##1{%
7018       \csletcs{footnote##1}{footnote@@##1}%
7019     }%
7020     \dolistloop{\@series}%
7021   \fi%
7022 }%
7023
7024 %

```

`\disable@sidenotes` The same, for side notes.

`\restore@sidenotes`

```

7025 \newcommand{\disable@sidenotes}{%
7026   \let\@@ledrightnote\ledrightnote%
7027   \let\@@ledleftnote\ledleftnote%
7028   \let\@@ledsidenote\ledsidenote%
7029   \let\ledrightnote\@gobble%
7030   \let\ledleftnote\@gobble%
7031   \let\ledsidenote\@gobble%
7032 }%
7033 \newcommand{\restore@sidenotes}{%
7034   \let\ledrightnote\@@ledrightnote%
7035   \let\ledleftnote\@@ledleftnote%
7036   \let\ledsidenote\@@ledsidenote%
7037 }%
7038 %

```

\disable@notes Disable/restore side and familiar notes.

```

\restore@notes
7039 \newcommand{\disable@notes}{%
7040   \disable@sidenotes%
7041   \disable@familiarnotes%
7042 }%
7043 \newcommand{\restore@notes}{%
7044   \restore@sidenotes%
7045   \restore@familiarnotes%
7046 }%
7047 %

```

\EDTEXT We need to be able to modify the \edtext macros and also restore their original definitions.

```

\edtext
7048 \let\EDTEXT=\edtext
7049 \newcommand{\xedtext}[2]{\EDTEXT{#1}{#2}}
7050 %

```

\EDLABEL We need to be able to modify and restore the \edlabel macro.

```

\edlabel
7051 \let\EDLABEL=\edlabel
7052 \newcommand*\xedlabel[1]{\EDLABEL{#1}}
7053 %

```

\EDINDEX Macros supporting modification and restoration of \edindex.

```

\xedindex
\nulledindex
7054 \let\EDINDEX=\edindex
7055 \newcommand{\xedindex}{\@bsphack%
7056   \@ifnextchar [{\led@index}{\led@index[\jobname]}}
7057 \newcommand{\nulledindex}[2][\jobname]{\@bsphack\esphack}
7058
7059 %

```

`\@line@num` Macro supporting restoration of `\linenum`.

```
7060 \let\@line@num=\linenum
7061 %
```

`\l@dgobblearg` `\l@dgobbleoptarg[⟨arg⟩]{⟨arg⟩}` replaces these two arguments (first is optional) by `\relax`.

```
7062 \newcommand*\l@dgobbleoptarg[2][\relax]{%
7063
7064 %
```

`\Relax` `\let\Relax=\relax`

`\NEXT` `\let\NEXT=\next`

```
7067
7068 %
```

`\l@dmodforedtext` Modify and restore various macros for when `\edtext` is used.
`\l@drestoreforedtext`

```
7069 \newcommand{\l@dmodforedtext}{%
7070 \let\edtext\relax
7071 \def\do##1{\global\csletcs{##1footnote}{\l@dgobbleoptarg}}%
7072 \dolistloop{\@series}%
7073 \let\edindex\nulledindex
7074 \let\linenum\@gobble}
7075 \newcommand{\l@drestoreforedtext}{%
7076 \def\do##1{\global\csletcs{##1footnote}{##1@footnote}}
7077 \dolistloop{\@series}%
7078 \let\edindex\xedindex}
7079 %
```

`\l@dnullfills` Nullify and restore some column fillers, etc.

`\l@drestorefills`

```
7080 \newcommand{\l@dnullfills}{%
7081 \def\edlabel##1{%
7082 \def\edrowfill##1##2##3{%
7083 }
7084 \newcommand{\l@drestorefills}{%
7085 \def\edrowfill##1##2##3{\@EDROWFILL@{##1}{##2}{##3}}%
7086 }
7087
7088 %
```

`\letsforverteilen` Gathers some lets and other code that is common to the `*verteilen*` macros.

```
7089 \newcommand{\letsforverteilen}{%
7090 \let\edtext\xedtext
7091 \let\edindex\xedindex
7092 \def\do##1{\global\csletcs{##1footnote}{##1@footnote}}
```

```

7093 \dolistloop{\@series}%
7094 \let\linenum\@line@num
7095 \hilfsskip=\l@dcwidth%
7096 \advance\hilfsskip by -\wd\hilfsbox
7097 \def\edlabel##1{\xedlabel{##1}}
7098
7099 %

```

`\disablel@dtabfeet` Declarations for using or using `\edtext` inside tabulars. The default at this point is for
`\enablel@dtabfeet` `\edtext`.

```

7100 \newcommand\disablel@dtabfeet{\l@dmodforedtext}%
7101 \newcommand\enablel@dtabfeet{\l@drestoreforedtext}%
7102 %

```

XXX.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.

```

7103 \newcount\l@dampcount
7104 \l@dampcount=1\relax
7105 \newcount\l@dcolcount
7106 \l@dcolcount=0\relax
7107
7108 %

```

`\hilfsbox` Some (temporary) helper items.

```

\hilfsskip
\Hilfsbox
\hilfscount
7109 \newbox\hilfsbox
7110 \newskip\hilfsskip
7111 \newbox\Hilfsbox
7112 \newcount\hilfscount
7113
7114 %

```

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).

```

7115 \newdimen\dcoli
7116 \newdimen\dcolii
7117 \newdimen\dcoliii
7118 \newdimen\dcoliv
7119 \newdimen\dcolv
7120 \newdimen\dcolvi
7121 \newdimen\dcolvii
7122 \newdimen\dcolviii
7123 \newdimen\dcolix
7124 \newdimen\dcolx
7125 \newdimen\dcolxi

```

```

7126 \newdimen\dcplxii
7127 \newdimen\dcplxiii
7128 \newdimen\dcplxiv
7129 \newdimen\dcpl xv
7130 \newdimen\dcplxvi
7131 \newdimen\dcplxvii
7132 \newdimen\dcplxviii
7133 \newdimen\dcplxix
7134 \newdimen\dcplx x
7135 \newdimen\dcplx xi
7136 \newdimen\dcplx xii
7137 \newdimen\dcplx xiii
7138 \newdimen\dcplx xiv
7139 \newdimen\dcplx xv
7140 \newdimen\dcplx xvi
7141 \newdimen\dcplx xvii
7142 \newdimen\dcplx xviii
7143 \newdimen\dcplx xix
7144 \newdimen\dcplx x x
7145 \newdimen\dcplxerr % added for error handling
7146
7147 %

```

\l@dcolwidth This is a cunning way of storing the columnwidths indexed by the column number **\l@dcolcount**, like an array. (was **\Dimenzuordnung**)

```

7148 \newcommand{\l@dcolwidth}{\ifcase \the\l@dcolcount \dcoli %???
7149 \or \dcoli \or \dcolii \or \dcoliii
7150 \or \dcoliv \or \dcolv \or \dcolvi
7151 \or \dcolvii \or \dcolviii \or \dcolix \or \dcolx
7152 \or \dcolxi \or \dcolxii \or \dcolxiii
7153 \or \dcolxiv \or \dcolxv \or \dcolxvi
7154 \or \dcolxvii \or \dcolxviii \or \dcolxix \or \dcolxx
7155 \or \dcolxxi \or \dcolxxii \or \dcolxxiii
7156 \or \dcolxxiv \or \dcolxxv \or \dcolxxvi
7157 \or \dcolxxvii \or \dcolxxviii \or \dcolxxix \or \dcolxxx
7158 \else \dcplxerr \fi}
7159
7160 %

```

\step\l@dcolcount This increments the column counter, and issues an error message if it is too large.

```

7161 \newcommand*{\step\l@dcolcount}{\advance\l@dcolcount\@ne
7162 \ifnum\l@dcolcount>30\relax
7163 \led@err@TooManyColumns
7164 \fi}
7165
7166 %

```

\l@dsetmaxcolwidth Sets the column width to the maximum value seen so far.

```

7167 \newcommand{\l@dssetmaxcolwidth}{%
7168   \ifdim\l@dcolwidth < \wd\hilfsbox
7169     \l@dcolwidth = \wd\hilfsbox
7170   \else \relax \fi}
7171
7172 %

```

\measuremcell Measure (recursively) the width required for a math cell.

```

7173 \def\measuremcell #1{%
7174   \ifx #1\ \ifnum\l@dcolcount=0\let\NEXT\relax%
7175     \else\l@dcheckcols%
7176       \l@dcolcount=0%
7177       \let\NEXT\measuremcell%
7178     \fi%
7179   \else\setbox\hilfsbox=\hbox{$\displaystyle{#1}$}%
7180     \step1@dcolcount%
7181     \l@dssetmaxcolwidth%
7182     \let\NEXT\measuremcell%
7183   \fi\NEXT}
7184
7185 %

```

\measuretcell Measure (recursively) the width required for a text cell.

```

7186 \def\measuretcell #1{%
7187   \ifx #1\ \ifnum\l@dcolcount=0\let\NEXT\relax%
7188     \else\l@dcheckcols%
7189       \l@dcolcount=0%
7190       \let\NEXT\measuretcell%
7191     \fi%
7192   \else\setbox\hilfsbox=\hbox{#1}%
7193     \step1@dcolcount%
7194     \l@dssetmaxcolwidth%
7195     \let\NEXT\measuretcell%
7196   \fi\NEXT}
7197
7198 %

```

\measuremrow Measure (recursively) the width required for a math row.

```

7199 \def\measuremrow #1\{%
7200   \ifx #1\let\NEXT\relax%
7201   \else\measuremcell #1\&\&\&%
7202     \let\NEXT\measuremrow%
7203   \fi\NEXT}
7204 %

```

\measuretrow Measure (recursively) the width required for a text row.


```

7205 \def\measuretrow #1\{f%
7206 \ifx #1&\let\NEXT\relax%
7207 \else\measuretcell #1&\&\&%
7208 \let\NEXT\measuretrow%
7209 \fi\NEXT}
7210
7211 %

```

\edtabcolsep The length `\edtabcolsep` controls the distance between columns.

```

7212 \newskip\edtabcolsep
7213 \global\edtabcolsep=10pt
7214
7215 %

```

```

\variab16 \newcommand{\variab}{\relax}
7217
7218 %

```

\l@dccheckcols Check that the number of columns is consistent.

```

7219 \newcommand*{\l@dccheckcols}{f%
7220 \ifnum\l@dcolcount=1\relax
7221 \else
7222 \ifnum\l@dampcount=1\relax
7223 \else
7224 \ifnum\l@dcolcount=\l@dampcount\relax
7225 \else
7226 \l@d@err@UnequalColumns
7227 \fi
7228 \fi
7229 \l@dampcount=\l@dcolcount
7230 \fi}
7231
7232 %

```

\edfilldimen A length.

```

7233 \newdimen\edfilldimen
7234 \edfilldimen=0pt
7235
7236 %

```

\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

```

7237 \newcounter{addcolcount}
7238 \renewcommand{\theadcolcount}{\romannumeral \c@addcolcount}
7239 %

```

XXX.2.3 Tabular typesetting

`\setmcellright` Typeset (recursively) cells of display math right justified.

```

7240 \def\setmcellright #1{\def\edlabel##1{}%
7241     \let\edindex\nulledindex
7242     \ifx #1\ \ifnum\l@dc@colcount=0%\removelastskip
7243         \let\Next\relax%
7244     \else\l@dc@colcount=0%
7245         \let\Next=\setmcellright%
7246     \fi%
7247 \else%
7248     \disablel@dtabfeet%
7249     \step1@dc@colcount%
7250     \disable@notes%
7251     \setbox\hilfsbox=\hbox{$\displaystyle{#1}$}%
7252     \restore@notes%
7253     \letsforverteilen%
7254     \hskip\hilfsskip$\displaystyle{#1}$%
7255     \hskip\edtabcolsep%
7256     \let\Next=\setmcellright%
7257 \fi\Next}
7258
7259 %

```

`\settcclright` Typeset (recursively) cells of text right justified.

```

7260 \def\settcclright #1{\def\edlabel##1{}%
7261     \let\edindex\nulledindex
7262     \ifx #1\ \ifnum\l@dc@colcount=0%\removelastskip
7263         \let\Next\relax%
7264     \else\l@dc@colcount=0%
7265         \let\Next=\settcclright%
7266     \fi%
7267 \else%
7268     \disablel@dtabfeet%
7269     \step1@dc@colcount%
7270     \disable@notes%
7271     \setbox\hilfsbox=\hbox{#1}%
7272     \restore@notes%
7273     \letsforverteilen%
7274     \hskip\hilfsskip#1%
7275     \hskip\edtabcolsep%
7276     \let\Next=\settcclright%
7277 \fi\Next}
7278 %

```

`\setmcelleft` Typeset (recursively) cells of display math left justified.

```

7279 \def\setmcelleft #1{\def\edlabel##1{}%

```

```

7280 \let\edindex\nulledindex
7281 \ifx #1\ \ifnum\l@dc@colcount=0 \let\Next\relax%
7282       \else\l@dc@colcount=0%
7283       \let\Next=\setmcellleft%
7284       \fi%
7285 \else \disablel@dtabfeet%
7286       \stepl@dc@colcount%
7287       \disable@notes%
7288       \setbox\hilfsbox=\hbox{$\displaystyle{#1}$}%
7289       \restore@notes%
7290       \letsforverteilen%
7291       $\displaystyle{#1}$\hskip\hilfsskip\hskip\edtabcolsep%
7292       \let\Next=\setmcellleft%
7293 \fi\Next}
7294
7295 %

```

\settcclleft Typeset (recursively) cells of text left justified.

```

7296 \def\settcclleft #1&{\def\edlabel##1{}%
7297   \let\edindex\nulledindex
7298   \ifx #1\ \ifnum\l@dc@colcount=0 \let\Next\relax%
7299     \else\l@dc@colcount=0%
7300     \let\Next=\settcclleft%
7301     \fi%
7302   \else \disablel@dtabfeet%
7303         \stepl@dc@colcount%
7304         \disable@notes%
7305         \setbox\hilfsbox=\hbox{#1}%
7306         \restore@notes%
7307         \letsforverteilen%
7308         #1\hskip\hilfsskip\hskip\edtabcolsep%
7309         \let\Next=\settcclleft%
7310   \fi\Next}
7311 %

```

\setmcellcenter Typeset (recursively) cells of display math centered.

```

7312 \def\setmcellcenter #1&{\def\edlabel##1{}%
7313   \let\edindex\nulledindex
7314   \ifx #1\ \ifnum\l@dc@colcount=0\let\Next\relax%
7315     \else\l@dc@colcount=0%
7316     \let\Next=\setmcellcenter%
7317     \fi%
7318   \else \disablel@dtabfeet%
7319         \stepl@dc@colcount%
7320         \disable@notes%
7321         \setbox\hilfsbox=\hbox{$\displaystyle{#1}$}%
7322         \restore@notes%
7323         \letsforverteilen%

```

```

7324         \hskip 0.5\hlfsskip$\displaystyle{#1}$\hskip0.5\hlfsskip%
7325         \hskip\edtabcolsep%
7326         \let\Next=\setmcellcenter%
7327     \fi\Next}
7328
7329 %

```

\settccllcenter Typeset (recursively) cells of text centered.

```

7330 \def\settccllcenter #1{\def\edlabel##1{}%
7331     \let\edindex\nulledindex
7332     \ifx #1\ \ifnum\l@dc@lcount=0 \let\Next\relax%
7333         \else\l@dc@lcount=0%
7334             \let\Next=\settccllcenter%
7335         \fi%
7336     \else \disablel@dt@bfeet%
7337         \step1@dc@lcount%
7338         \disable@notes%
7339         \setbox\hlf@b@x=\hbox{#1}%
7340         \restore@notes%
7341         \letsforverteilen%
7342         \hskip 0.5\hlfsskip #1\hskip 0.5\hlfsskip%
7343         \hskip\edtabcolsep%
7344         \let\Next=\settccllcenter%
7345     \fi\Next}
7346
7347 %

```

\NEXT 48 \let\NEXT=\relax

```

7349
7350 %

```

\setmrowright Typeset (recursively) rows of right justified math.

```

7351 \def\setmrowright #1\{%
7352     \ifx #1& \let\NEXT\relax
7353     \else \centerline{\setmcellright #1&\&\&}
7354         \let\NEXT=\setmrowright
7355     \fi\NEXT}
7356 %

```

\settroright Typeset (recursively) rows of right justified text.

```

7357 \def\settroright #1\{%
7358     \ifx #1& \let\NEXT\relax
7359     \else \centerline{\settccllright #1&\&\&}
7360         \let\NEXT=\settroright
7361     \fi\NEXT}
7362
7363 %

```

`\setmrowleft` Typeset (recursively) rows of left justified math.

```

7364 \def\setmrowleft #1\{\{
7365   \ifx #1& \let\NEXT\relax
7366   \else \centerline{\setmcellleft #1&\&\&\&}
7367   \let\NEXT=\setmrowleft
7368   \fi\NEXT}
7369 %

```

`\settrrowleft` Typeset (recursively) rows of left justified text.

```

7370 \def\settrrowleft #1\{\{
7371   \ifx #1& \let\NEXT\relax
7372   \else \centerline{\settrcellleft #1&\&\&\&}
7373   \let\NEXT=\settrrowleft
7374   \fi\NEXT}
7375 %
7376 %

```

`\setmrowcenter` Typeset (recursively) rows of centered math.

```

7377 \def\setmrowcenter #1\{\{
7378   \ifx #1& \let\NEXT\relax%
7379   \else \centerline{\setmcellcenter #1&\&\&\&}
7380   \let\NEXT=\setmrowcenter
7381   \fi\NEXT}
7382 %

```

`\settrrowcenter` Typeset (recursively) rows of centered text.

```

7383 \def\settrrowcenter #1\{\{
7384   \ifx #1& \let\NEXT\relax
7385   \else \centerline{\settrcellcenter #1&\&\&\&}
7386   \let\NEXT=\settrrowcenter
7387   \fi\NEXT}
7388 %
7389 %

```

`\nullsetzen`⁹⁰ `\newcommand{\nullsetzen}{\{%`

```

7391   \step1@dcolcount%
7392   \l@dcolwidth=0pt%
7393   \ifnum\l@dcolcount=30\let\NEXT\relax%
7394   \l@dcolcount=0\relax
7395   \else\let\NEXT\nullsetzen%
7396   \fi\NEXT}
7397 %
7398 %

```

\edatleft `\edatleft[$\langle math \rangle$]{ $\langle symbol \rangle$ }{ $\langle len \rangle$ }`. Left $\langle symbol \rangle$, $2\langle len \rangle$ high with prepended $\langle math \rangle$ vertically centered.

```
7399 \newcommand{\edatleft}[3][\@empty]{%
7400   \ifx#1\@empty
7401     \vbox to 10pt{\vss\hbox{\$ \left#2\vrule width0pt height #3
7402                           depth 0pt \right. \$\hss}\vfil}
7403   \else
7404     \vbox to 4pt{\vss\hbox{\$#1\left#2\vrule width0pt height #3
7405                           depth 0pt \right. \$}\vfil}
7406   \fi}
7407 %
```

\edatright `\edatright[$\langle math \rangle$]{ $\langle symbol \rangle$ }{ $\langle len \rangle$ }`. Right $\langle symbol \rangle$, $2\langle len \rangle$ high with appended $\langle math \rangle$ vertically centered.

```
7408 \newcommand{\edatright}[3][\@empty]{%
7409   \ifx#1\@empty
7410     \vbox to 10pt{\vss\hbox{\$ \left.\vrule width0pt height #3
7411                           depth 0pt \right#2 \$\hss}\vfil}
7412   \else
7413     \vbox to 4pt{\vss\hbox{\$ \left.\vrule width0pt height #3
7414                           depth 0pt \right#2 #1 \$}\vfil}
7415   \fi}
7416 %
7417 %
```

\edvertline `\edvertline{ $\langle len \rangle$ }` vertical line $\langle len \rangle$ high.

```
7418 \newcommand{\edvertline}[1]{\vbox to 8pt{\vss\hbox{\vrule height #1}\vfil}}
7419 %
7420 %
```

\edvertdots `\edvertdots{ $\langle len \rangle$ }` vertical dotted line $\langle len \rangle$ high.

```
7421 \newcommand{\edvertdots}[1]{\vbox to 1pt{\vss\vbox to #1%
7422   {\cleaders\hbox{\$ \m@th\hbox{.}\vbox to 0.5em{ }\$}\vfil}}}
7423 %
7424 %
```

\l@dtabaddcols `\l@dtabaddcols{ $\langle startcol \rangle$ }{ $\langle endcol \rangle$ }` adds the widths of the columns $\langle startcol \rangle$ through $\langle endcol \rangle$ to `\edfilldimen`. It is a L^AT_EX style reimplementation of the original `\@add@`.

```
7425 \newcommand{\l@dtabaddcols}[2]{%
7426   \l@dccheckstartend{#1}{#2}%
7427   \ifl@dstartendok
7428     \setcounter{addcolcount}{#1}%
7429     \@whilenum \value{addcolcount}<#2\relax \do
7430     {\advance\edfilldimen by \the\csname dcol\theaddcolcount\endcsname
7431      \advance\edfilldimen by \edtabcolsep}
```

```

7432 \stepcounter{addcolcount}}}%
7433 \advance\edfilldimen by \the \csname dcol\theaddcolcount\endcsname
7434 \fi
7435 }
7436
7437 %

```

`\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.

```

7438 \newif\ifl@dstartendok
7439 \newcommand{\l@dcheckstartend}[2]{%
7440 \l@dstartendoktrue
7441 \ifnum #1<\@ne
7442 \l@dstartendokfalse
7443 \led@err@LowStartColumn
7444 \fi
7445 \ifnum #2>30\relax
7446 \l@dstartendokfalse
7447 \led@err@HighEndColumn
7448 \fi
7449 \ifnum #1>#2\relax
7450 \l@dstartendokfalse
7451 \led@err@ReverseColumns
7452 \fi
7453 }
7454
7455 %

```

`\edrowfill` `\edrowfill{⟨startcol⟩}{⟨endcol⟩}` fill fills columns `⟨startcol⟩` to `⟨endcol⟩` inclusive with `⟨fill⟩` (e.g. `\hrulefill`, `\upbracefill`). This is a \LaTeX style reimplementa-
`\@edrowfill@` tion and generalization of the original `\waklam`, `\Waklam`, `\waklamec`, `\wastricht` and
`\@EDROWFILL@` `\wapunktel` macros.

```

7456 \newcommand*{\edrowfill}[3]{%
7457 \l@dtabaddcols{#1}{#2}%
7458 \hb@xt@ \the\l@dcolwidth{\hb@xt@ \the\edfilldimen{#3}\hss}}
7459 \let\@edrowfill@=\edrowfill
7460 \def\@EDROWFILL@#1#2#3{\@edrowfill@{#1}{#2}{#3}}
7461
7462 %

```

`\edbeforetab` The macro `\edbeforetab{⟨text⟩}{⟨math⟩}` puts `⟨text⟩` at the left margin before
`\edaftertab` 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`.

```

7463 \newcommand{\leftltab}[1]{%
7464   \hb@xt@z@{\vbox{\edtabindent%
7465     \moveleft\Hilfsskip\hbox{\ #1}}\hss}}
7466
7467 %

```

\lefttab `\lefttab{<text>}{<math>}` for `\edbeforetab` in `\rtab`.

```

7468 \newcommand{\lefttab}[2]{%
7469   #2\hb@xt@z@{\vbox{\edtabindent%
7470     \advance\Hilfsskip by\dcoli%
7471     \moveleft\Hilfsskip\hbox{\ #1}}\hss}}
7472
7473 %

```

\leftctab `\leftctab{<text>}{<math>}` for `\edbeforetab` in `\ctab`.

```

7474 \newcommand{\leftctab}[2]{%
7475   \hb@xt@z@{\vbox{\edtabindent\l@dcolcount=\l@dampcount%
7476     \advance\Hilfsskip by 0.5\dcoli%
7477     \setbox\hilfsbox=\hbox{\def\edlabel##1{}%
7478       \disablel@dtabfeet$\displaystyle{#2}$}%
7479     \advance\Hilfsskip by -0.5\wd\hilfsbox%
7480     \moveleft\Hilfsskip\hbox{\ #1}}\hss}%
7481   #2}
7482
7483 %

```

\rightctab `\rightctab{<math>}{<text>}` for `\edaftertab` in `\ctab`.

```

7484 \newcommand{\rightctab}[2]{%
7485   \setbox\hilfsbox=\hbox{\def\edlabel##1{}%
7486     \disablel@dtabfeet#2}\l@dampcount=\l@dcolcount%
7487   #1\hb@xt@z@{\vbox{\edtabindent\l@dcolcount=\l@dampcount%
7488     \advance\Hilfsskip by 0.5\l@dcolwidth%
7489     \advance\Hilfsskip by -\wd\hilfsbox%
7490     \setbox\hilfsbox=\hbox{\def\edlabel##1{}%
7491       \disablel@dtabfeet$\displaystyle{#1}$}%
7492     \advance\Hilfsskip by -0.5\wd\hilfsbox%
7493     \advance\Hilfsskip by \edtabcolsep%
7494     \moveright\Hilfsskip\hbox{\ #2}}\hss}%
7495   }
7496
7497 %

```

\rightltab `\rightltab{<math>}{<text>}` for `\edaftertab` in `\ltab`.

```

7498 \newcommand{\rightltab}[2]{%
7499   \setbox\hilfsbox=\hbox{\def\edlabel##1{}%
7500     \disablel@dtabfeet#2}\l@dampcount=\l@dcolcount%

```



```

7501      #1\hb@xt@\z@\vbox{\edtabindent\l@dcolcount=\l@dampcount%
7502      \advance\Hilfsskip by\l@dcolwidth%
7503      \advance\Hilfsskip by-\wd\hilfsbox%
7504      \setbox\hilfsbox=\hbox{\def\edlabel##1{}}%
7505      \disablel@dtabfeet$\displaystyle{#1}$}%
7506      \advance\Hilfsskip by-\wd\hilfsbox%
7507      \advance\Hilfsskip by\edtabcolsep%
7508      \moveright\Hilfsskip\hbox{ #2}}\hss}%
7509    }
7510
7511  %

```

\righttab `\righttab{<math>}<{<text>}` for `\edaftertab` in `\rtab`.

```

7512 \newcommand{\righttab}[2]{%
7513   \setbox\hilfsbox=\hbox{\def\edlabel##1{}}%
7514   \disablel@dtabfeet#2}%
7515   #1\hb@xt@\z@\vbox{\edtabindent%
7516   \advance\Hilfsskip by-\wd\hilfsbox%
7517   \advance\Hilfsskip by\edtabcolsep%
7518   \moveright\Hilfsskip\hbox{ #2}}\hss}%
7519 }
7520
7521 %

```

\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.

```

7522 \newcommand{\rtab}[1]{%
7523   \l@dnnullfills
7524   \def\edbeforetab##1##2{\lefttab{##1}{##2}}%
7525   \def\edaftertab##1##2{\righttab{##1}{##2}}%
7526   \measurebody{#1}%
7527   \l@drestorefills
7528   \variab
7529   \setmrowright #1\&\&%
7530   \enablel@dtabfeet}
7531
7532 %

```

\measurebody `\measurebody{<body>}` measures the array `<body>`.

```

7533 \newcommand{\measurebody}[1]{%
7534   \disablel@dtabfeet%
7535   \l@dcolcount=0%
7536   \nullsetzen%
7537   \l@dcolcount=0
7538   \measuremrow #1\&\&%
7539   \global\l@dampcount=1}

```

```
7540
7541 %
```

\rtabtext `\rtabtext{⟨body⟩}` typesets *⟨body⟩* as a tabular with the entries right justified.

```
7542 \newcommand{\rtabtext}[1]{%
7543   \l@dnnullfills
7544   \measuretbody{#1}%
7545   \l@drestorefills
7546   \variab
7547   \settroright #1\\&\\%
7548   \enablel@dtabfeet}
7549
7550 %
```

\measuretbody `\measuretbody{⟨body⟩}` measures the tabular *⟨body⟩*.

```
7551 \newcommand{\measuretbody}[1]{%
7552   \disable@notes%
7553   \disablel@dtabfeet%
7554   \l@dcolcount=0%
7555   \nullsetzen%
7556   \l@dcolcount=0
7557   \measuretror #1\\&\\%
7558   \restore@notes%
7559   \global\l@dampcount=1}
7560
7561 %
```

\ltab Array with entries left justified.

\edbeforetab

\edaftertab

```
7562 \newcommand{\ltab}[1]{%
7563   \l@dnnullfills
7564   \def\edbeforetab##1##2{\leftltab{##1}{##2}}%
7565   \def\edaftertab##1##2{\rightltab{##1}{##2}}%
7566   \measuretbody{#1}%
7567   \l@drestorefills
7568   \variab
7569   \setmrowleft #1\\&\\%
7570   \enablel@dtabfeet}
7571
7572 %
```

\ltabtext Tabular with entries left justified.

```
7573 \newcommand{\ltabtext}[1]{%
7574   \l@dnnullfills
7575   \measuretbody{#1}%
7576   \l@drestorefills
7577   \variab
```

```

7578 \settroughleft #1\\&\\%
7579 \enablel@dtabfeet}
7580
7581 %

```

\ctab Array with centered entries.

```

\edbeforetab
\edaftertab
7582 \newcommand{\ctab}[1]{%
7583 \l@dnnullfills
7584 \def\edbeforetab##1##2{\leftctab{##1}{##2}}%
7585 \def\edaftertab##1##2{\rightctab{##1}{##2}}%
7586 \measuretbody{#1}%
7587 \l@drestorefills
7588 \variab
7589 \setmrowcenter #1\\&\\%
7590 \enablel@dtabfeet}
7591
7592 %

```

\ctabtext Tabular with entries centered.

```

7593 \newcommand{\ctabtext}[1]{%
7594 \l@dnnullfills
7595 \measuretbody{#1}%
7596 \l@drestorefills
7597 \variab
7598 \settroughcenter #1\\&\\%
7599 \enablel@dtabfeet}
7600
7601 %

```

```

\spreadtext02 \newcommand{\spreadtext}[1]{%\l@dcolcount=\l@dampcount%
7603 \hb@xt@ \the\l@dcolwidth{\hbox{#1}\hss}}
7604 %

```

```

\spreadmath05 \newcommand{\spreadmath}[1]{%
7606 \hb@xt@ \the\l@dcolwidth{\hbox{$\displaystyle{#1}$}\hss}}
7607
7608 %

```

\HILFSskip More helpers.

```

\Hilfsskip
7609 \newskip\HILFSskip
7610 \newskip\Hilfsskip
7611
7612 %

```

```

\EDTABINDENT \newcommand{\EDTABINDENT}{%
7614   \ifnum\l@dcolcount=30\let\NEXT\relax\l@dcolcount=0%
7615   \else\step1@dcolcount%
7616     \advance\Hilfsskip by\l@dcolwidth%
7617     \ifdim\l@dcolwidth=0pt\advance\hilfscout\@ne
7618     \else\advance\Hilfsskip by \the\hilfscout\edtabcolsep%
7619     \hilfscout=1\fi%
7620     \let\NEXT=\EDTABINDENT%
7621   \fi\NEXT}%
7622 %

```

`\edtabindent` (was `\tabindent`)

```

7623 \newcommand{\edtabindent}{%
7624   \l@dcolcount=0\relax
7625   \Hilfsskip=0pt%
7626   \hilfscout=1\relax
7627   \EDTABINDENT%
7628   \hilfsskip=\hsize%
7629   \advance\hilfsskip -\Hilfsskip%
7630   \Hilfsskip=0.5\hilfsskip%
7631   }%
7632 %
7633 %

```

`\EDTAB` (was `\TAB`)

```

7634 \def\EDTAB #1|#2|{%
7635   \setbox\tabhilfbox=\hbox{$\displaystyle{#1}$}%
7636   \setbox\tabHilfbox=\hbox{$\displaystyle{#2}$}%
7637   \advance\tabelskip -\wd\tabhilfbox%
7638   \advance\tabelskip -\wd\tabHilfbox%
7639   \unhbox\tabhilfbox\hskip\tabelskip%
7640   \unhbox\tabHilfbox}%
7641 %
7642 %

```

`\EDTABtext` (was `\TABtext`)

```

7643 \def\EDTABtext #1|#2|{%
7644   \setbox\tabhilfbox=\hbox{#1}%
7645   \setbox\tabHilfbox=\hbox{#2}%
7646   \advance\tabelskip -\wd\tabhilfbox%
7647   \advance\tabelskip -\wd\tabHilfbox%
7648   \unhbox\tabhilfbox\hskip\tabelskip%
7649   \unhbox\tabHilfbox}%
7650 %

```

`\tabhilfbox` Further helpers.

`\tabHilfbox`

edarrayl edarrayc edarrayr The ‘environment’ forms for \ltab, \ctab and \rtab.

edtabularl edtabularc edtabularr The ‘environment’ forms for `\ltabtext`, `\ctabtext` and `\rtabtext`.

`\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.

```

7665 \quotation
7666 \endquotation
7667 \quote
7668 \endquote
7669 \initnumbering@quote defines quotation environment.
7670
7671 \newcommand{\initnumbering@quote}{
7672   \ifnoquotation@else
7673     \renewcommand{\quotation}{\par\leavevmode%
7674       \parindent=1.5em%
7675       \skipnumbering%
7676       \ifautopar%
7677         \vskip-\parskip%
7678       \else%
7679         \vskip\topsep%
7680       \fi%
7681       \global\leftskip=\leftmargin%
7682       \global\rightskip=\leftmargin%
7683     }
7684   \renewcommand{\endquotation}{\par%
7685     \global\leftskip=0pt%
7686     \global\rightskip=0pt%
7687     \leavevmode%

```

```

7682         \skipnumbering%
7683         \ifautopar%
7684             \vskip-\parskip%
7685         \else%
7686             \vskip\topsep%
7687         \fi%
7688     }
7689     \renewcommand{\quote}{\par\leavevmode%
7690         \parindent=0pt%
7691         \skipnumbering%
7692         \ifautopar%
7693             \vskip-\parskip%
7694         \else%
7695             \vskip\topsep%
7696         \fi%
7697         \global\leftskip=\leftmargin%
7698         \global\rightskip=\leftmargin%
7699     }
7700     \renewcommand{\endquote}{\par%
7701         \global\leftskip=0pt%
7702         \global\rightskip=0pt%
7703         \leavevmode%
7704         \skipnumbering%
7705         \ifautopar%
7706             \vskip-\parskip%
7707         \else%
7708             \vskip\topsep%
7709         \fi%
7710     }
7711     \fi
7712 }
7713 %

```

XXXII Section's title commands

XXXII.1 Commands to disable some feature

\ledsectnotoc The `\ledsectnotoc` only disables the `\addcontentsline` macro.

```

7714 \newcommand{\ledsectnotoc}{\let\addcontentsline@gobblethree}
7715 %

```

\ledsectnomark The `\ledsectnomark` only disables the `\chaptermark`, `\sectionmark` and `\subsectionmark` macros.

```

7716 \newcommand{\ledsectnomark}{%
7717     \let\chaptermark@gobble%
7718     \let\sectionmark@gobble%
7719     \let\subsectionmark@gobble%

```

```
7720 }
7721 %
```

XXXII.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 \TeX 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 \TeX sectioning command.
4. This last command is called when the `pstart` is effectively printed.

XXXII.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).

```
7722 \catcode`\#=12
7723 \notbool{@noeled@sec}{%
7724 %
```

`\beforeeledchapter` For technical reasons, not yet solved, page-breaking before chapters can't be made automatically by `eledmac`. Users have to use `\beforeeledchapter`.

```
7725 \ifl@dmemoir
7726 \newcommand\beforeeledchapter{%
7727 \clearforchapter%
7728 }
7729 \else
```

```

7730 \newcommand\beforeeledchapter{%
7731 \if@openright%
7732 \cleardoublepage%
7733 \else%
7734 \clearpage%
7735 \fi%
7736 }
7737 \fi
7738 %

```

XXXII.4 Auxiliary commands

\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.

```

7739 \newif\if@eled@sectioning
7740 %

```

\print@leftmargin@eledsection **\print@rightmargin@eledsection** `\print@leftmargin@eledsection` and `\print@rightmargin@eledsection` are added by `reledmac` inside the code of sectioning command, in order to affix lines numbers. They include tests for RTL languages.

```

7741 \def\print@rightmargin@eledsection{%
7742 \if@eled@sectioning%
7743 \begingroup%
7744 \if@RTL%
7745 \let\llap\rlap%
7746 \let\leftlinenum\rightlinenum%
7747 \let\leftlinenumR\rightlinenumR%
7748 \let\l@drd@ta\l@dld@ta%
7749 \let\l@drsn@te\l@dlsn@te%
7750 \fi%
7751 \hfill\l@drd@ta \csuse{LR}{\l@drsn@te}%
7752 \endgroup%
7753 \fi%
7754 }%
7755
7756 \def\print@leftmargin@eledsection{%
7757 \if@eled@sectioning%
7758 \leavevmode%
7759 \begingroup%
7760 \if@RTL%
7761 \let\rlap\llap%
7762 \let\rightlinenum\leftlinenum%
7763 \let\rightlinenumR\leftlinenumR%
7764 \let\l@dld@ta\l@drd@ta%
7765 \let\l@dlsn@te\l@drsn@te%
7766 \fi%

```



```

7767 \l@dld@ta\csuse{LR}{\l@dlsn@te}%
7768 \endgroup%
7769 \fi%
7770 }%
7771
7772 %

```

XXXII.5 Patching standard commands

`\chapter` We have to patch `LATEX`, book and memoir sectioning commands in order to:

- `\M@sect`
 - Disable `\edtext` inside.
- `\@mem@old@ssect`
 - Disable page breaking (for `\chapter`).
- `\@makechapterhead`
 - Add line numbers and sidenotes.
- `\@makechapterhead`
- `\@makeschapterhead`

`\@sect` Unfortunately, Maïeul Rouquette was not able to try if memoir is loaded. That is why
`\@ssect` `eledmac` tries to define for both standard class and memoir class.

```

7773 \AtBeginDocument{%
7774 \patchcmd{\chapter}{\clearforchapter}{%
7775 \if@eled@sectioning\else%
7776 \ifl@dprintingpages\else%
7777 \clearforchapter%
7778 \fi%
7779 \fi%
7780 }
7781 {}
7782 {}
7783
7784
7785 \pretocmd{\M@sect}
7786 {\let\old@edtext=\edtext%
7787 \let\edtext=\dummy@edtext@showlemma%
7788 }
7789 {}
7790 {}
7791
7792 \apptocmd{\M@sect}
7793 {\let\edtext=\old@edtext}
7794 {}
7795 {}
7796
7797 \patchcmd{\M@sect}
7798 { #9}
7799 { #9%
7800 \print@rightmargin@eledsection%
7801 }
7802 {}

```

```

7803 {}
7804
7805 \patchcmd{\M@sect}
7806 {\hskip #3\relax}
7807 {\hskip #3\relax%
7808 \print@leftmargin@eledsection%
7809 }
7810 {}
7811 {}
7812
7813 \patchcmd{\@mem@old@ssect}
7814 {#5}
7815 {#5%
7816 \print@leftmargin@eledsection%
7817 }
7818 {}
7819 {}
7820
7821 \patchcmd{\@mem@old@ssect}
7822 {\hskip #1}
7823 {\hskip #1%
7824 \print@rightmargin@eledsection%
7825 }
7826 {}
7827 {}
7828
7829 \patchcmd{\chapter}{\if@openright\cleardoublepage\else\clearpage\fi}{%
7830 \if@eled@sectioning\else%
7831 \ifl@dprintingpages\else%
7832 \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
7833 \fi%
7834 \fi%
7835 }%
7836 {}%
7837 {}%
7838
7839 \patchcmd{\scr@startchapter}{\if@openright\cleardoublepage\else\clearpage\
fi}{%
7840 \if@eled@sectioning\else%
7841 \ifl@dprintingpages\else%
7842 \if@openright\cleardoublepage\else\clearpage\fi}%No clearpage inside a
\Pages: will keep critical notes from printing on the title page. Here for
scrbook.
7843 \fi%
7844 \fi%
7845 }
7846 {}
7847 {}

```

```

7848 \patchcmd{\@makechapterhead}
7849   {#1}
7850   {\print@leftmargin@eledsection%
7851     #1%
7852     \print@rightmargin@eledsection%
7853   }
7854   {}
7855   {}
7856   {}
7857
7858 \patchcmd{\@makechapterhead}% For BIDI
7859   {\if@RTL\raggedleft\else\raggedright\fi}%
7860   {\if@eled@sectioning\else%
7861     \if@RTL\raggedleft\else\raggedright\fi%
7862     \fi%
7863   }%
7864   {}%
7865   {}%
7866
7867 \patchcmd{\@makeschapterhead}
7868   {#1}
7869   {\print@leftmargin@eledsection%
7870     #1%
7871     \print@rightmargin@eledsection%
7872   }
7873   {}
7874   {}
7875
7876 \pretocmd{\@sect}
7877   {\let\old@edtext=\edtext
7878     \let\edtext=\dummy@edtext@showlemma%
7879   }
7880   {}
7881   {}
7882
7883 \apptocmd{\@sect}
7884   {\let\edtext=\old@edtext}
7885   {}
7886   {}
7887
7888 \pretocmd{\@ssect}
7889   {\let\old@edtext=\edtext%
7890     \let\edtext=\dummy@edtext@showlemma%
7891   }
7892   {}
7893   {}
7894
7895 \apptocmd{\@ssect}
7896   {\let\edtext=\old@edtext}
7897   {}

```

```

7898 {}
7899
7900 %

```

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.

```

7901 \@ifpackageloaded{nameref}{
7902
7903   \patchcmd{\NR@sect}
7904     {#8}
7905     {#8%
7906       \print@rightmargin@eledsection%
7907     }
7908     {}
7909     {}
7910
7911   \patchcmd{\NR@sect}
7912     {\hskip #3\relax}
7913     {\hskip #3\relax%
7914       \print@leftmargin@eledsection%
7915     }
7916     {}
7917     {}
7918
7919   \patchcmd{\NR@ssect}
7920     {#5}
7921     {#5%
7922       \print@rightmargin@eledsection%
7923     }
7924     {}
7925     {}
7926
7927   \patchcmd{\NR@ssect}
7928     {\hskip #1}
7929     {\hskip #1%
7930       \print@leftmargin@eledsection%
7931     }
7932     {}
7933     {}
7934   }%
7935   {
7936     \patchcmd{\@sect}
7937       {#8}
7938       {#8%
7939         \print@rightmargin@eledsection%
7940       }
7941       {}
7942       {}
7943
7944     \patchcmd{\@sect}

```

```

7945     {\hskip #3\relax}
7946     {\hskip #3\relax%
7947     \print@leftmargin@eledsection%
7948     }
7949     {}
7950     {}
7951
7952     \patchcmd{\@ssect}
7953     {#5}
7954     {#5%
7955     \print@rightmargin@eledsection%
7956     }
7957     {}
7958     {}
7959
7960     \patchcmd{\@ssect}
7961     {\hskip #1}
7962     {\hskip #1%
7963     \print@leftmargin@eledsection%
7964     }
7965     {}
7966     {}
7967     }%
7968 }
7969 %

```

Now, we have finished to patch the commands, using # with a catcode equals to 12. We close the `\notbool{@noeled@sec}` statement, restore the normal catcode for # and reopen a new `\notbool{@noeled@sec}` statement.

```

7970 {}}%
7971 \protect\catcode`\#=6 %Space NEEDS by \catcode
7972 \notbool{@noeled@sec}{%
7973 %

```

XXXII.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.

```

7974 \newwrite\eled@sectioning@out
7975 %

```

`\eledchapter` `\eledchapter` And now, the user sectioning commands, which write to the file, and also add content as a “normal” line.

```

\eledsection
\eledsubsection
\eledsubsubsection
7976 \newcommand{\eledchapter}[2][]{%
7977   #2%
\eledchapter*
7978   \ifledRcol%
\eledsection*
7979   \immediate\write\eled@sectioningR@out{%
\eledsubsection*
\eledsubsubsection*

```

```

7980     \string\eled@chapter{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR}{R}
7981   }%
7982   \else%
7983     \immediate\write\eled@sectioning@out{%
7984       \string\eled@chapter{#1}{\unexpanded{#2}}{\the\l@dnumpstartsL}{R}
7985     }%
7986   \fi%
7987 }
7988
7989 \newcommand{\eledsection}[2][{}]{%
7990   #2%
7991   \ifledRcol%
7992     \immediate\write\eled@sectioningR@out{%
7993       \string\eled@section{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR}{R}
7994     }%
7995   \else%
7996     \immediate\write\eled@sectioning@out{%
7997       \string\eled@section{#1}{\unexpanded{#2}}{\the\l@dnumpstartsL}{R}
7998     }%
7999   \fi%
8000 }
8001
8002 \newcommand{\eledsubsection}[2][{}]{%
8003   #2%
8004   \ifledRcol%
8005     \immediate\write\eled@sectioningR@out{%
8006       \string\eled@subsection{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR}{R}
8007     }%
8008   \else%
8009     \immediate\write\eled@sectioning@out{%
8010       \string\eled@subsection{#1}{\unexpanded{#2}}{\the\l@dnumpstartsL}{R}
8011     }%
8012   \fi%
8013 }
8014 \newcommand{\eledsubsubsection}[2][{}]{%
8015   #2%
8016   \ifledRcol%
8017     \immediate\write\eled@sectioningR@out{%
8018       \string\eled@subsubsection{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR}
8019     }%
8020   \else%
8021     \immediate\write\eled@sectioning@out{%
8022       \string\eled@subsubsection{#1}{\unexpanded{#2}}{\the\l@dnumpstartsL}
8023     }%
8024   \fi%
8025 }
8026

```

```

8027 \WithSuffix\newcommand\eledchapter*[2] [] {%
8028   #2%
8029   \ifledRcol%
8030     \immediate\write\eled@sectioningR@out{%
8031       \string\eled@chapter{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR}{*}{R}
8032     }%
8033   \else%
8034     \immediate\write\eled@sectioning@out{%
8035       \string\eled@chapter{#1}{\unexpanded{#2}}{\the\l@dnumpstartsL}{*}{L}
8036     }%
8037   \fi%
8038 }
8039
8040
8041 \WithSuffix\newcommand\eledsection*[2] [] {%
8042   #2%
8043   \ifledRcol%
8044     \immediate\write\eled@sectioningR@out{%
8045       \string\eled@section{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR}{*}{R}
8046     }%
8047   \else%
8048     \immediate\write\eled@sectioning@out{%
8049       \string\eled@section{#1}{\unexpanded{#2}}{\the\l@dnumpstartsL}{*}{L}
8050     }%
8051   \fi%
8052 }
8053
8054 \WithSuffix\newcommand\eledsubsection*[2] [] {%
8055   #2%
8056   \ifledRcol%
8057     \immediate\write\eled@sectioningR@out{%
8058       \string\eled@subsection{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR}{*}{R}
8059     }%
8060   \else%
8061     \immediate\write\eled@sectioning@out{%
8062       \string\eled@subsection{#1}{\unexpanded{#2}}{\the\l@dnumpstartsL}
8063     }{*}{L}
8064   \fi%
8065 }
8066
8067 \WithSuffix\newcommand\eledsubsubsection*[2] [] {%
8068   #2%
8069   \ifledRcol%
8070     \immediate\write\eled@sectioningR@out{%
8071       \string\eled@subsubsection{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR}
8072     }{*}{R}
8073   \else%

```

```

8074 \immediate\write\eled@sectioning@out{%
8075 \string\eled@subsubsection{#1}{\unexpanded{#2}}{\the\l@dumpstartsL
} *}{}
8076 }%
8077 \fi%
8078 }
8079 %

```

XXXII.7 Macros written in the auxiliary file

- `\eled@chapter` The sectioning macros, called in the auxiliary file. They have five arguments:
- `\eled@section`
- `\eled@subsection`
- `\eled@subsubsection`
1. Optional arguments of \LaTeX sectioning command.
 2. Mandatory arguments of \LaTeX sectioning command.
 3. Pstart number.
 4. Side: R if right, nothing if left.
 5. Starred or not.

```

8080 \def\eled@chapter#1#2#3#4#5{%
8081 \ifstrempy{#4}%
8082 {%
8083 \ifstrempy{#1}%
8084 {%
8085 \global\csdef{eled@sectioning@#3#5}{\let\edtext=\
dummy@edtext@showlemma\chapter{#2}}%
8086 \global\csdef{eled@sectmark@#3#5}{\let\edtext=\dummy@edtext{}\
chaptermark{#2}}%
8087 }%Need for \pairs, because of using parbox.
8088 {%
8089 \global\csdef{eled@sectioning@#3#5}{\let\edtext=\
dummy@edtext@showlemma\chapter[#1]{#2}}%
8090 \global\csdef{eled@sectmark@#3#5}{\let\edtext=\dummy@edtext{}\
chaptermark{#2}}%Need for \pairs, because of using parbox.
8091 }%
8092 }%
8093 {%
8094 \ifstrempy{#1}%
8095 {\global\csdef{eled@sectioning@#3#5}{\let\edtext=\
dummy@edtext@showlemma\chapter*{#2}}}%
8096 {\global\csdef{eled@sectioning@#3#5}{\let\edtext=\
dummy@edtext@showlemma\chapter*{#1}{#2}}}%Bug in LaTeX!
8097 }%
8098 \listcsgadd{eled@sections#5@@}{#3}%
8099 }
8100 \def\eled@section#1#2#3#4#5{%
8101 \ifstrempy{#4}%

```



```

8102     {\ifstrempy{#1}%
8103     {%
8104     \global\csdef{eled@sectioning@#3#5}{\section{#2}}%
8105     \global\csdef{eled@sectmark@#3#5}{\let\edtext=\dummy@edtext}\
sectionmark{#2}}%Need for \pairs, because of using parbox.
8106     }%
8107     {%
8108     \global\csdef{eled@sectioning@#3#5}{\section[#1]{#2}}%
8109     \global\csdef{eled@sectmark@#3#5}{\let\edtext=\dummy@edtext}\
sectionmark{#1}}%Need for \pairs, because of using parbox.
8110     }%
8111     }%
8112     {\ifstrempy{#1}%
8113     {\global\csdef{eled@sectioning@#3#5}{\section*{#2}}}%
8114     {\global\csdef{eled@sectioning@#3#5}{\section*{#1}{#2}}}%Bug in
LaTeX!
8115     }
8116     \listcsadd{eled@sections#5@@}{#3}%
8117     }
8118 \def\eled@subsection#1#2#3#4#5{%
8119     \ifstrempy{#4}%
8120     {\ifstrempy{#1}%
8121     {%
8122     \global\csdef{eled@sectioning@#3#5}{\subsection{#2}}%
8123     \global\csdef{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)
8124     }%
8125     {%
8126     \global\csdef{eled@sectioning@#3#5}{\subsection[#1]{#2}}%
8127     \global\csdef{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)
8128     }%
8129     }%
8130     {\ifstrempy{#1}%
8131     {\global\csdef{eled@sectioning@#3#5}{\subsection*{#2}}}%
8132     {\global\csdef{eled@sectioning@#3#5}{\subsection*{#1}{#2}}}%Bug in
LaTeX!
8133     }
8134     \listcsadd{eled@sections#5@@}{#3}%
8135     }
8136 \def\eled@subsubsection#1#2#3#4#5{%
8137     \ifstrempy{#4}%
8138     {\ifstrempy{#1}%
8139     {\global\csdef{eled@sectioning@#3#5}{\subsubsection{#2}}}%
8140     {\global\csdef{eled@sectioning@#3#5}{\subsubsection[#1]{#2}}}%
8141     }%
8142     {\ifstrempy{#1}%
8143     {\global\csdef{eled@sectioning@#3#5}{\subsubsection*{#2}}}%

```

```

8144     {\global\csdef{eled@sectioning@#3#5}{\subsubsection*{#1}{#2}}}%Bug
in LaTeX!
8145     }
8146     \listcsadd{eled@sections#5@0}{#3}%
8147     }
8148
8149 %

```

End of the conditional test about noeledsec option.

```

8150 }{}
8151 %

```

XXXIII 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.

```

8152 \def\normal@page@break{}
8153 %

```

\l@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.

```

8154 \def\l@prev@pb{}
8155 \def\l@prev@nopb{}
8156 %

```

\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.

```

8157 \newcommand{\ledpb}{\write\linenum@out{\string\led@pb}}
8158 \newcommand{\ledpbnum}[1]{\write\linenum@out{\string\led@pbnum{#1}}}
8159 \newcommand{\lednopb}{\write\linenum@out{\string\led@nopb}}
8160 \newcommand{\lednopbnum}[1]{\write\linenum@out{\string\led@nopbnum{#1}}}
8161 %

```

`\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.

```

8162 \newcommand{\led@pb}{\listxadd{\l@prev@pb}{\the\absline@num}}
8163 \newcommand{\led@pbnum}[1]{\listxadd{\l@prev@pb}{#1}}
8164 \newcommand{\led@nopb}{\listxadd{\l@prev@nopb}{\the\absline@num}}
8165 \newcommand{\led@nopbnum}[1]{\listxadd{\l@prev@nopb}{#1}}
8166 %

```

`\ledpbsetting` The `\ledpbsetting` macro only changes the value of `\led@pb@macro`, for which the default value is before.

```

8167 \def\led@pb@setting{before}
8168 \newcommand{\ledpbsetting}[1]{\gdef\led@pb@setting{#1}}
8169 %

```

`\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`.

```

8170 \newcommand{\led@check@pb}{\xifinlist{\the\absline@num}{\l@prev@pb}{\pagebreak[4]}}
8171 \newcommand{\led@check@nopb}{%
8172   \IfStrEq{\led@pb@setting}{before}{%
8173     \xifinlist{\the\absline@num}{\l@prev@nopb}{%
8174       {\numdef{\abs@prevline}{\the\absline@num-1}%
8175       \xifinlist{\abs@prevline}{\normal@page@break}%
8176       {\nopagebreak[4]\enlargethispage{\baselineskip}}}%
8177     }%
8178   }%
8179 }%
8180 }%
8181 \IfStrEq{\led@pb@setting}{after}{%
8182   \xifinlist{\the\absline@num}{\l@prev@nopb}{%
8183     \xifinlist{\the\absline@num}{\normal@page@break}%
8184     {\nopagebreak[4]\enlargethispage{\baselineskip}}}%
8185   }%
8186 }%
8187 }%
8188 }%
8189 }%
8190 }
8191 %

```

XXXIV 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.

```

8192 \newcommand{\check@pb@in@verse}{%
8193   \ifinstanza\iflednopbinverse\ifinserthangingsymbol% Using stanzas and
enabling page breaks in verse control, while on a hanging verse.
8194   \ifnum\page@num=\last@page@num\else%If we have change page
8195     \IfStrEq{\led@pb@setting}{before}{%
8196       \numgdef{\abs@line@verse}{\the\absline@num-1}%
8197       \ledpbnum{\abs@line@verse}%
8198     }{}%
8199     \IfStrEq{\led@pb@setting}{after}{%
8200       \numgdef{\abs@line@verse}{\the\absline@num-1}%
8201       \lednopbnum{\abs@line@verse}%
8202     }{}%
8203   \fi%
8204 \fi\fi\fi%
8205 }
8206 %

```

XXXV 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.³³

However, this command disrupts the line breaking mechanism when it is called before any word. This is why `reledmac` defines `\Hy@raisedlink@left` that is called to the left of words, at the beginning of `\edtext` or inside the `\edlabel` commands.³⁴

```

8207 \def\Hy@raisedlink@left#1{%
8208   \ifvmode
8209     #1%
8210   \else

```

³³<http://tex.stackexchange.com/a/17138/7712>.

³⁴The 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>.

```

8211 \Hy@SaveSpaceFactor
8212 \llap{\smash{%
8213 \begingroup
8214 \let\HyperRaiseLinkLength\@tempdima
8215 \setlength\HyperRaiseLinkLength\HyperRaiseLinkDefault
8216 \HyperRaiseLinkHook
8217 \expandafter\endgroup
8218 \expandafter\raise\the\HyperRaiseLinkLength\hbox{%
8219 \Hy@RestoreSpaceFactor
8220 #1%
8221 \Hy@SaveSpaceFactor
8222 }%
8223 }}%
8224 \Hy@RestoreSpaceFactor
8225 \penalty\@M\hskip\z@\relax
8226 \fi
8227 }
8228 %

```

XXXVI Compatibility with eledmac

Here, we define some commands for the eledmac-compat option.

```

8229 \ifeledmaccompat{%
8230
8231 \newcommand{\footnormalX}[1]{\arrangementX[#1]{normal}}%
8232 \newcommand{\footparagraphX}[1]{\arrangementX[#1]{paragraph}}%
8233 \newcommand{\foottwocolX}[1]{\arrangementX[#1]{twocol}}%
8234 \newcommand{\footthreecolX}[1]{\XarrangementX[#1]{threecol}}%
8235
8236 \unless\ifnocritical@
8237 \newcommand{\footnormal}[1]{\Xarrangement[#1]{normal}}%
8238 \newcommand{\footparagraph}[1]{\Xarrangement[#1]{paragraph}}%
8239 \newcommand{\foottwocol}[1]{\Xarrangement[#1]{twocol}}%
8240 \newcommand{\footthreecol}[1]{\Xarrangement[#1]{threecol}}%
8241 \let\hsizetwocol\Xhsizetwocol
8242 \let\hsizethreecol\Xsizethreecol
8243 \let\bhookXnote\Xbhooknote
8244 \let\boxsymlinenum\Xboxsymlinenum
8245 \let\symlinenum\Xsymlinenum
8246 \let\beforenumberinfootnote\Xbeforenumber
8247 \let\afternumberinfootnote\Xafternumber
8248 \let\beforeXsymlinenum\Xbeforesymlinenum
8249 \let\afterXsymlinenum\Xaftersymlinenum
8250 \let\inplaceofnumber\Xinplaceofnumber
8251 \let\Xlemmaseparator\lemmaseparator
8252 \let\afterlemmaseparator\Xafterlemmaseparator
8253 \let\beforelemmaseparator\Xbeforelemmaseparator
8254 \let\inplaceoflemmaseparator\Xinplaceoflemmaseparator

```

```

8255 \let\txtbeforeXnotes\Xtxtbeforenotes
8256 \let\afterXrule\Xafterrule
8257 \let\numberonlyfirstinline\Xnumberonlyfirstinline
8258 \let\numberonlyfirstintwlines\Xnumberonlyfirstintwlines
8259 \let\nonumberinfootnote\Xnonumberinfootnote
8260 \let\pstartinfootnote\Xpstart
8261 \let\pstartinfootnoteeverytime\Xpstarteverytime
8262 \let\onlyXpstart\Xonlypstart
8263 \let\Xnonumberinfootnote\Xnonumber
8264 \let\Xnonbreakableafternumber\Xnonbreakableafternumber
8265 \let\maxhXnotes\Xmaxhnotes
8266 \let\beforeXnotes\Xbeforenotes
8267 \let\boxlinenum\Xboxlinenum
8268 \let\boxlinenumalign\Xboxlinenumalign
8269 \let\boxstartlinenum\Xboxstartlinenum
8270 \let\boxendlinenum\Xboxendlinenum
8271 \let\twlines\Xtwlines
8272 \let\morethantwlines\Xmorethantwlines
8273 \let\twlinesbutnotmore\Xtwlinesbutnotmore
8274 \let\twlinesonlyinsamepage\Xtwlinesonlyinsamepage
8275 \fi
8276
8277 \unless\ifnofamiliar@
8278   \let\notesXwidthliketwocolumns\noteswidthliketwocolumnsX
8279 \fi
8280 \newcommandx{\parafootsep}[2][1,usedefault]{%
8281   \Xparafootsep[#1]{#2}%
8282   \parafootsepX[#1]{#2}
8283 }%
8284
8285 \newcommandx{\afternote}[2][1,usedefault]{%
8286   \Xafternote[#1]{#2}%
8287   \afternoteX[#1]{#2}%
8288 }%
8289
8290 \unless\ifnoend@
8291   \let\XendXtwlines\Xendtwlines
8292   \let\XendXmorethantwlines\Xendmorethantwlines
8293   \let\XhookXendnote\Xendhooknote
8294   \let\XboxXendlinenum\Xendboxlinenum%
8295   \let\XboxXendlinenumalign\Xendboxlinenumalign%
8296   \let\XboxXendstartlinenum\Xendboxstartlinenum%
8297   \let\XboxXendendlinenum\Xendboxendlinenum%
8298   \let\XendXlemmaseparator\Xendlemmaseparator
8299   \let\XendXbeforelemmaseparator\Xendbeforelemmaseparator
8300   \let\XendXafterlemmaseparator\Xendafterlemmaseparator
8301   \let\XendXinplaceoflemmaseparator\Xendinplaceoflemmaseparator
8302 \fi
8303
8304 \AtBeginDocument{%

```

```
8305 \ifdef\lineref{}\let\lineref\edlineref}%  
8306 }%  
8307  
8308  
8309 \fi%  
8310 %
```

</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 changed³⁵ 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>	<u>1–2 I saw my friend</u>
<code>July 16, 1954.}</code>	Smith on Tuesday.] The
<code>/</code>	date was July 16, 1954.

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

The second argument of the `\critext` macro, `⟨commands⟩`, is the same as the second argument to the `\edtext` macro.

It is possible to define aliases for `\critext`, which can be easier to type. You can make a single character substitute for `\critext` by saying this:

```
\catcode`\<=\active
```

³⁵A 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. 112 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 6.6.1 p. 31), 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. 32) or `\Xfootnote` options (6.2.2 p. 23). Otherwise please add a new ticket on Github to request a new function for doing this.³⁶

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]
```

³⁶<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 `stanzaindentrepitition` (cf. 9.3 p. 45). This bug had two consequences:

1. `stanzaindentrepitition` did not work when its value was greater than 2.
2. `stanzaindentrepitition` worked wrong when its value was equal to 2.

So, if you used `stanzaindentrepitition` with a value equal to 2, you had to change your `\setstanzaindent`s. Explanation:

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

to:

```
\setcounter{stanzaindentrepitition}{2}
\setstanzaindent{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. 17). If so, add a `\relax` between `\pstart`/`\pend` and the first bracket.

The version 1.12.0 also introduce a better way to handle sectional divisions inside numbered text. Please read 16.2 p. 60.

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. 63 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>\interparanoteglue</code>	<code>\Xafternote</code> and <code>\afternoteX</code>
<code>\ledchapter</code>	<code>\eledchapter</code>
<code>\ledsection</code>	<code>\eledsection</code>
<code>\ledsetnormalparstuff</code>	<code>\Xledsetnormalparstuff</code> and <code>\ledsetnormalparstuffX</code>
<code>\ledsubsection</code>	<code>\eledsubsection</code>
<code>\ledsubsubsection</code>	<code>\eledsubsubsection</code>
<code>\noeledsec</code>	Package option <code>noeledsec</code>
<code>\noendnotes</code>	Package option <code>noendnotes</code>
<code>\pageparbreak</code>	<code>\ledpb</code>

The `ledsecnolinenumber` option has been removed, because it was related to deprecated commands.

The `oldprintnpnumspace` option has been removed too, because it was related to a historical bug. The `\usingedtext` and `\usingcritext` commands are also deprecated.

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 \renewcommand</i>	<i>Replaced with</i>
\@led@extranofeet	\newseries
\apprefprefixmore	\setapprefprefixmore
\apprefprefixsingle	\setapprefprefixsingle
\endstanzaextra	Optional argument of \&
\hangingsymbol	\sethangingsymbol
\ledfootinsdim	\Xmaxhnotes and \maxhnotesX
\parafootftmsep	\Xparafootsep and \parafootsepX
\notenumfont	\Xnotenumfont, \Xendnotenumfont and \notenumfontX
\notefontsetup	\Xnotefontsize, \Xendnotefontsize and \notefontsizeX
\sidenoteseq	\setsidenotsep
\startstanzahook	Optional argument of \stanza
\symplinenum	\Xsymplinenum

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>
\footparagraph	\Xarrangement
\footnormal	\Xarrangement
\foottwocol	\Xarrangement
\footthreecol	\Xarrangement
\footparagraphX	\arrangementX
\footnormalX	\arrangementX
\foottwocolX	\arrangementX
\footthreecolX	\arrangementX
\afterlemmaseparator	\Xafterlemmaseparator
\afternote	\Xafternote and \afternoteX
\afternumberinfootnote	\Xafternumber
\afterXrule	\Xafterrule
\afterXsymplinenum	\Xaftersymplinenum
\beforelemmaseparator	\Xbeforelemmaseparator
\beforenumberinfootnote	\Xbeforenumber
\beforeXnotes	\Xbeforenotes
\beforeXsymplinenum	\Xbeforesymplinenum

<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}{\langle...\rangle}` to the use of `\Xlinangeseparator` or `/` and `\Xendlinangeseparator` (7.2.3 p. 33).

Appendix A.14 Migration to reledmac 2.5.0

It is strongly recommended to stop redefining `\printnpnum` and to use the hooks documented in 7.3 p. 37.

`\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`.

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 `edstanzan.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>	44
<code>\@EDROWFILL@</code>	1
<code>\@adv</code>	1
<code>\@advancestanzanumber</code>	1
<code>\@doclearpage</code>	1
<code>\@doreinfeetX</code>	1
<code>\@edindex@hyperref</code>	1
<code>\@edrowfill@</code>	1
<code>\@edtext@level</code>	1
<code>\@emptytoks</code>	1
<code>\@endmsd</code>	1

\@fnpos	1
\@footnotemark	1
\@footnotetext	1
\@getfirstseries	1
\@gobblefive	1
\@gobblefour	1
\@gobblethree	1
\@h	1
\@hangingsymbol	1
\@iiiminipage	1
\@insertstanzaumber	1
\@k	1
\@l@tempcnta	1
\@l@tempcntb	1
\@lab	1
\@led@testifnofoot	1
\@lemma	1
\@line@num	1
\@lock	1
\@lopL	1
\@lopR	1
\@makechapterhead	1
\@makeschapterhead	1
\@mem@extranofeet	1
\@mem@old@ssect	1
\@mpfnpos	1
\@msd	1
\@msd@c	1
\@msd@options@iffullpage	1
\@msdata@list	1
\@nl	1
\@nl@reg	1
\@opXfeet	1
\@pend	1
\@pendR	1
\@ref	1
\@ref@reg	1
\@sect	1
\@series	1
\@set	1
\@sidenotesep	1
\@ssect	1
\@startstanza	1
\@stopstanza	1
\@sw	1
\@tag	1
\@wredindex	1
\@xloop	1
\@xympar	1
CLASSarticle	63

CLASSbook	63, 313
CLASSmemoir	180, 181, 235, 236, 266, 313, 332, 387, 391
CLASSscrbook	391
COMMAND*footnote	64
COMMAND\...\@footnotemark...	184
COMMAND\...d@ta	136
COMMAND\<hook	
@<series	224
COMMAND\<hookname	
<pseudoseries	226, 227
COMMAND\<type	
footfmt	172
COMMAND\@@line	164
COMMAND\@MM	152, 388
COMMAND\@Rlineflag	267, 388
COMMAND\@SErefprefix	247
COMMAND\@SErefprefixmore	247
COMMAND\@add@	302
COMMAND\@adv	99
COMMAND\@apprefprefixmore	247
COMMAND\@apprefprefixsingle	247
COMMAND\@bsphack	238
COMMAND\@doclearpage	236, 381, 391
COMMAND\@doreinfeetX	391
COMMAND\@dprintingcolumns	388
COMMAND\@edindex@hyperref	268, 269
COMMAND\@edtext@level	116
COMMAND\@esphack	238
COMMAND\@fnpos	202
COMMAND\@footnotemark	181, 381, 391
COMMAND\@footnotetext	181, 182, 381
COMMAND\@gobble	114
COMMAND\@gobblefive	221, 390
COMMAND\@gobblefour	387
COMMAND\@gobblethree	380
COMMAND\@h	167
COMMAND\@hangingsymbol	272
COMMAND\@iiiminipage	258, 259, 380, 391
COMMAND\@iiiminipage	258
COMMAND\@l	386
COMMAND\@l@dtmpcnta	138, 140, 146
COMMAND\@l@dtmpcntb	140
COMMAND\@l@reg	386
COMMAND\@lab	96, 238, 240, 244, 380
COMMAND\@ldunboxmpfoot	260
COMMAND\@led@extranofeet	333
COMMAND\@ledinnote@command	264, 265
COMMAND\@lemma	118, 120
COMMAND\@lock	90, 272
COMMAND\@lopL	381

COMMAND\@lopR	381
COMMAND\@makecol	232, 233, 235, 391
COMMAND\@mpfnpos	202
COMMAND\@msd	280, 281
COMMAND\@msd@c	280
COMMAND\@msd@options@iffullpage	286
COMMAND\@msdata@list	281, 282
COMMAND\@nl	96–99, 101, 109, 240, 380, 381
COMMAND\@nl@reg	97, 330, 381, 386
COMMAND\@opXfeet	381
COMMAND\@opfeetX	391
COMMAND\@opextrafeeti	391
COMMAND\@page	98, 240
COMMAND\@pend	381
COMMAND\@pendR	381
COMMAND\@ref	96, 105, 106, 109, 114
COMMAND\@ref@reg	105, 381
COMMAND\@reinserts	232, 234, 235, 391
COMMAND\@secondoftwo	65
COMMAND\@sect	316
COMMAND\@series	223
COMMAND\@set	100
COMMAND\@sidenotesep	257
COMMAND\@stopmsd	281
COMMAND\@sw	106, 122, 124, 125
COMMAND\@tag	115, 117, 120
COMMAND\@tempcnta	76
COMMAND\@tempcntb	76
COMMAND\@toksa	82
COMMAND\@toksb	82
COMMAND\@xloop	147, 148
COMMAND\@xympar	251, 391
COMMAND\Aendnote	15, 24
COMMAND\Afootfmt	151
COMMAND\Afootgroup	151
COMMAND\Afootnote	8, 15, 22–24, 26, 117, 159, 180, 203, 215, 390
COMMAND\Afootstart	151
COMMAND\AtEveryPend	17, 18, 47, 131, 388, 389, 391
COMMAND\AtEveryPstart	17, 18, 47, 388, 389, 391, 394
COMMAND\AtEveryStanza	46, 395
COMMAND\AtEveryStopStanza	47, 395
COMMAND\Bendnote	15, 22
COMMAND\Bfootnote	8, 15, 180, 203, 215
COMMAND\Centering	40
COMMAND\Cfootnote	180
COMMAND\Columns	77, 155
COMMAND\Dfootnote	180
COMMAND\Efootnote	180
COMMAND\Gls	56
COMMAND\Hy@raisedlink	324

COMMAND\Hy@raisedlink@left	324
COMMAND\LTR	40
COMMAND\NR	316
COMMAND\Pages	77, 233, 234
COMMAND\ProcessOptionsX	68
COMMAND\RL	39
COMMAND\RaggedLeft	40
COMMAND\RaggedRight	40
COMMAND\SEonlypage	245, 393
COMMAND\Seref	50–52, 245, 248, 394
COMMAND\Serefonlypage	50–52, 336, 393
COMMAND\Serefwithpage	50, 52, 245, 248, 336, 393
COMMAND\Stanza	386
COMMAND\Waklam	303
COMMAND\X@doreinfeet	234, 391
COMMAND\XR@prefix	251
COMMAND\XR@test	250, 251
COMMAND\XR@test@mac	251
COMMAND\XR@test@mac@test	251
COMMAND\XXXXXXfmt	329
COMMAND\XXXXXXfmt	329
COMMAND\Xafterlemmaseparator	37, 333
COMMAND\Xafternote	41, 332, 333
COMMAND\Xafternumber	35, 333
COMMAND\Xafterrule	42, 203, 333, 386, 389
COMMAND\Xaftersymlinenum	35, 333
COMMAND\Xarrangement	32, 64, 152, 153, 225, 333
COMMAND\Xarrangement@footparagraph	157
COMMAND\Xarrangement@normal	153
COMMAND\Xarrangement@paragraph	157
COMMAND\Xbeforeinserting	40
COMMAND\Xbeforelemmaseparator	37, 333
COMMAND\Xbeforenotes	42, 202, 333, 386, 389
COMMAND\Xbeforenumber	33, 35, 333
COMMAND\Xbeforesymlinenum	35, 333
COMMAND\Xbhookendnote	334
COMMAND\Xbhookgroup	41, 393, 394
COMMAND\Xbhooknote	40, 334, 335, 391, 392
COMMAND\Xboxendlinenum	36, 334, 390
COMMAND\Xboxlinenum	36, 334
COMMAND\Xboxlinenumalign	36, 334, 390
COMMAND\Xboxstartlinenum	36, 334, 390
COMMAND\Xboxsymlinenum	36, 334
COMMAND\Xcolalign	40, 389
COMMAND\Xdo@feet	233, 381, 391
COMMAND\Xend	221
COMMAND\XendXafterlemmaseparator	334
COMMAND\XendXbeforelemmaseparator	334
COMMAND\XendXinplaceoflemmaseparator	334
COMMAND\XendXlemmaseparator	334

COMMAND\XendXmorethantwolines	334
COMMAND\XendXtwolines	334
COMMAND\Xendafterenumber	35, 392
COMMAND\Xendafterlemmaseparator	38, 334
COMMAND\Xendafternote	43, 394
COMMAND\Xendafternumber	37
COMMAND\Xendafterpagenumber	37, 52
COMMAND\XendafterpagenumberSErefonlypage	336
COMMAND\Xendaftersymlinenum	35, 37, 392
COMMAND\Xendahookinplaceofnumber	37, 392
COMMAND\Xendahooklinenum	37, 392
COMMAND\Xendbeforelemmaseparator	38, 334
COMMAND\Xendbeforelinenum	37
COMMAND\Xendbeforenumber	35, 392
COMMAND\Xendbeforepagenumber	37, 51, 52
COMMAND\XendbeforepagenumberSErefonlypage	51
COMMAND\Xendbeforesymlinenum	35, 37, 392
COMMAND\Xendbhookinplaceofnumber	37, 392
COMMAND\Xendbhooklinenum	37, 392
COMMAND\Xendbhooknote	40
COMMAND\Xendboxendlinum	36, 390
COMMAND\Xendboxlinenum	36, 334, 388
COMMAND\Xendboxlinenumalign	36, 334, 390
COMMAND\Xendboxstartlinenum	36, 390
COMMAND\Xendboxsymlinenum	36, 392
COMMAND\Xendhangindent	39, 392, 394
COMMAND\Xendinplaceoflemmaseparator	24, 38, 334
COMMAND\Xendinplaceofnumber	36, 391
COMMAND\Xendinsertsep@	208
COMMAND\Xendlemmadisablefontselection	38
COMMAND\Xendlemmafont	38, 393
COMMAND\Xendlemmaseparator	25, 38, 334
COMMAND\Xendletboxendlinum	334
COMMAND\Xendlineflag	52
COMMAND\Xendlineprefixmore	37, 52
COMMAND\Xendlineprefixsingle	37, 52
COMMAND\Xendlinerangeseparator	33, 52, 336, 392
COMMAND\Xendmorethantwolines	24, 34, 52, 334, 389, 390
COMMAND\Xendnonumber	34, 391
COMMAND\Xendnote	206, 220, 221, 389
COMMAND\Xendnotefontsize	39, 333
COMMAND\Xendnotenumfont	37, 38, 333
COMMAND\Xendnumberonlyfirstinline	33, 392
COMMAND\Xendnumberonlyfirstintwolines	33, 392
COMMAND\Xendparagraph	43, 386
COMMAND\Xendsep	43
COMMAND\Xendsublinesep	35, 52
COMMAND\Xendsymlinenum	33, 392
COMMAND\Xendtwolines	24, 34, 52, 334, 389, 390
COMMAND\Xendtwolinesbutnotmore	34, 52, 389, 390

COMMAND\Xendtwolinesonlyinsamepage	34, 52, 389, 390
COMMAND\Xendwrapcontent	39, 395
COMMAND\Xendwraplemma	39, 395
COMMAND\Xfootfmt	335
COMMAND\Xfootgroup	156
COMMAND\Xfootins	155
COMMAND\Xfootnote	49, 55, 115, 329, 383, 387, 389, 393
COMMAND\Xfootstarts	156
COMMAND\Xhangindent	39, 392
COMMAND\Xhsize	336, 393, 394
COMMAND\Xhsizethreecol	41, 43, 334
COMMAND\Xhsizetwocol	41, 43, 226, 334
COMMAND\Xinplaceoflemmaseparator	24, 37, 334
COMMAND\Xinplaceofnumber	36, 334, 389, 390
COMMAND\Xinsertparafootsep	162, 163
COMMAND\Xledsetnormalparstuff	331, 332, 389
COMMAND\Xlemmadisablefontselection	38
COMMAND\Xlemmafont	38, 393
COMMAND\Xlemmaseparator	37, 227, 230, 232, 334
COMMAND\Xlineflag	51
COMMAND\Xlinangeseparator	33, 52, 336, 392
COMMAND\Xmaxhnotes	42, 63, 64, 333, 334, 386, 388
COMMAND\Xmorethantwolines	23, 34, 52, 334, 388
COMMAND\Xnoindent	392
COMMAND\Xnolemmaseparator	37, 232, 334
COMMAND\Xnonbreakableafternumber	35, 334, 384
COMMAND\Xnonumber	34, 334
COMMAND\Xnonumberinfootnote	334
COMMAND\Xnotefontsize	39, 333
COMMAND\Xnotefontsize@⟨s⟩	162, 166, 167
COMMAND\Xnotenumfont	38, 333
COMMAND\Xnoteswidthliketwocolumns	43, 387
COMMAND\Xnumberonlyfirstinline	33, 93, 226, 227, 230, 334, 383, 388
COMMAND\Xnumberonlyfirstintwolines	33, 334, 383
COMMAND\Xonlypstart	35, 334, 383, 388
COMMAND\Xparafootsep	41, 93, 333, 334, 395
COMMAND\Xparafootsep@series	162
COMMAND\Xparinden	336
COMMAND\Xparindent	39, 389, 392, 394
COMMAND\Xpstart	34, 35, 331, 334, 383, 388
COMMAND\Xpstarteverytime	34, 331, 334, 388
COMMAND\Xragged	41
COMMAND\Xrigidbalance	164, 336, 393
COMMAND\Xstanza	35, 47
COMMAND\Xstanzaseparator	35
COMMAND\Xsublinesep	21, 35, 52, 336
COMMAND\Xsublinesepside	21, 35
COMMAND\Xsymlinenum	33, 41, 333, 334, 390
COMMAND\Xtwolines	23, 34, 52, 177, 226, 334, 388
COMMAND\Xtwolinesappref	226

COMMAND\Xtwolinesbutnotmore	34, 52, 334, 389
COMMAND\Xtwolinesbutnotmoreappref	227
COMMAND\Xtwolinesonlyinsamepage	34, 52, 334, 389
COMMAND\Xtxtbeforenotes	41, 334, 395
COMMAND\Xunvxh	159, 335
COMMAND\Xwidth	43, 336, 394
COMMAND\Xwrapcontent	39, 395
COMMAND\Xwraplemma	39, 40, 395
COMMAND\&	333
COMMAND\absline@num	90, 137
COMMAND\accent	114
COMMAND\actionlines@list	91, 139
COMMAND\actions@list	91
COMMAND\add@inserts	91, 145
COMMAND\add@inserts@next	145, 146
COMMAND\add@msd@	280
COMMAND\add@msdata	281, 282
COMMAND\add@msdata@firstlineofpage	284
COMMAND\add@msddata	280
COMMAND\add@penalties	137, 146
COMMAND\addcontentsline	310
COMMAND\addfootins	329, 332
COMMAND\addfootinsX	329, 332
COMMAND\advancelabel@refs	239
COMMAND\advanceline	21, 92, 99, 111, 391
COMMAND\advancepageno	232
COMMAND\affixlin@num	257
COMMAND\affixline@num	140, 143, 144, 381
COMMAND\affixpstart@num	144
COMMAND\afterXrule	333
COMMAND\afterXsymlinenum	333
COMMAND\afterenumber	35
COMMAND\aftergroup	113, 114, 118
COMMAND\afterlemmaseparator	333
COMMAND\afternote	333
COMMAND\afternoteX	41, 332, 333
COMMAND\afternumberinfootnote	333
COMMAND\afterruleX	42, 386, 389
COMMAND\applabel	51, 241, 242, 389
COMMAND\appref	49, 51, 52, 245, 248, 393, 394
COMMAND\apprefprefixmore	51, 333
COMMAND\apprefprefixsingle	51, 333
COMMAND\apprefwithpage	51, 52, 245, 248, 390, 393
COMMAND\arrangementX	32, 64, 184, 225, 333
COMMAND\arrangementX@normal	189
COMMAND\article	14
COMMAND\at@every@pend	131
COMMAND\autopar	17, 128, 131, 132, 200, 382, 384, 385, 389
COMMAND\ballast	64
COMMAND\ballast@count	137, 146

COMMAND\baselineskip	32, 158, 162
COMMAND\beforeXnotes	333
COMMAND\beforeXsymlinenum	333
COMMAND\beforeelectedchapter	10, 61, 311
COMMAND\beforeinsertingX	40
COMMAND\beforelemmaseparator	333
COMMAND\beforenotesX	42, 335, 385, 386, 389
COMMAND\beforenumberinfootnote	333
COMMAND\begin	288
COMMAND\beginnumbering	16, 18, 77, 78, 81, 89, 94, 108, 131, 205, 280, 383, 386, 390, 391
COMMAND\bf	383
COMMAND\bfseries	38, 383
COMMAND\bhookXnote	334
COMMAND\bhookgroupX	41, 393
COMMAND\bhooknoteX	40, 335, 391, 392
COMMAND\body	273
COMMAND\bodyfootmarkA	30
COMMAND\book	14
COMMAND\boxXendlinenum	334
COMMAND\boxXendlinenumalign	334
COMMAND\boxXendstartlinenum	334
COMMAND\boxendlinenum	334
COMMAND\boxlinefootnote	174
COMMAND\boxlinenum	334
COMMAND\boxlinenumalign	334
COMMAND\boxstartlinenum	334
COMMAND\boxsymlinenum	334
COMMAND\break	33, 159
COMMAND\brokenpenalty	146
COMMAND\centering	40
COMMAND\ch@ck@l@ck	381
COMMAND\ch@cksub@l@ck	143, 381
COMMAND\chapter	60, 313, 386, 389, 391
COMMAND\chaptermark	310
COMMAND\check@pb@in@verse	324
COMMAND\colalignX	40, 389
COMMAND\collect@body	288
COMMAND\color	395
COMMAND\colorbox	65
COMMAND\columns	43
COMMAND\columnwidth	158, 387
COMMAND\command names	226
COMMAND\copyright	114
COMMAND\correct@Xfootins@box	388
COMMAND\correct@footinsX@box	388
COMMAND\count	165
COMMAND\critex	382
COMMAND\critext	120, 328, 329, 332
COMMAND\csname	69, 123, 124
COMMAND\ctab	304, 309

COMMAND\ctabtext	309
COMMAND\dcoll	297
COMMAND\def	66
COMMAND\detokenize	124
COMMAND\dimen	165
COMMAND\dimexpr	43
COMMAND\discretionary	159
COMMAND\displaywidowpenalty	146
COMMAND\do@actions	137–139, 381
COMMAND\do@actions@fixedcode	381
COMMAND\do@actions@next	138, 139
COMMAND\do@ballast	137, 146
COMMAND\do@feetX	391
COMMAND\do@insidelinehook	384
COMMAND\do@line	91, 113, 129, 133, 135, 136, 145, 146, 273, 381, 382, 384, 386
COMMAND\do@linehook	381
COMMAND\do@lockoff	92
COMMAND\do@lockon	92
COMMAND\dodoreintrafeet	380
COMMAND\doendnotes	24, 208, 209, 335, 390
COMMAND\doendnotesbysection	24, 209, 221, 390
COMMAND\doinsidelinehook	22, 387
COMMAND\dolinehook	22, 387
COMMAND\doreintrafeeti	391
COMMAND\doreintrafeetii	391
COMMAND\doxtrafeet	233, 380
COMMAND\doxtrafeeti	391
COMMAND\doxtrafeetii	391
COMMAND\dummy@ref	114
COMMAND\edaftertab	59, 303–305
COMMAND\edatleft	59, 302
COMMAND\edatright	59, 302
COMMAND\edbeforetab	59, 303, 304
COMMAND\edfilldimen	302
COMMAND\edfont@info	120
COMMAND\edgls	56, 263
COMMAND\edgls...	394
COMMAND\edindex	54–56, 263, 264, 267–269, 292, 384, 387, 388, 391, 392
COMMAND\edindexlab	56
COMMAND\edlabel	48–51, 53, 114, 237–240, 243, 250, 263, 292, 324, 380, 383–385, 388, 393
COMMAND\edlabelE	50, 242
COMMAND\edlabelS	50, 242
COMMAND\edlabelSE	50
COMMAND\edlineref	48, 49, 237, 334, 388, 390, 393
COMMAND\edmakelabel	50, 250
COMMAND\edpageref	48, 49, 237, 243, 250
COMMAND\edrowfill	303
COMMAND\edtabcolsep	297
COMMAND\edtext	6, 23, 25–28, 44, 49–51, 54, 57, 64, 90, 91, 105, 106, 109, 112–120, 122–124, 126, 241, 242, 245, 292, 294, 313, 324, 328, 329, 332, 380, 382, 384, 386–390

COMMAND\edtext@level	390
COMMAND\edvertdots	60, 302
COMMAND\edvertline	59, 60, 302
COMMAND\elechapter	61
COMMAND\eled@sectioning@out	317
COMMAND\eledchapter	60, 332, 387, 391
COMMAND\eledchapter*	60
COMMAND\eledmac@error	380
COMMAND\eledsection	6, 15, 60, 114, 135, 311, 332, 389
COMMAND\eledsection*	60
COMMAND\eledsubsection	60, 332
COMMAND\eledsubsection*	60
COMMAND\eledsubsubsection	60, 332
COMMAND\eledsubsubsection*	60
COMMAND\eledxxx	10, 61, 312, 317, 386
COMMAND\eledxxxx	311
COMMAND\else	262, 311
COMMAND\empty	76, 141, 238
COMMAND\end	287, 288
COMMAND\end@lemmas	113, 114
COMMAND\endashchar	43, 170
COMMAND\endgraf	129, 161, 200
COMMAND\endlock	21, 92, 111, 277
COMMAND\endminipage	258, 260, 380, 391
COMMAND\endmsdata	30
COMMAND\endnotes	389, 393
COMMAND\endnumbering	16, 18, 77, 78, 80, 381, 390
COMMAND\endprint	205, 208, 221, 331
COMMAND\endstanzaextra	333
COMMAND\endsub	20, 92, 110
COMMAND\endverse	335
COMMAND\everypar	132
COMMAND\extensionchars	62, 78
COMMAND\externaldocument	53, 250
COMMAND\f@x@l@cks	381
COMMAND>falseverse	332, 384, 386
COMMAND\fi	311
COMMAND\firstlinenum	19, 140, 382
COMMAND\firstsublinenum	19, 382
COMMAND\fix@page	97, 98, 381
COMMAND\flag@end	109, 119, 386
COMMAND\flag@start	109, 119, 386, 387
COMMAND\flagstanza	47
COMMAND\floatingpenalty	151, 152, 388
COMMAND\flush@notes	147
COMMAND\fnpos	202, 385
COMMAND\footfmt	151, 154
COMMAND\footfmt...	185
COMMAND\footfootmarkA	30
COMMAND\footfudgefactor	160

COMMAND\footfudgefiddle	64, 157, 158, 380
COMMAND\footgroup	151
COMMAND\footins	155
COMMAND\footnormal	225, 333, 380
COMMAND\footnormalX	333
COMMAND\footnote	30, 63, 180–182, 330, 381
COMMAND\footnote@lang	170
COMMAND\footnoteA	15, 30
COMMAND\footnoteB	15
COMMAND\footnoteC	22
COMMAND\footnoteE	30
COMMAND\footnoteX	8, 219
COMMAND\footnoteX@reading	220
COMMAND\footnoteXmk	232
COMMAND\footnotelang@lua	150
COMMAND\footnotelang@poly	150
COMMAND\footnoteoption@	149, 392
COMMAND\footnoterule	165
COMMAND\footnotesize	39
COMMAND\footparagraph	158, 225, 333, 386
COMMAND\footparagraphX	194, 333, 386
COMMAND\footplitskips	381, 388
COMMAND\footstart	151, 155, 165
COMMAND\footstrut	161
COMMAND\footthreecol	333
COMMAND\footthreecolX	333, 390
COMMAND\foottwocol	333
COMMAND\foottwocolX	333, 390
COMMAND\foreignlanguage	39
COMMAND\fullstop	43, 336
COMMAND\get@edindex@hyperref	268
COMMAND\get@edindex@ledinnote@command	264
COMMAND\get@fnmark	182
COMMAND\get@index@command	385
COMMAND\get@linelistfile	381
COMMAND\get@thisfootnote	188
COMMAND\getline@num	137, 138
COMMAND\gl@p	82
COMMAND\global	96
COMMAND\globaldefs	96
COMMAND\gls	56, 270
COMMAND\hangindentX	39, 389, 392
COMMAND\hangingsymbol	333, 382
COMMAND\hbox	159
COMMAND\hfill	385
COMMAND\hidenumbering	22, 104, 389
COMMAND\hline	57
COMMAND\hrulefill	303
COMMAND\hspace	33, 155, 158, 160, 166, 168, 200, 381, 387
COMMAND\hspaceX	336, 393, 394

COMMAND\hsizethreecol	334
COMMAND\hsizethreecolX	41, 43
COMMAND\hsizetwocol	334
COMMAND\hsizetwocolX	41, 43
COMMAND\hyperlinkR	267
COMMAND\hyperlinkformat	267
COMMAND\hyperlinkformatR	267
COMMAND\if@RTL	70
COMMAND\if@edtext@	387, 390
COMMAND\if@eled@sectioning	312
COMMAND\if@firstlineofpage	70
COMMAND\if@firstlineofpageR	70
COMMAND\if@msd@options@fullpage	286
COMMAND\if@msdata@insertedfrompreviouspage	284
COMMAND\if@nobreak	130
COMMAND\if@noneed@Footnote	109
COMMAND\ifXnote@	77
COMMAND\ifbypage@	83
COMMAND\ifbypage@R	83
COMMAND\ifbypstart@	83
COMMAND\ifbypstart@R	83
COMMAND\iffirst@linenum@out@	107, 108
COMMAND\ifindtl@innote	77
COMMAND\ifindtl@notenumber	77
COMMAND\ifinserthangingsymbol	272
COMMAND\ifinstanza	272
COMMAND\ifistwofollowinglines	177, 178
COMMAND\ifl@d@Xmorethantwolines	175, 388
COMMAND\ifl@d@Xtwolines	175
COMMAND\ifl@d@dash	175
COMMAND\ifl@d@elin	175
COMMAND\ifl@d@esl	175
COMMAND\ifl@d@pnum	175
COMMAND\ifl@d@ssub	175
COMMAND\ifl@dend@X	220
COMMAND\ifl@dmemoir	380
COMMAND\ifl@dpaging	387
COMMAND\ifl@dpairing	77, 382
COMMAND\ifl@dprintingpages	388
COMMAND\ifl@dskipnumber	140
COMMAND\ifl@dstartendok	303
COMMAND\ifl@imakeidx	70
COMMAND\ifledRcol	77, 382
COMMAND\ifledRcol@	77, 386
COMMAND\iflemmacommand@	387
COMMAND\ifnoend@	209
COMMAND\ifnoledgroup@	262
COMMAND\ifnoteschanged@	93
COMMAND\ifnumberedpar@	128
COMMAND\ifnumbering	78, 80

COMMAND\ifnumberingR	77, 382
COMMAND\ifnumberline	119, 140
COMMAND\ifpst@rted	382
COMMAND\ifpst@rtedL	78
COMMAND\ifseriesbefore	223
COMMAND\ifstopmsdata@inserted@	280
COMMAND\ifsublines@	90, 102
COMMAND\iftrue	390
COMMAND\ifvmode	239
COMMAND\ifxxx	311
COMMAND\ignorespaces	117
COMMAND\imki@wrindexentry	70
COMMAND\immediate	107, 108, 205
COMMAND\indent	17, 131, 335
COMMAND\index	270
COMMAND\indtl@wrindexentry	70
COMMAND\initnumbering@quote	309, 391
COMMAND\initnumbering@reg	381
COMMAND\initnumbering@sectcmd	391
COMMAND\inplaceoflemmaseparator	334
COMMAND\inplaceofnumber	334
COMMAND\insert	145, 151, 154, 185
COMMAND\insert@Xtxtbeforenotes	148
COMMAND\insert@count	105, 109, 117
COMMAND\insert@countR	117
COMMAND\insert@msdata	281, 286
COMMAND\inserthangingsymbol	385
COMMAND\insertlines@list	91, 105
COMMAND\insertparafootsepX	198
COMMAND\inserts@list	113, 128, 145, 159
COMMAND\interAfootnotelinepenalty	382
COMMAND\interfootnotelinepenalty	382
COMMAND\interlinepenalty	151
COMMAND\interparanoteglue	332
COMMAND\justifying	40
COMMAND\l@advance@parledegroup@beforenormalnotes	391
COMMAND\l@d@@wrindexhyp	387
COMMAND\l@d@add	121
COMMAND\l@d@end	205, 220
COMMAND\l@d@nums	117, 119, 121, 175
COMMAND\l@d@section	205
COMMAND\l@d@set	101, 111
COMMAND\l@dampcount	294
COMMAND\l@dbfnote	182, 381
COMMAND\l@dcheckstartend	303
COMMAND\l@dchset@num	101
COMMAND\l@dcolcount	294, 295
COMMAND\l@dcollect@@body	288
COMMAND\l@dcollect@body	287
COMMAND\l@dcsnote	386

COMMAND\l@dcsnotetext	136, 255
COMMAND\l@dcsnotetext@l	136, 255
COMMAND\l@dcsnotetext@r	136, 255
COMMAND\l@ddodoreintrafeet	234, 380
COMMAND\l@ddoxtrafeet	234, 380
COMMAND\l@demptyd@ta	382
COMMAND\l@dend@close	205
COMMAND\l@dend@open	205
COMMAND\l@dend@stuff	205
COMMAND\l@denvbody	288
COMMAND\l@dfeetbeginmini	380
COMMAND\l@dfeetendmini	380
COMMAND\l@dgetline@margin	382
COMMAND\l@dgetlock@disp	382
COMMAND\l@dgetref@num	244
COMMAND\l@dgetsidenote@margin	252, 382
COMMAND\l@dgobbeloptarg	387
COMMAND\l@dgonblearg	387
COMMAND\l@dgonbleoptarg	293
COMMAND\l@dlabel@parse	244
COMMAND\l@dld@ta	140, 142
COMMAND\l@dldp@rbox	256
COMMAND\l@dlsn@te	382
COMMAND\l@dlsnote	386
COMMAND\l@dmake@labels	240, 251
COMMAND\l@dmake@labelsR	251
COMMAND\l@dnumpstartsL	78, 382
COMMAND\l@dp@rsefootspec	175
COMMAND\l@dpush@begins	288
COMMAND\l@drd@ta	140, 142
COMMAND\l@dref@undefined	244
COMMAND\l@drrsn@te	382
COMMAND\l@drrsnote	386
COMMAND\l@dtabaddcols	302
COMMAND\l@dtabnoexpands	380
COMMAND\l@dumboxmpfoot	391
COMMAND\l@dunboxmpfoot	382
COMMAND\l@dzeropenalties	382, 387
COMMAND\l@pb	323
COMMAND\l@prev@nopb	322
COMMAND\l@prev@pb	322
COMMAND\l@reg	330
COMMAND\label	18, 50, 53, 56, 238, 245
COMMAND\label@refs	238
COMMAND\labelstarttrue	18, 383
COMMAND\labelref@list	237, 238, 240
COMMAND\language	159
COMMAND\last@page@num	381
COMMAND\lastbox	132
COMMAND\lastskip	110

COMMAND\leavevmode	17, 131
COMMAND\led@check@nopb	323
COMMAND\led@check@pb	323
COMMAND\led@nopb	322–324
COMMAND\led@nopbnum	322, 323
COMMAND\led@pb	322–324
COMMAND\led@pb@macro	323
COMMAND\led@pbnum	322, 323
COMMAND\led@reinit@index@fornote	270
COMMAND\led@set@index@fornote	270
COMMAND\ledRflag	267
COMMAND\ledchapter	332, 384
COMMAND\ledfootinsdim	333
COMMAND\ledinnernote	53, 253, 386
COMMAND\ledinnote	265, 331, 390
COMMAND\ledinnotemark	55, 331, 389
COMMAND\ledleftnote	53, 253
COMMAND\ledlinenum	88, 382
COMMAND\ledllfill	136
COMMAND\ledlsnotefontsetup	395
COMMAND\ledlsnotesep	54
COMMAND\ledlsnotewidth	53
COMMAND\lednopb	62, 322
COMMAND\lednopbinverse	324
COMMAND\lednopbinversetrue	46, 62
COMMAND\lednopbnum	322
COMMAND\ledouternote	53, 253, 386
COMMAND\ledpb	62, 322, 332
COMMAND\ledpbnum	322
COMMAND\ledpbsetting	62, 323, 392
COMMAND\ledrightnote	53, 253
COMMAND\ledrsnotefontsetup	395
COMMAND\ledrsnotesep	54
COMMAND\ledrsnotewidth	53
COMMAND\ledsection	332
COMMAND\ledsectnomark	310
COMMAND\ledsectnotoc	310
COMMAND\ledsetnormalparstuff	331, 332, 389
COMMAND\ledsetnormalparstuff@common	200
COMMAND\ledsetnormalparstuffX	331, 332, 389
COMMAND\ledsidenote	53, 253, 255
COMMAND\ledsubsection	332
COMMAND\ledsubsubsection	332
COMMAND\ledxxx	386
COMMAND\left	59
COMMAND\leftctab	304
COMMAND\leftheadline	88
COMMAND\leftlinenum	20, 88, 380, 382
COMMAND\leftltab	303
COMMAND\leftnoteupfalse	53

COMMAND\leftstartnum	144
COMMAND\lefttab	304
COMMAND\leftsidenote	255
COMMAND\leftskip	155, 158, 160
COMMAND\lemma	3, 23, 25–28, 113, 116, 117, 120, 122, 328, 382, 383, 390, 391, 393
COMMAND\lemmaseparator	334
COMMAND\let	26, 44, 277, 380
COMMAND\letboxXendendlinenum	334
COMMAND\line	164, 167
COMMAND\line@list	90, 106, 119
COMMAND\line@list@stuff	78, 94, 108, 380, 382
COMMAND\line@list@version	96
COMMAND\line@margin	84, 142, 251
COMMAND\line@num	89, 90, 92, 140, 380
COMMAND\line@set	121
COMMAND\lineation	20, 83
COMMAND\linebreak	33
COMMAND\linenum	23, 25, 26, 49–51, 113, 121, 243, 245, 250, 328, 394
COMMAND\linenum@out	107, 238, 240
COMMAND\linenumberlist	19, 76, 141, 380
COMMAND\linenumberstyle	21, 88, 380
COMMAND\linenumincrement	19, 382
COMMAND\linenummargin	20, 84, 251
COMMAND\linenumr@p	88, 380, 382
COMMAND\linenumrep	88, 382
COMMAND\linenumsep	20, 54, 88, 253
COMMAND\linerangesep@	232
COMMAND\lineref	237, 243, 250, 334, 388
COMMAND\list@clear	82
COMMAND\list@clearing@reg	382
COMMAND\list@create	82
COMMAND\lock@disp	87
COMMAND\lock@off	103
COMMAND\lock@on	102
COMMAND\lockdisp	21, 87
COMMAND\loop	147, 148, 273
COMMAND\ltab	303, 304, 309
COMMAND\ltabtext	309
COMMAND\m@mmf@prepare	181
COMMAND\makeatletter	136
COMMAND\makehboxoffhboxes	160, 162
COMMAND\makeindex	54, 267
COMMAND\makelabel	250
COMMAND\managestanza@modulo	274
COMMAND\marginpar	53, 63, 251, 381
COMMAND\marginparwidth	53, 253
COMMAND\markboth	135
COMMAND\mathchardef	274
COMMAND\maxhXnotes	334
COMMAND\maxhnotesX	42, 64, 333, 385, 386, 388–390

COMMAND\maxlinesinpar@list	94
COMMAND\measurembody	305
COMMAND\measurtbody	306
COMMAND\memorybreak	19
COMMAND\morenoexpands	64, 65, 113, 115
COMMAND\morethantwolines	334
COMMAND\mpfnpos	202, 385
COMMAND\mpnormalfootgroup	381
COMMAND\mpnormalvfootnote	381
COMMAND\msdata	29, 30, 279–281
COMMAND\multfootsep	31, 181
COMMAND\multiplefootnotemarker	181
COMMAND\musixtex	386
COMMAND\n@num	382, 389
COMMAND\n@num@ref	389
COMMAND\new@line	109, 381
COMMAND\newcommand	26, 66, 180, 240
COMMAND\newcommandx	26
COMMAND\newhookarg@specific	231
COMMAND\newhookcommand@series	226, 227, 389
COMMAND\newhookcommand@series@reload	227
COMMAND\newhookcommand@toggle@reload	227, 387
COMMAND\newhooktoggle@series	226, 227, 389
COMMAND\newhooktoggle@specific	231
COMMAND\newif	389
COMMAND\newline	33
COMMAND\newlinechar	220
COMMAND\newseries	31, 329, 332, 333
COMMAND\newseries@	213, 224
COMMAND\newverse	47, 332, 386
COMMAND\next	273
COMMAND\next@action	95
COMMAND\next@actionline	95
COMMAND\next@insert	146
COMMAND\nl@regR	97
COMMAND\no@expands	64, 120, 380
COMMAND\noXlemmaseparator	334
COMMAND\nobreak	174
COMMAND\nocritical	214
COMMAND\noeledsec	61, 332
COMMAND\noendnotes	332
COMMAND\noexpand	330
COMMAND\nofamiliar	229
COMMAND\noindent	17, 18, 131, 392
COMMAND\noindentX	392
COMMAND\nomk@	232
COMMAND\nonbreakableafternumber	334
COMMAND\nonnumberinfootnote	334
COMMAND\normal@footnotemarkX	184
COMMAND\normal@page@break	322

COMMAND\normal@pars	200
COMMAND\normalbfnoteX	382
COMMAND\normalbodyfootmarkX	185
COMMAND\normalfootfmt	44, 154, 161, 170, 205, 394
COMMAND\normalfootfmtX	185, 186
COMMAND\normalfootfootmarkX	186
COMMAND\normalfootgroup	156
COMMAND\normalfootgroupX	187
COMMAND\normalfootnoterule	152
COMMAND\normalfootstart	155, 158
COMMAND\normalfootstartX	186
COMMAND\normalvfootnote	154
COMMAND\normalvfootnoteX	185
COMMAND\notbool	311
COMMAND\notfontsetup	333
COMMAND\notfontsizeX	39, 333
COMMAND\notenumfont	333
COMMAND\notenumfontX	38, 333
COMMAND\notesXwidthliketwocolumns	334
COMMAND\noteswidthliketwocolumnsX	43, 334, 387, 389
COMMAND\num@lines	128, 146
COMMAND\numberlinefalse	19
COMMAND\numberlinetrue	19
COMMAND\numberonlyfirstinline	224, 334
COMMAND\numberonlyfirstintwolines	334
COMMAND\numberpstartfalse	18
COMMAND\numberpstarttrue	18, 34, 331, 383, 391
COMMAND\numberstanza	35
COMMAND\numberstanzafalse	47
COMMAND\numberstanzatrue	47
COMMAND\numlabfont	20, 43, 44, 88
COMMAND\none@line	128
COMMAND\nonehalfspacing	392
COMMAND\nonlyXpstart	334
COMMAND\npage@action	91, 101
COMMAND\npage@start	91, 382
COMMAND\npagecontents	91
COMMAND\npagelinesep	55
COMMAND\npageno	232
COMMAND\npageparbreak	332
COMMAND\npageref	50, 243
COMMAND\npar	25, 33, 131, 132, 200
COMMAND\npar@line	128, 146
COMMAND\npara@footgroup	158
COMMAND\npara@footgroupX	197
COMMAND\npara@footsetup	158, 380
COMMAND\npara@footsetupX	195, 380, 387
COMMAND\npara@vfootnoteX	196
COMMAND\nparafootfmt	160, 161, 394
COMMAND\nparafootfmtX	197

COMMAND\parafootftm	163
COMMAND\parafootftmX	198
COMMAND\parafootftmsep	333
COMMAND\parafootsep	334, 385, 390
COMMAND\parafootsepX	41, 93, 333, 334, 395
COMMAND\parafootstart	158
COMMAND\parafootstartX	195
COMMAND\paravfootnote	159, 162
COMMAND\parfillskip	161
COMMAND\parindent	392
COMMAND\parindentX	39, 336, 392, 394
COMMAND\parshape	64
COMMAND\parskip	132
COMMAND\pausenumbering	18, 80, 81, 94, 96, 132, 385, 387, 394
COMMAND\penalty	161
COMMAND\pend	2, 6, 16–21, 61, 111, 113, 116, 121, 128–133, 144, 145, 330, 385, 386
COMMAND\preXnotes	42, 203, 389
COMMAND\preXnotes@	155, 203, 383
COMMAND\prenotesX	42, 204, 389
COMMAND\prepare@preXnotes	202
COMMAND\prev@nopb	323
COMMAND\prev@pb	323
COMMAND\prevlineX	93
COMMAND\prevpageX@num	93
COMMAND\print@Xfootnoterule	389
COMMAND\print@Xnotes	233, 234
COMMAND\print@Xnotes@forpages	388
COMMAND\print@eledsection	135
COMMAND\print@footnoteXrule	389
COMMAND\print@leftmargin@eledsection	312
COMMAND\print@lemma	171
COMMAND\print@line	133
COMMAND\print@notesX@forpages	388
COMMAND\print@rightmargin@eledsection	312
COMMAND\printendlines	209, 248, 336, 380, 382
COMMAND\printlinefootnote	172, 173, 388
COMMAND\printlinefootnotearea	173, 174, 388
COMMAND\printlinefootnotenumbers	172
COMMAND\printlines	154, 170, 174, 176, 209, 248, 336, 380, 382, 388, 393
COMMAND\printnpnum	331, 336
COMMAND\printpstart	170
COMMAND\protect	114, 330
COMMAND\providecommand	180, 380
COMMAND\pstart	2, 6, 16–21, 60, 61, 101, 111, 116, 121, 128, 130–132, 135, 145, 330, 382–384, 386, 387, 389–391, 394
COMMAND\pstartinfootnote	334
COMMAND\pstartinfootnoteeverytime	334
COMMAND\pstartnum	144
COMMAND\pstartref	49, 237, 243, 385
COMMAND\pstarts	383

COMMAND\raggedX	41
COMMAND\raggedleft	40
COMMAND\raggedright	40
COMMAND\raw@text	128
COMMAND\rbracket	37, 38, 43
COMMAND\read@linelist	94–96
COMMAND\ref	50, 53, 56
COMMAND\reformatted@	248
COMMAND\reformattedwithpage	248
COMMAND\relax	18, 101, 138, 145, 277, 293, 330
COMMAND\renewcommand	64, 333, 336
COMMAND\reset@msd@options@iffullpage	286
COMMAND\resetprevline@	93
COMMAND\resetprevpage@	93
COMMAND\resumenumbering	18, 77, 81, 94, 96, 132, 382, 386, 387, 394
COMMAND\right	59
COMMAND\rightctab	304
COMMAND\rightlinenum	20, 88, 380, 382
COMMAND\rightltab	304
COMMAND\rightnoteupfalse	53
COMMAND\rightrtab	305
COMMAND\rightright	255
COMMAND\rightright	155, 158, 160, 161
COMMAND\rightright	144
COMMAND\rigidbalance	164, 165, 167, 336, 393
COMMAND\rigidbalanceX	164, 336, 393
COMMAND\robustify	33
COMMAND\roman	297, 393
COMMAND\rtab	304, 305, 309
COMMAND\rtabtext	306, 309
COMMAND\sameword	26–28, 122–124, 126, 388, 390, 392
COMMAND\sameword@inedtext	123
COMMAND\saweword	122
COMMAND\scriptsize	88
COMMAND\section	60, 382
COMMAND\section@num	77
COMMAND\sectionmark	310
COMMAND\select@lemmafont	44, 149
COMMAND\series	213
COMMAND\series@	213
COMMAND\seriesatbegin	31, 223, 389
COMMAND\seriesatend	31, 223, 389
COMMAND\set@Xtxtbeforenotes	148
COMMAND\set@line	119
COMMAND\set@line@action	92, 101
COMMAND\setSErefonlypageprefixmore	51, 247, 394
COMMAND\setSErefonlypageprefixsingle	51, 247, 394
COMMAND\setSErefprefixmore	51
COMMAND\setSErefprefixsingle	51
COMMAND\setapprefprefixmore	51, 333

COMMAND\setapprefprefixsingle	51, 333, 393
COMMAND\setcommand@series	225
COMMAND\sethangingsymbol	46, 272, 333, 392
COMMAND\sethanginsymbol	45
COMMAND\setistwoofollowinglines	177
COMMAND\setl@dlprbox	256
COMMAND\setline	21, 92, 97, 100, 111, 114, 130, 391
COMMAND\setlinenum	21, 97, 101, 111, 380
COMMAND\setmsdatalabel	30
COMMAND\setmsdataseries	30
COMMAND\setprintendlines	209, 211, 382
COMMAND\setprintlines	176, 177, 209, 382
COMMAND\setsidenotesep	54
COMMAND\setsidenotsep	333
COMMAND\setstanzaindent	274
COMMAND\setstanzaindents	45, 274, 330
COMMAND\setstanzapenalties	274
COMMAND\setstanzavalues	274
COMMAND\settoggle@series	224, 383, 387
COMMAND\showlemma	114, 380, 381
COMMAND\showwordrank	28, 123
COMMAND\sidenote@margin	381
COMMAND\sidenotemargin	53, 381, 386
COMMAND\sidenotesep	333
COMMAND\sidesstartnumtrue	18
COMMAND\skip	155
COMMAND\skipnumbering	22, 104, 112, 382, 389, 390
COMMAND\skipnumbering@reg	389
COMMAND\small	39
COMMAND\special	12
COMMAND\splitmaxdepth	151, 166
COMMAND\splitoff	164
COMMAND\splittopskip	151, 166, 167
COMMAND\stanza	21, 22, 46, 47, 277, 333, 335, 392
COMMAND\stanza@hang	276
COMMAND\stanza@line	276
COMMAND\stanzaindent	45, 274, 388
COMMAND\stanzaindent*	45
COMMAND\stanzaindentbase	273, 274
COMMAND\stanzanumwrapper	47
COMMAND\startlock	21, 92, 111, 277
COMMAND\startstanzahook	333
COMMAND\startsub	20, 92, 110
COMMAND\stopmsd	281
COMMAND\stopmsdata	29, 279, 280
COMMAND\strip@pt	158
COMMAND\strutbox	166
COMMAND\sub@action	92, 102
COMMAND\sub@lock	90
COMMAND\sub@off	99, 240

COMMAND\sub@on	99, 240
COMMAND\subline@num	89, 90, 92
COMMAND\sublinenum@rep	380
COMMAND\sublinenumberstyle	22, 88, 380
COMMAND\sublinenumincrement	19
COMMAND\sublinenumr@p	88, 380, 382
COMMAND\sublinenumrep	88, 382
COMMAND\sublineref	49, 237, 243
COMMAND\subsectionmark	310
COMMAND\sw@inthisedtext	116
COMMAND\sw@list@inedtext	120, 126
COMMAND\symlinenum	334
COMMAND\symplinenum	333
COMMAND\sza@penalty	276
COMMAND>tag	388
COMMAND\text	328
COMMAND\text{ <i>language</i> }	39
COMMAND\textcolor	65
COMMAND\textheight	64
COMMAND\the	380
COMMAND\thefootnoteA	30
COMMAND\thefootnoteX	384
COMMAND\thelabidx	268, 269
COMMAND\thepage	97
COMMAND\thepstart	18
COMMAND\thepstartL	383
COMMAND\thepstartR	383
COMMAND\thestanza	47
COMMAND\this@line@list@version	108
COMMAND\thisfootnote	188
COMMAND\threecolfootfmt	166, 394
COMMAND\threecolfootfmtX	193
COMMAND\threecolfootgroup	165
COMMAND\threecolfootgroupX	194
COMMAND\threecolfootsetup	165
COMMAND\threecolfootsetupX	193
COMMAND\threecolvfootnote	166
COMMAND\threecolvfootnoteX	193
COMMAND\twocolfootfmt	394
COMMAND\twocolfootfmtX	191
COMMAND\twocolfootgroupX	192
COMMAND\twocolfootsetupX	191
COMMAND\twocolvfootnoteX	191
COMMAND\twolines	224, 334
COMMAND\twolines@A	224
COMMAND\twolines@B	224
COMMAND\twolines@C	224
COMMAND\twolinesbutnotmore	334
COMMAND\twolinesonlyinsamepage	334
COMMAND\txbeforeXnotes	334

COMMAND\unhbox	159
COMMAND\unpenalty	160–162
COMMAND\unskip	161
COMMAND\unvxh	161, 335
COMMAND\unvxhX	335
COMMAND\upbracefill	303
COMMAND\usingcritext	329, 332
COMMAND\usingdtext	329, 332
COMMAND\VAfootnote	151
COMMAND\variant	26
COMMAND\ vbox	129, 131, 159, 164, 202
COMMAND\ vfootnote	151, 155, 159, 166
COMMAND\ vl@dbfnote	182, 381
COMMAND\ vnumfootnoteX	382
COMMAND\ vsize	42, 64
COMMAND\ vsplit	146
COMMAND\ waklam	303
COMMAND\ waklamec	303
COMMAND\ wapunktel	303
COMMAND\ wastricht	303
COMMAND\ widthX	43, 336, 394
COMMAND\ wrap@edcrossref	242, 387
COMMAND\ wrapcontentX	39, 395
COMMAND\ wrapped@bodyfootmarkX	199
COMMAND\ wrapped@footfootmarkX	199
COMMAND\ x...	49
COMMAND\ xdef	82, 277
COMMAND\ xflagref	49, 243, 336, 393
COMMAND\ xleft@appenditem	82, 113
COMMAND\ xlineref	49, 336, 393
COMMAND\ xpageref	49
COMMAND\ xpstartref	49, 385
COMMAND\ xr	53
COMMAND\ xright@appenditem	82
COMMAND\ xsublineref	49
COMMAND\ xxref	50, 244, 245, 250, 385, 388
COMMAND\ zz@@@	380
ENVIRONMENTastanza	393
ENVIRONMENTedarrayc	309
ENVIRONMENTedarrayl	309
ENVIRONMENTedarrayr	309
ENVIRONMENTedtabularc	309
ENVIRONMENTedtabularl	309
ENVIRONMENTedtabularr	309
ENVIRONMENTledgroup	69, 261, 336, 393
ENVIRONMENTledgroup sized	261
PACKAGE(r)(e)ledmac	31
PACKAGEEledmac	11, 66, 91, 266, 331, 332, 388, 389
PACKAGEEledpar	389
PACKAGEEtoolbox	68

PACKAGEParallel	337
PACKAGEReledmac	335, 336
PACKAGEamsgen	287
PACKAGEamsmath	287
PACKAGEbabel	39, 65, 297, 393
PACKAGEbiblatex	63
PACKAGEbidi	39, 40, 70, 392
PACKAGEcaption	76
PACKAGEcolor	65
PACKAGEedmac	1, 6, 10, 12, 13, 66, 175, 180, 238, 274, 328, 337, 380
PACKAGEedstanza	1, 13, 272
PACKAGEeledmac	1, 10, 13–15, 55, 122, 180, 262, 266, 290, 313, 325, 329, 331–333, 384, 386, 388
PACKAGEeledpar	77, 151, 311, 337, 382, 386–388
PACKAGEetex	392
PACKAGEetoolbox	82, 122, 213, 224, 232, 255, 311, 322
PACKAGEfloatrow	63, 336
PACKAGEfootmisc	31, 65, 180, 337
PACKAGEgeometry	14
PACKAGEglossaries	56, 270, 393
PACKAGEhandout	387
PACKAGEhyperlink	220
PACKAGEhyperref	49, 116, 199, 238, 268, 316, 324, 385–387, 394
PACKAGEifluatex	69
PACKAGEifxetex	69
PACKAGEimakeidx	54, 63, 69, 70, 262, 266, 332, 384–386, 388
PACKAGEindextols	270
PACKAGEindextool	332
PACKAGEindextools	54, 63, 69, 70, 77, 262, 266, 270, 332, 388, 393
PACKAGEinputenc	123
PACKAGEledarab	65
PACKAGEledmac	1, 10, 13, 65, 82, 266, 328, 329, 332, 335
PACKAGEledpar	65
PACKAGEMemoir	69, 266, 332, 337, 387
PACKAGEMorewrites	63
PACKAGEMusixtex	386
PACKAGEperpage	393
PACKAGEpolyglossia	37, 65, 117, 150, 170, 393
PACKAGERagged2e	40, 69
PACKAGEREledmac	1, 2, 10, 11, 13–16, 18, 19, 22, 24–27, 29, 31, 33, 36, 39, 40, 42–44, 46, 47, 49–51, 53–57, 61, 63–67, 83, 85, 91, 92, 95, 96, 99, 107, 108, 114, 115, 145, 152, 155, 160, 180, 205, 214, 218, 224, 232, 233, 242, 245, 250, 266, 284, 290, 311, 312, 324, 332, 333, 335, 336, 391, 394
PACKAGEREledpar	1, 4, 6, 8, 15, 18, 43, 49, 51, 52, 61–63, 65, 67, 77, 83, 94, 99, 117, 153, 155, 200, 201, 213, 219, 232–234, 262, 272, 392, 393
PACKAGESuffix	69
PACKAGETabmac	1, 13, 337
PACKAGEuninormalize	27
PACKAGExargs	26, 68
PACKAGExkeyval	67, 232
PACKAGEXr	4, 53, 250, 251, 394

PACKAGEXref	250
PACKAGEXstring	69, 268

A

\absline@num	<u>1</u>
Abu Kamil Shuja' b. Aslam	12
\actionlines@list	<u>1</u>
\actions@list	<u>1</u>
\add@inserts	<u>1</u>
\add@inserts@next	<u>1</u>
\add@msdata	<u>1</u>
\add@msdata@firstlineofpage	<u>1</u>
\add@penalties	<u>1</u>
\addtol@denvbody	<u>1</u>
Adelard II	12
\advancelabel@refs	<u>1</u>
\advanceline	<u>1</u> , 21
\advancepageno	<u>1</u>
\Aendnote	24
\affixline@num	<u>1</u>
\affixpstart@num	<u>1</u>
\affixside@note	<u>1</u>
\Afootnote	23
\afternoteX	41
\afterruleX	42
\ampersand	<u>1</u> , 47
\applabel	<u>1</u> , 51
\appref	<u>1</u> , 51
\apprefwithpage	<u>1</u> , 51
\arrangementX	<u>1</u> , 32
\arrangementX@normal	<u>1</u>
\arrangementX@threecol	<u>1</u>
\arrangementX@twocol	<u>1</u>
\at@every@pend	<u>1</u>
\AtEveryPend	<u>1</u> , 18
\AtEveryPstart	<u>1</u> , 18
\AtEveryStanza	<u>1</u>
\AtEveryStopStanza	<u>1</u>
\autopar	<u>1</u> , 17

B

\ballast	64
\ballast@count	<u>1</u>
Beeton, Barbara Ann Neuhaus Friend	17
\beforeeledchapter	<u>1</u>
\beforeinsertingX	40
\beforenotesX	42
\beginnumbering	<u>1</u> , 16
\Bendnote	24
\Bfootnote	23

\bhookgroupX	41
\bhooknoteX	40
\bodyfootmarkA	30
\boxfootnotenumbers	1
Bredon, Simon	12
Breger, Herbert	13, 290
Brey, Gerhard	12
Busard, Hubert L. L.	12
\bypage@false	1
\bypage@true	1
\bypstart@false	1
\bypstart@true	1

C

\c@addcolcount	1
\c@ballast	1
\c@firstlinenum	1
\c@firstsublinenum	1
\c@labidx	1
\c@linenumincrement	1
\c@sublinenumincrement	1
\Cendnote	24
\Cfootnote	23
\ch@ck@l@ck	1
\ch@cksub@l@ck	1
\chapter	1
\check@pb@in@verse	1
Chester, Robert of	12
Claassens, Geert H. M.	13
\colalignX	40
Copernicus, Nicolaus	12
\critext	328
\ctab	1
\ctabtext	1

D

Dekker, Dirk-Jan	65
\Dendnote	24
\Dfootnote	23
\disable@familiarnotes	1
\disable@notes	1
\disable@sidenotes	1
\disablel@dtabfeet	1
\do@actions	1
\do@actions@fixedcode	1
\do@actions@next	1
\do@ballast	1
\do@feetX	1
\do@insidelinehook	1
\do@line	1

<code>\do@linehook</code>	1
<code>\do@lockoff</code>	1
<code>\do@lockoffL</code>	1
<code>\do@lockon</code>	1
<code>\do@lockonL</code>	1
<code>\doedindexlabel</code>	1
<code>\doendnotes</code>	1, 24
<code>\doendnotesbysection</code>	1, 24
<code>\doinsidelinehook</code>	1, 22
<code>\dolinehook</code>	1, 22
<code>\dosplits</code>	1
Downes, Michael	64, 159, 161
<code>\doxtrafeet</code>	1
<code>\dummy@edtext</code>	1
<code>\dummy@edtext@showlemma</code>	1
<code>\dummy@ref</code>	1

E

<code>\edaftertab</code>	1, 59, 303
<code>edarrayc</code> (environment)	57
<code>edarrayl</code> (environment)	57
<code>edarrayr</code> (environment)	57
<code>\edatleft</code>	1, 59
<code>\edatright</code>	1, 59
<code>\edbeforetab</code>	1, 59, 303
<code>\edfilldimen</code>	1
<code>\edfont@info</code>	1
<code>\EDINDEX</code>	1
<code>\edindex</code>	1, 54
<code>\edindexlab</code>	1, 56
<code>\EDLABEL</code>	1
<code>\edlabel</code>	1, 48
<code>\edlabelE</code>	1, 50
<code>\edlabelS</code>	1, 50
<code>\edlabelSE</code>	1, 50
<code>\edlineref</code>	1, 48
<code>\edmakelabel</code>	1, 50
<code>\edpageref</code>	1, 48
<code>\edrowfill</code>	1, 58
<code>\EDTAB</code>	1
<code>\edtabcolsep</code>	1, 57
<code>\EDTABINDENT</code>	1
<code>\edtabindent</code>	1
<code>\EDTABtext</code>	1
<code>edtabularc</code> (environment)	57
<code>edtabularl</code> (environment)	57
<code>edtabularr</code> (environment)	57
<code>\EDTEXT</code>	1
<code>\edtext</code>	1, 23
<code>\edvertdots</code>	1, 59

<code>\edvertline</code>	1, 59
<code>\Eendnote</code>	24
<code>\Efootnote</code>	23
<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, 21
<code>\endminipage</code>	1
<code>\endnumbering</code>	1, 16
<code>\endpage@num</code>	1
<code>\endprint</code>	1
<code>\endquotation</code>	1
<code>\endquote</code>	1
<code>\endsub</code>	1, 20
<code>\endsubline@num</code>	1
environments:	
<code>edarrayc</code>	57
<code>edarrayl</code>	57
<code>edarrayr</code>	57
<code>edtabularc</code>	57
<code>edtabularl</code>	57
<code>edtabularr</code>	57
<code>ledgroup</code>	48
<code>ledgroupsize</code>	48
<code>minipage</code>	48
<code>Euclid</code>	12
<code>\extensionchars</code>	1, 62

F

<code>\f@x@l@cks</code>	1
Fairbairns, Robin	31
<code>\first@linenum@out@false</code>	1
<code>\first@linenum@out@true</code>	1
<code>\firstlinenum</code>	1, 19
<code>\firstseriesX@</code>	1
<code>\firstsublinenum</code>	1, 19

<code>\firstXseries@</code>	1
<code>\fix@page</code>	1
<code>\flag@end</code>	1
<code>\flag@start</code>	1
<code>\flagstanza</code>	1, 47
<code>\flush@notes</code>	1
<code>\fnpos</code>	1, 31
Folkerts, Menso	12
<code>\footfootmarkA</code>	30
<code>\footfudgefiddle</code>	1, 64
<code>\footnote</code>	1
<code>\footnoteA</code>	30
<code>\footnoteB</code>	30
<code>\footnoteC</code>	30
<code>\footnoteD</code>	30
<code>\footnoteE</code>	30
<code>\footnotelang@lua</code>	1
<code>\footnotelang@poly</code>	1
<code>\footnoteoptions@</code>	1
<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>	39
<code>\hidenumbering</code>	1, 22
<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>	41
<code>\hsizetwocolX</code>	41
<code>\Hy@raisedlink@left</code>	1
<code>\hyperlinkformat</code>	1

\hyperlinkformatR	1
\hyperlinkR	1

I

\if@addsw	1
\if@edindex@fornote@true	1
\if@eled@sectioning	1
\if@firstlineofpage	1
\if@led@nofoot	1
\if@ledgroup	1
\if@lemmacommand@	1
\if@msdata@insertedfrompreviouspage	1
\if@noeled@sec	1
\if@noneed@Footnote	1
\if@RTL	1
\ifautopar@pause	1
\ifbypage@	1
\ifbypage@R	1
\ifbypstart@	1
\ifbypstart@R	1
\ifeledmaccompat@	1
\iffirst@linenum@out@	1
\ifindtl@innote	1
\ifindtl@notenumber	1
\ifinserthangingsymbol	1
\ifistanza	1
\ifl@d@dash	1
\ifl@d@elin	1
\ifl@d@esl	1
\ifl@d@pnum	1
\ifl@d@ssub	1
\ifl@d@Xmorethantwolines	1
\ifl@d@Xtwolines	1
\ifl@dend@X	1
\ifl@dhidenumber	1
\ifl@dmemoir	1
\ifl@dpaging	1
\ifl@dpairing	1
\ifl@dprintingcolumns	1
\ifl@dprintingpages	1
\ifl@dskipnumber	1
\ifl@dskipversenumber	1
\ifl@dstartendok	1
\ifl@imakeidx	1
\ifl@indextools	1
\ifledfinal	1, 62
\ifledgroupnotesL@	1
\ifledgroupnotesR@	1
\iflednopbinverse	1
\ifledRcol	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>\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@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	148
---------------------------	-----

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@demptyd@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@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@dlabel@parse</code>	1
<code>\l@dld@ta</code>	1
<code>\l@dlp@rbox</code>	1
<code>\l@dlsn@te</code>	1
<code>\l@dlsnote</code>	1
<code>\l@dmake@labels</code>	1
<code>\l@dmodforedtext</code>	1
<code>\l@dnullfills</code>	1
<code>\l@dnumpstartsL</code>	1
<code>\l@doldold@footnotetext</code>	1
<code>\l@dp@rsefootspec</code>	1
<code>\l@dparsedendline</code>	1
<code>\l@dparsedendpage</code>	1
<code>\l@dparsedendsub</code>	1
<code>\l@dparsedstartline</code>	1
<code>\l@dparsedstartpage</code>	1
<code>\l@dparsedstartsub</code>	1
<code>\l@dpush@begins</code>	1
<code>\l@drd@ta</code>	1
<code>\l@dref@undefined</code>	1
<code>\l@drestorefills</code>	1
<code>\l@drestoreforedtext</code>	1

<code>\ledrp@rbox</code>	1
<code>\ledrsn@te</code>	1
<code>\ledrsnote</code>	1
<code>\ledsetmaxcolwidth</code>	1
<code>\ledskipnumberfalse</code>	1
<code>\ledskipnumbertrue</code>	1
<code>\ledtabaddcols</code>	1
<code>\ledtabnoexpands</code>	1
<code>\ledunboxmpfoot</code>	1
<code>\ledunhbox@line</code>	1
<code>\ledzeropenalties</code>	1
<code>\label</code>	50
<code>\labelpstartfalse</code>	1
<code>\labelpstarttrue</code>	1, 18
<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@Edtextoutsidepstart</code>	1
<code>\led@err@EdtextWithoutFootnote</code>	1
<code>\led@err@FootnoteWithoutEdtext</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@TooManyColumns</code>	1
<code>\led@err@UnequalColumns</code>	1
<code>\led@error@fail@patch@@doclearpage</code>	1
<code>\led@error@fail@patch@@iiiminipage</code>	1
<code>\led@error@fail@patch@@makecol</code>	1
<code>\led@error@fail@patch@@reinserts</code>	1
<code>\led@error@fail@patch@endminipage</code>	1
<code>\led@error@ImakeidxAfterEledmac</code>	1
<code>\led@error@IndextoolsAfterEledmac</code>	1
<code>\led@mess@NotesChanged</code>	1

\led@mess@SectionContinued	1
\led@nopb	1
\led@nopbnum	1
\led@pb	1
\led@pb@setting	1
\led@pbnum	1
\led@reinit@index@fornote	1
\led@set@index@fornote	1
\led@toksa	1
\led@toksb	1
\led@warn@AppLabelOutEdtext	1
\led@warn@BadAction	1
\led@warn@BadAdvancelineLine	1
\led@warn@BadAdvancelineSubline	1
\led@warn@BadLineation	1
\led@warn@BadLinenummargin	1
\led@warn@BadLockdisp	1
\led@warn@BadSetline	1
\led@warn@BadSetlinenum	1
\led@warn@BadSidenotemargin	1
\led@warn@BadSublockdisp	1
\led@warn@DuplicateLabel	1
\led@warn@LineFileObsolete	1
\led@warn@NoIndexFile	1
\led@warn@NoLineFile	1
\led@warn@NoMarginpars	1
\led@warn@RefUndefined	1
\led@warn@SeriesStillExist	1
\led@warning@hsizeX@deprecated	1
\led@warning@msdatawithoutstop	1
\led@warning@Xhsize@deprecated	1
ledgroup (environment)	48
ledgroupsize (environment)	48
\ledinnernote	1, 53
\ledinnote	1
\ledinnotehyperpage	1
\ledinnotemark	1
\ledleftnote	1, 53
\ledlinenum	1
\ledllfill	1
\ledlsnotefontsetup	1, 54
\ledlsnotesep	1, 54
\ledlsnotewidth	1, 53
\lednopb	1, 61
\lednopbinversetrue	62
\lednopbnum	1
\ledouternote	53
\ledouterote	1
\ledpb	1, 61
\ledpbnum	1

<code>\ledpbsetting</code>	1, 62
<code>\ledrightnote</code>	1, 53
<code>\ledrlfill</code>	1
<code>\ledrsnotefontsetup</code>	1, 54
<code>\ledrsnotesep</code>	1, 54
<code>\ledrsnotewidth</code>	1, 53
<code>\ledsectnomark</code>	1
<code>\ledsectnotoc</code>	1
<code>\ledsetnormalparstuff@common</code>	1
<code>\ledsetnormalparstuffX</code>	1
<code>\ledsidenote</code>	1, 53
<code>\leftctab</code>	1
<code>\leftlinenum</code>	1, 20
<code>\leftltab</code>	1
<code>\leftnoteupfalse</code>	53
<code>\leftpstartnum</code>	1
<code>\leftrtab</code>	1
Leibniz	13
<code>\lemma</code>	1, 25
<code>\letsforverteilen</code>	1
<code>\line@list</code>	1
<code>\line@list@stuff</code>	1
<code>\line@list@version</code>	1
<code>\line@margin</code>	1
<code>\line@num</code>	1
<code>\line@set</code>	1
<code>\lineation</code>	1, 20
<code>\linenum</code>	1, 25
<code>\linenum@out</code>	1
<code>\linenumberlist</code>	1, 19
<code>\linenumberstyle</code>	1, 21
<code>\linenumincrement</code>	1, 19
<code>\linenummargin</code>	1, 20
<code>\linenumr@p</code>	1
<code>\linenumrep</code>	1
<code>\linenumsep</code>	1, 20
<code>\linerrangesep@</code>	1
<code>\list@clear</code>	1
<code>\list@clearing@reg</code>	1
<code>\list@create</code>	1
<code>\lock@disp</code>	1
<code>\lock@off</code>	1
<code>\lock@on</code>	1
<code>\lockdisp</code>	1, 21
Lorch, Richard	12
<code>\ltab</code>	1
<code>\ltabtext</code>	1
Luecking, Dan	68

M

<code>\m@mmf@check</code>	1
<code>\m@mmf@prepare</code>	1
<code>\M@ssect</code>	1
<code>\makehboxofhboxes</code>	1
<code>\managestanza@modulo</code>	1
<code>\maxhnotesX</code>	42
Mayer, Gyula	13
<code>\measurembody</code>	1
<code>\measuremcell</code>	1
<code>\measuremrow</code>	1
<code>\measuretbody</code>	1
<code>\measuretcell</code>	1
<code>\measuretrow</code>	1
Middleton, Thomas	13, 90
<code>minipage</code> (environment)	48
Mittelbach, Frank	12
<code>\morenoexpands</code>	1, 64
<code>\mpfnpos</code>	1, 31
<code>\mpnormalfootgroup</code>	1
<code>\mpnormalfootgroupX</code>	1
<code>\mpnormalvfootnote</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, 29
<code>\multfootsep</code>	1, 31
<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>\no@expands</code>	1
<code>\noeledsec</code>	61
<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>\normalvfootnoteX</code>	1
<code>\notefontsizeX</code>	39
<code>\notenumfontX</code>	38
<code>\noteschanged@false</code>	1
<code>\noteschanged@true</code>	1
<code>\noteswidthliketwocolumnsX</code>	43
<code>\nulledindex</code>	1
<code>\nullsetzen</code>	1
<code>\num@lines</code>	1
<code>\numberedpar@false</code>	1
<code>\numberedpar@true</code>	1
<code>\numberingfalse</code>	1
<code>\numberingtrue</code>	1
<code>\numberlinefalse</code>	19
<code>\numberlinetrue</code>	19
<code>\numberpstartfalse</code>	1, 18
<code>\numberpstarttrue</code>	1, 18
<code>\numberstanzafalse</code>	47
<code>\numberstanzatrue</code>	47
<code>\numlabfont</code>	1, 44

O

<code>\old@hsize</code>	1
<code>\one@line</code>	1
<code>optioncontinuousnumberingwithcolumns</code>	394
<code>optioninnnote</code>	393
<code>optioninnnote</code>	393
<code>optionlinrangesep</code>	232
<code>optionnocritical</code>	393
<code>optionnoend</code>	393

optionnopenalties	64
optionnotenumber	393

P

\page@action	1
\page@num	1
\pagelinesep	1, 55
\pageno	1
\pageref	50
\par@line	1
\para@footgroupX	1
\para@footsetup	1
\para@footsetupX	1
\para@vfootnoteX	1
\parafootfmt	1
\parafootfmtX	1
\parafootgroup	1
\parafootsepX	41
\parafootstart	1
\parafootstartX	1
\paravfootnote	1
\parindentX	39
\pausenumbering	1, 18
\pend	1, 16
Plato of Tivoli	12
\postbodyfootmark	1
\prebodyfootmark	1
\prenotesX	42
\prepare@edindex@fornote	1
\prepare@prenotesX	1
\prepare@preXnotes	1
\prev@nopb	1
\prev@pb	1
\prevpage@num	1
\preXnotes	1, 42
\preXnotes@	1
\print@eledsection	1
\print@footnoteXrule	1
\print@leftmargin@eledsection	1
\print@lemma	1
\print@line	1
\print@notesX	1
\print@rightmargin@eledsection	1
\print@Xfootnoterule	1
\print@Xnotes	1
\printendlines	1
\printlineendnote	1
\printlineendnotearea	1
\printlinefootnote	1
\printlinefootnotearea	1

<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, 16
<code>\pstarteref</code>	1
<code>\pstartnum</code>	1
<code>\pstartref</code>	48

Q

<code>\quotation</code>	1
<code>\quote</code>	1

R

<code>\raggedX</code>	41
<code>\raw@text</code>	1
<code>\rbracket</code>	1
<code>\read@linelist</code>	1
<code>\ref</code>	50
<code>\Relax</code>	1
<code>\reledmac@error</code>	1
<code>\reledmac@warning</code>	1
<code>\removehboxes</code>	1
<code>\reset@msd@options@iffullpage</code>	1
<code>\resetprevline@</code>	1, 93
<code>\resetprevpage@</code>	1
<code>\resetprevpage@num</code>	93
<code>\restore@familiarnotes</code>	1
<code>\restore@notes</code>	1
<code>\restore@sidenotes</code>	1
<code>\resumenumbering</code>	1, 18
<code>\rightctab</code>	1
<code>\rightlinenum</code>	1, 20
<code>\rightltab</code>	1
<code>\rightnoteupfalse</code>	53
<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

<code>Sacrobosco</code>	13
<code>\sameword</code>	1, 26
<code>\sameword@inedtext</code>	1

Schöpf, Rainer	12
\section@num	1
\select@lemmafont	1
\select@lemmafont	1, 44
\Seref	1, 50
\Serefonlypage	50
\Serefwithpage	1, 50
\series	1
\seriesatbegin	1, 31
\seriesatend	1, 31
\set@line	1
\set@line@action	1
\set@Xtxtbeforenotes	1
\setapprefprefixmore	51
\setapprefprefixsingle	51
\setcommand@series	1
\sethangingsymbol	1, 46
\setistwofollowinglines	1
\setl@dlp@rbox	1
\setl@drpr@box	1
\setline	1, 21
\setlinenum	1, 21
\setmcellcenter	1
\setmcellleft	1
\setmcellright	1
\setmrowcenter	1
\setmrowleft	1
\setmrowright	1
\setmsdatalabel	1, 30
\setmsdataseries	1, 30
\setnoteswidthliketwocolumnsX@	1
\setnotesXpositionliketwocolumns@	1
\setprintendlines	1
\setprintlines	1
\setSErefonlypageprefixmore	51
\setSErefonlypageprefixsingle	51
\setSErefprefixmore	51
\setSErefprefixsingle	51
\setsidenotesep	54
\setstanzaindents	1, 44
\setstanzapenalties	1, 46
\setstanzavalues	1
\settccllcenter	1
\settccllleft	1
\settccllright	1
\settoggle@series	1
\settrowcenter	1
\settrowleft	1
\settrowright	1
\setXnotespositionliketwocolumns@	1

<code>\setXnoteswidthliketwocolumns@</code>	1
<code>\showlemma</code>	1, 62
<code>\showwordrank</code>	1, 28
<code>\sidenote@margin</code>	1
<code>\sidenotemargin</code>	1, 53
<code>\sidepstartnumtrue</code>	18
<code>\skip@lockoff</code>	1
<code>\skipnumbering</code>	1, 22
<code>\splitoff</code>	1
<code>\spreadmath</code>	1, 58
<code>\spreadtext</code>	1, 58
<code>\stanza</code>	1, 44
<code>\stanza@count</code>	1
<code>\stanza@hang</code>	1
<code>\stanza@line</code>	1
<code>\stanzaindent</code>	1, 45
<code>\stanzaindent*</code>	1, 45
<code>\stanzaindentbase</code>	1, 44
<code>\stanzanumwrapper</code>	1, 47
<code>\startlock</code>	1, 21
<code>\startsub</code>	1, 20
<code>\stepl@dcolcount</code>	1
<code>\stopmsdata</code>	1, 29
<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, 21
<code>\sublinenumincrement</code>	1, 19
<code>\sublinenumr@p</code>	1
<code>\sublinenumrep</code>	1
<code>\sublineref</code>	1, 48
<code>\sublines@false</code>	1
<code>\sublines@true</code>	1
<code>\sublock@disp</code>	1
<code>\sublockdisp</code>	1
Sullivan, Wayne	12, 13, 64, 76, 80, 159, 160, 238, 272
<code>\sza@penalty</code>	1

T

<code>\tabHilfbox</code>	1
<code>\tabhilfbox</code>	1
<code>\theadcolcount</code>	1
<code>\theadtext</code>	1
<code>\theendpageline</code>	1
<code>\thefootnoteA</code>	30
Theodosius	13
<code>\thepageline</code>	1

\thepstart	1, 18
\thestanza	1, 47
\thestartpageline	1
\this@line@list@version	1
\threecolfootfmt	1
\threecolfootfmtX	1
\threecolfootgroup	1
\threecolfootgroupX	1
\threecolfootsetup	1
\threecolfootsetupX	1
\threecolvfootnote	1
\threecolvfootnoteX	1
\twocolfootfmt	1
\twocolfootfmtX	1
\twocolfootgroup	1
\twocolfootgroupX	1
\twocolfootsetup	1
\twocolfootsetupX	1
\twocolvfootnote	1
\twocolvfootnoteX	1

U

\unvxhX	1
---------------	---

V

Vamana	13
\variab	1
\vbfnoteX	1
\vl@dbfnote	1
\vl@dcsnote	1
\vl@dlsnote	1
\vl@drsnote	1
\vnumfootnoteX	1

W

Whitney, Ron	12
\widthX	43
\wrap@edcrossref	1
\wrapcontentX	39
\wrapped@bodyfootmarkX	1
\wrapped@footfootmarkX	1
Wujastyk, Dominik	12

X

\X@doreinfeet	1
\Xafterlemmaseparator	37
\Xafternote	41
\Xafternumber	35
\Xafterrule	42
\Xaftersymlinenum	35

\Xarrangement	1, 32
\Xarrangement@normal	1
\Xarrangement@paragraph	1
\Xarrangement@threecol	1
\Xarrangement@twocol	1
\Xbeforeinserting	40
\Xbeforelemmaseparator	37
\Xbeforenotes	42
\Xbeforenumber	33, 35
\Xbeforesymlinenum	35
\Xbhookgroup	41
\Xbhooknote	40
\Xboxlinenum	36
\Xboxlinenumalign	36
\Xboxsymlinenum	36
\Xcolalign	40
\Xdo@feet	1
\xedindex	1
\xedlabel	1
\xedtext	1
\Xendaftererenum	35
\Xendafterlemmaseparator	38
\Xendafternote	43
\Xendafterpagenumber	37
\Xendaftersymlinenum	35
\Xendahookinplaceofnumber	37
\Xendahooklinenumber	37
\Xendbeforelemmaseparator	38
\Xendbeforenumber	35
\Xendbeforepagenumber	37
\Xendbeforesymlinenum	35
\Xendbhookinplaceofnumber	37
\Xendbhooklinenumber	37
\Xendbhooknote	40
\Xendboxendlinenumalign	36
\Xendboxlinenum	36
\Xendboxlinenumalign	36
\Xendboxstartlinenumalign	36
\Xendboxsymlinenum	36
\Xendhangindent	39
\Xendinplaceoflemmaseparator	38
\Xendinplaceofnumber	36
\Xendlemmadisablefontselection	38
\Xendlemmafont	38
\Xendlemmaseparator	38
\Xendlineprefixmore	37
\Xendlineprefixsingle	37
\Xendlinerangeseparator	33
\Xendmorethantwolines	34
\Xendnonumber	34

\Xendnotefontsize	39
\Xendnotenumfont	38
\Xendnumberonlyfirstinline	33
\Xendnumberonlyfirstintwoline	33
\Xendparagraph	43
\Xendsep	43
\Xendsublinesep	35
\Xendsymlinenum	33
\Xendtwoline	34
\Xendtwolinebutnotmore	34
\Xendwrapcontent	39
\xflagref	1
\Xhangindent	39
\Xhsizethreecol	41
\Xhsizetwocol	41
\Xinplaceoflemmaseparator	37
\Xinplaceofnumber	36
\Xinsertparafootsep	1
\Xledsetnormalparstuff	1
\xleft@appenditem	1
\Xlemmadisablefontselection	38
\Xlemmafont	38
\Xlemmaseparator	37
\Xlinrangeseparator	33
\xlineref	1, 49
\Xmaxhnotes	42
\Xmorethantwoline	33
\Xnolemmaseparator	1, 37
\Xnonbreakableafternumber	35
\Xnonumber	34
\Xnotefontsize	39
\Xnotenumfont	38
\Xnoteswidthliketwocolumns	43
\Xnumberonlyfirstinline	33
\Xnumberonlyfirstintwoline	33
\Xonlypstart	35
\xpagerref	1, 49
\Xparafootsep	41
\Xparindent	39
\Xpstart	34
\Xpstarteverytime	34
\xpstartref	1, 49
\XR@test	1
\XR@test@mac	1
\XR@test@mac@test	1
\Xragged	41
\xright@appenditem	1
\Xrigidbalance	1
\Xstanza	35
\Xstanzaseparator	35

<code>\xsublineref</code>	<u>1</u> , 49
<code>\Xsublinesep</code>	35
<code>\Xsymlinenum</code>	33
<code>\Xtwolines</code>	33
<code>\Xtwolinesonlyinsamepage</code>	34
<code>\Xtxtbeforenotes</code>	41
<code>\Xunvxh</code>	<u>1</u>
<code>\Xwidth</code>	43
<code>\Xwrapcontent</code>	39
<code>\Xwrapendlemma</code>	39
<code>\Xwraplemma</code>	39
<code>\xxref</code>	<u>1</u> , 50

Z

<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	69
\morenoexpands: Added \l@dtabnoexpands to \no@expands	115
\reledmac@error: Added \eledmac@error and replaced error messages	70
General: Added tabmac code, and extended indexing	1
v0.2.1.	
\@lab: Removed page setting from \@lab	240
\doxtrafeet: Renamed \doxtrafeet to \l@ddoxtrafeet	233
\edlabel: Tweaked \edlabel to get correct page numbers	238
\l@ddodoreinxtrafeet: Renamed \dodoreinxtrafeet to \l@ddodoreinxtrafeet	234
\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	115
\zz@@@: Minor change to \zz@@@	238
General: Added text about normal labeling	50
Bug fixes and match with mempatch v1.8	1
Major changes to insert code when memoir is loaded	235
v0.2.2.	
\footfudgefiddle: Added \footfudgefiddle	157
\line@list@stuff: Added initial write of page number in \line@list@stuff	108
\para@footsetup: Added \footfudgefiddle to \para@footsetup	158
\para@footsetupX: Added \footfudgefiddle to \para@footsetupX	195
General: Improved paragraph footnotes	1
New Dekker example	1
Used \providecommand for \@gobblethree to avoid clash with the amsfonts package	76
v0.3.0.	
\@lab: Replaced \the\line@num by \linenumr@p\line@num in \@lab, and similar for sub-lines	240
\@nl@reg: Added a bunch of code to \@nl for handling \setlinenum	97
\ledlinenum: Added \linenumr@p and \sublinenum@rep to \leftlinenum and \rightlinenum	88
\linenumberlist: Added \linenumberlist mechanism	76
\printendlines: Added \linenumr@p and \sublinenumr@p to \printendlines	211
\printlines: Added \linenumr@p and \sublinenumr@p to \printlines	179
\sublinenumr@p: Added \linenumberstyle and \sublinenumberstyle	88
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	259
\Xarrangement@normal: Added minpage footnote setup to \footnormal	153
\edtext: Added \showlemma to \edtext	116
\l@dfeetendmini: Added \l@dfeetbeginmini, \l@dfeetendmini and all their supporting code	258

\mpnormalfootgroup: Added \mpnormalfootgroup	156
\mpnormalvfootnote: Added \mpnormalvfootnote	154
\showlemma: Added \showlemma	76
General: Added final/draft options	67
Added ledgroup environment	261
Added ledgroupsize environment	261
Added minipage, etc., support	1
v0.4.1.	
\Xdo@feet: Changed \Xdo@feet code for easier extensions	233
\edindex: Let eledmac take advantage of memoir's indexing	266
\print@Xnotes: Added \@opXfeet	233
General: Added code for changing \@doclearpage	236
Not released. Minor editorial improvements and code tweaks	1
Only change \@footnotetext and \@footnotemark if memoir not used	181
v0.5.0.	
\@footnotetext: Enabled regular \footnote in numbered text	182
\@xympar: Eliminated \marginpar disturbance	251
General: Added left and right side notes	251
Added sidenotes, familiar footnotes in numbered text	1
v0.5.1.	
\affixline@num: Changed \affixline@num to cater for sidenotes	140
\l@getsidenote@margin: Added \sidenotemargin and \sidenote@margin	252
General: Added moveable side note	251
Fixed right line numbers killed in v0.5	1
Only change \hsize in ledgroupsize environment otherwise page number can be in wrong place	261
v0.6.0.	
\@lopR: Added \@pend, \@pendR, \@lopL and \@lopR in anticipation of parallel processing	99
\@nl@reg: Added \fix@page to \@nl	97
Extended \@nl to include the page number	97
\fix@page: Added \last@page@num and \fix@page	98
\get@thisfootnote: Changed \l@dbfnote and \v1@dbfnote as originals could give incorrect markers in the footnotes	182
\new@line: Extended \new@line to output page numbers	109
General: Fixed long paragraphs looping	1
Fixed minor typos	1
Prepared for eledpar package	1
v0.7.0.	
\@nl@reg: Added \@nl@reg	97
\@ref@reg: Added \@ref@reg	105
\affixline@num: Added skipnumering to \affixline@num	140
\do@actions@fixedcode: Added \do@actions@fixedcode	139
\do@actions@next: Added number skipping to \do@actions	138
\do@insidelinehook: Added \do@linehook for use in \do@line	136
\endnumbering: Changed \endnumbering for eledpar	80
\fix@l@cks: Added \ch@cksub@l@ck, \ch@ck@l@ck and \fix@l@cks	143
\footplitskips: Added \footplitskips for use in many footnote styles	152
\get@linelistfile: Added \get@linelistfile	95
\initnumbering@reg: Added \initnumbering@reg	78

\l@advance@parledgroup@beforenormalnotes: Added \l@dunboxmpfoot containing some common code	260
\l@dcstotetext@r: Added \l@emptyd@ta	136
\l@dgetline@margin: Added \l@dgetline@margin	84
\l@dgetlock@disp: Added \l@dgetlock@disp	87
\l@dgetsidenote@margin: Added \l@dgetsidenote@margin	252
\l@dnumpstartsL: Added \l@dnumpstartsL, \ifl@dpairing and \ifpst@rted for/from eledpar	77
\l@drsn@te: Added \l@dlsn@te and \l@drsn@te for use in \do@line	136
\l@dzeropenalties: Added \l@dzeropenalties	131
\ledlinenum: Added \ledlinenum for use by \leftlinenum and \rightlinenum	88
\line@list@stuff: Deleted \page@start from \line@list@stuff	108
\list@clearing@reg: Added \list@clearing@reg	95
\n@num: Added \n@num	104
\normalbfnoteX: Removed extraneous space from \normalbfnoteX	187
\resumenumbering: Changed \resumenumbering for eledpar	81
\setprintendlines: Added \setprintendlines for use by \printendlines	209
\setprintlines: Added \setprintlines for use by \printlines	176
\skipnumbering: Added \skipnumbering and supports	112
\sublinenumincrement: Added \firstlinenum, \linenumincrement, \firstsublinenum and \linenumincrement	86
\sublinenumr@p: Using \linenumrep instead of \linenumr@p	88
Using \sublinenumrep instead of \sublinenumr@p	88
\vnumfootnoteX: Removed extraneous space from \vnumfootnoteX	189
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	70
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 use 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	77
General: For compatibility with eledpar, possibility to use \autopar on the right side.	1
Possibility to number \pstart.	18

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.	274
General: New <code>stanzaindentsrepetition</code> counter to repeat stanza indents every n verses.	45
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.	238
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.	140
<code>\ifinserthangingsymbol</code> : New management of <code>hangingsymbol</code> insertion, preventing undesirable insertions.	273
<code>\printlines</code> : Line numbering can be reset at each <code>pstart</code> .	178
General: Line numbering can be reset at each <code>pstart</code> .	83
Possibility to print <code>\pstart</code> number inside.	18
v0.17.0.	
<code>\ifinserthangingsymbol</code> : New new management of <code>hangingsymbol</code> insertion, preventing undesirable insertions.	273
v1.0.0.	
<code>\morenoexpands</code> : Change to be compatible with new features	115
General: <code>\lemma</code> can contain commands.	25
Debug in lineation command	20
New generic commands to customize footnote display.	32, 224
Options <code>nonum</code> and <code>nosep</code> in <code>\Xfootnote</code> .	23
Options of <code>\Xfootnotes</code> .	149
Possibility to have commands in sidenotes.	53
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.	33
v1.1.0.	
<code>\preXnotes</code> : New skip <code>\preXnotes@</code>	203
<code>\settoggle@series</code> : <code>\settoggle@series</code> switch the global value of the toggle, not only the local value.	224
General: Add <code>\labelpstarttrue</code> .	18
Add <code>\Xnumberonlyfirstintwolines</code>	33
Add <code>\Xpstart</code> and <code>\Xonlypstart</code>	34
New hook to add arbitrary code at the beginning of the notes	40
New options for block of notes.	41
New package option: <code>parapparatus</code> .	1
New tools to change order of series	223
Sectioning commands.	60

v1.2.0.	
\endquote:	Compatibility of \ledchapter with the <i>memoir</i> class. 309
\preXnotes:	Debug in familiar footnotes (bug introduced by v1.1). 203
v1.3.0.	
\endquote:	<i>Quotation</i> and quote environment inside numbered sections. 309
v1.4.0.	
\edtext:	Compatibility of \edtext with the right-to-left direction (with Polyglossia). 116
\ledsetnormalparstuffX:	Direction of footnotes with polyglossia. 200
\newseries@:	Remembers the language of the lemma, in order to create a correct direction for the footnote separator. 215
\rbracket:	Switch the right bracket to a left bracket when the lemma is RTL (needs polyglossia or LuaTeX). 170
General:	Compatibility with LuaTeX of RTL notes. 1
v1.4.1.	
\affixside@note:	Remove spurious spaces. 257
\endquote:	New option <i>noquotation</i> 309
\get@thisfootnote:	Compatibility of standard footnotes with eledmac when these footnotes contain any commands. 182
\labelrefsparsesubline:	Fix bug with \edlabel. 239
v1.4.2.	
General:	Debug with some special classes. 1
v1.4.3.	
General:	Add \Xnonbreakableaftersubline. 35
Spurious space after familiar footnotes. 1
v1.4.4.	
General:	Label inside familiar footnotes. 1
v1.4.5.	
General:	Bug with komasscript + eledpar + chapter. 1
v1.4.6.	
General:	Bug with memoir class introduced by 1.4.5. 1
v1.4.7.	
\endquote:	Compatibility of sectioning commands with \autopar. 309
v1.4.8.	
General:	Corrects a bug with parallel texts introduced by 1.1. 1
v1.4.9.	
\normalbfnoteX:	Allow to redefine \thefootnoteX with alph when some packages are loaded. 187
v1.5.0.	
\do@insidelinehook:	Added \do@insidelinehook for use in \do@line 136
\edindex:	Compatibility with imakeidx package, and possibility to use multiple index with \edindex. 266
General:	Correct indexing when the call is made in critical notes. 262
v1.5.1.	
\managestanza@modulo:	Correct stanzaindentsrepetition counter 274
\normalvfootnoteX:	Fix bug with normal familiar footnotes when mixing RTL and LTR text. 185
v1.6.0.	
\newverse:	Add \falseverse macro. 277
v1.6.1.	
\AtEveryPstart:	Spurious space in \pstart. 128

\ifinserthangingsymbol: Hang verse is now not automatically flush right.	273
\l@dunhbox@line: Move the call to \inserthangingsymbol to allow use \hfill inside.	133
\pend: Spurious space in \pend.	129
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.	61
v1.8.0.	
\endquote: Correction of sectioning commands in parallel texts.	309
\get@index@command: Debug \get@index@command and compatibility with hyperref package.	265
\newhookcommand@series@reload: Debug \beforenotesX and \maxnotesX which did not work.	227
\prevpage@num: Correct \parafootsep when using with ledgroup.	163
General: Compatibility with parledgroup option of eldpar package.	1
If imakeidx and hyperref are loaded, adds hyperref in the index.	262
v1.8.1.	
General: Debug endnotes when more than one series is used (change the position where tools for endnotes are defined).	205
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 eldpar.	262
v1.9.0.	
\doxtrafeet: Add \fnpos to choice the order of footnotes.	233
\l@dfeetendmini: Add \mpfnpos to choice the order of footnotes in minipage / ledgroup.	258
v1.10.0.	
\endquote: Correction of sectioning commands in parallel texts.	309
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.	38
v1.11.1.	
General: Correct a bug when a critical note starts with plus or minus.	1

v1.12.0.	
\@nl@reg: To ensure compatibility with \musixtex, \@l becomes \@l. Consequently, \@l@reg becomes \@nl@reg.	97
\AtEveryPstart: New optional argument for \pstart, to execute code before it. . . .	128
\edindex: Use correctly default index when imakeidx is loaded.	266
\endquote: \ledxxx sectioning commands are deprecated and replaced by \eledxxx commands.	309
\initnumbering@reg: \beginnumbering is defined only on eledmac, not on eledpar. . .	78
\l@dcnote: \l@dlsnote, \l@drsnote and \l@dcnote defined only one time, in eledmac, including needs for eledpar case.	253
\l@dgetsidenote@margin: \sidenotemargin is now directly defined in eledmac to be able to manage eledpar.	252
\l@dnumpstartsL: Add \ifledRcol@ for eledpar	77
\l@dunhbox@line: \do@line is split in more little commands.	133
\newhookcommand@series@reload: Debug \beforenotesX and \maxhnotesX which did not work when called after \footparagraphX.	227
Debug \Xbeforenotes and \Xmaxhnotes which did not work when called after \footparagraph.	227
\pend: New optional argument for \pend, to execute code after it.	129
\stanza: &can have an optional argument: content to be printed after.	277
\Stanza can have an optional argument: content to be printed before.	277
Add \newverse macro, \falseverse deprecated.	277
General: Add \ledinnernote and \ledouternote commands.	53
Add \Xendparagraph and related settings.	43
Add hyperlink to crossref (needs hyperref package).	48
Compatibility with musixtex.	1
Debug eledmac sectioning command after using \resumenumbering.	1
Ensure that imakeidx is loaded <i>before</i> eledmac	262
New hooks: \Xafterrule and \afterruleX	42
New options for ragged-paragraph notes	41
New sectioning commands.	60
Optional arguments for \pstart and \pend.	17
v1.12.1.	
\wrap@edcrossref: Fix spurious spaces.	242
v1.12.2.	
\l@dunhbox@line: Fix a bug with critical notes at the tops of pages (added by v12.0.0)	133
v1.12.3.	
\flag@end: \flag@start and \flag@end are now defined only one time for eledmac and eledpar	109
\flag@start send a error message when a \edtext is done without insert (note)	109
\reledmac@error: Replaced error messages	70
General: Add macros for new messages since v0.7	70
Correct bug with side and familiar notes in tabular environments.	1
Debug \eledxxx with some paper size	1
Debug \ledinnernote and \ledouternote commands in the top of pages.	53
Debug left and right notes (bugs added by 1.12.0)	1
Underline lemma in \eledxxx when using draft mode.	1
v1.12.4.	
General: Debug spurious page breaks before \chapter (bug added in 1.12.0)	1

v1.12.5.	
\@edindex@hyperref: Debug \edindex when hyperref is not loaded	268
\@ssect: Debug \eledchapter in parallel with memoir	313
\doinsidelinehook: Added \dolinehook and \doinsidelinehook	136
\endnumbering: Allow to mix parallel columns and normal text when using	
\pausenumbering	80
\l@dgobblearg: \l@dgobblearg becomes \l@dgobbeloptarg	293
\l@drestoreforedtext: Debug optional arguments of \Xfootnote in tabular context	293
\resumenumbering: Debug \resumenumbering	81
v1.12.7.	
\wrap@edcrossref: \wrap@edcrossref is now robust	242
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	109
v1.13.0.	
\newhooktoggle@series@reload: Add \newhookcommand@toggle@reload	227
\para@footsetupX: In \para@footsetupX, use \columnwidth instead of \hsize	195
\settoggle@series: \settoggle@series can take an optional arguments to reload	
series setup.	224
General: Add \Xnoteswidthliketwocolumns and \noteswidthliketwocolumnsX	43
Added widthliketwocolumns option	67
v1.13.1.	
\thepstart: Add \l@dzeropenalties in \pstart	128
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@dpadding for \eledpar	77
General: Fix 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 bug with index when memoir class is used without hyperref	1
v1.14.0.	
\edindex: Let \eledmac take advantage of \makeidx even when memoir class is used	266
General: Debug spurious characters before endnotes.	205
Delete previous override of \l@d@@wrindexhyp at the beginning of a document	
when hyperref is not loaded.	269
Move gobbling command	76
Provide \@gobblefour	76
v1.14.1.	
\@ssect: Debug sectioning commands when using both handout and hyperref	
package.	316
v1.14.2.	
\@ssect: Debug \edtext after starred sectioning commands when using memoir class.	313
v1.15.0.	
\@edtext@level: New boolean \if@edtext@.	115
\arrangementX@threecol: Correct bug with paragraphed familiar footnotes setting.	194
\endsub: Restore subline feature (disabled by mistake in v1.8.0).	110
\if@lemmacommand@: New boolean \iflemmacommand@.	120

General: Fix bug with footnotes layout when using some options of the geometry package (bug add by v1.13.0).	1
New commands <code>\AtEveryPstart</code> and <code>\AtEveryPend</code>	18
New tools to prevent ambiguous references in lemma	26
v1.15.1.	
<code>\line@list@stuff</code> : Revert modification of 1.5.2 which makes bug with numbering. Leave vertical mode to solve spurious space before minipage.	108
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.	115
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.	243
v1.17.0.	
<code>\edtext</code> : Error message when calling <code>\edtext</code> outside of a numbered paragraph. . . .	116
v1.18.0.	
<code>\@edindex@hyperref</code> : Fix spurious space with <code>\edindex</code> when using <code>imakeidx/indextools + hyperref</code>	268
<code>\edlabel</code> : <code>\edlabel</code> is now defined only one time for both <code>eledmac</code> and <code>eledpar</code> . . .	238
<code>\l@d@section</code> : Option <code>parapparatus</code> works for endnotes.	205
<code>\l@dnumppstartsl</code> : Add <code>\ifl@dprintingpages</code> and <code>\@dprintingcolumns</code> for <code>eledpar</code>	77
<code>\print@line</code> : Compatibility with Lua \TeX RTL languages.	133
<code>\printlinefootnote</code> : Code refactoring in <code>\printlinefootnote</code> : the printing of the numbers are factorized in <code>\printlinefootnotearea</code>	172
<code>\printpstart</code> : Debug <code>\Xpstart</code> with parallel pages and columns (<code>eledpar</code>)	170
General: Add <code>\Xpstarteverytime</code>	34
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.	34
v1.19.0.	
<code>\footssplitskips</code> : <code>\footssplitskips</code> doesn't set <code>\floatingpenalty</code> to <code>\@MM</code> when processing parallel pages.	152
<code>\xxref</code> : <code>\xxref</code> works also with right side numbers, when <code>\@Rlineflag</code> is not empty. .	244
General: <code>\Xmaxhnotes</code> and <code>\maxhnotesX</code> work now for both two-columns and three-columns setting.	1
Compatibility with <code>eledpar</code> v1.13.0.	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>	179
<code>\stanza</code> : <code>&</code> and <code>&</code> can be preceded by spaces.	277
<code>\xxref</code> : Debug <code>\xxref</code> when not loading <code>eledpar</code> (fix bug added in 1.19.0).	244
General: Add <code>\Xendboxlinenum</code>	36
Add <code>\Xtwolines</code> and <code>\Xmorethantwolines</code> hooks	33
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 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>	23
The <code>\newifs</code> are not followed by boolean values set to false, 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>	268
<code>\l@d@section</code> : <code>\endnotes</code> take five arguments.	205
<code>\ledinnotemark</code> : Add <code>\ledinnotemark</code>	265
<code>\ledsetnormalparstuffX</code> : <code>\ledsetnormalparstuff</code> is deprecated and becomes <code>\ledsetnormalparstuffX</code> and <code>\Xledsetnormalparstuff</code>	200
<code>\n@num</code> : <code>\n@num@ref</code> deleted	104
<code>\n@num</code> defined only one time for both <code>Eledmac</code> and <code>Eledpar</code>	104
<code>\newhookcommand@series</code> : <code>\newhookcommand@series</code> can take an optional argument.	226
<code>\newhooktoggle@series</code> : <code>\newhooktoggle@series</code> can take an optional argument.	227
<code>\print@footnoterule</code> : Code refactoring: the spaces after the footnote rules are directly managed in <code>\print@Xfootnoterule</code> and <code>\print@footnoteruleX</code>	202
<code>\seriesatend</code> : Fix spurious space in <code>\seriesatend</code>	223
<code>\skipnumbering</code> : <code>\skipnumbering</code> defined only one time for both <code>Eledmac</code> and <code>Eledpar</code>	112
Correct <code>\skipnumbering</code> for stanza.	112
Delete <code>\skipnumbering@reg</code>	112
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	223
Add <code>\applabel</code> and related	51
Add <code>\beforenotesX</code> and <code>\Xbeforenotes</code> features for notes set in two and three column.	1
Add <code>\hidenumbering</code>	22
Add <code>\Xcolalign</code> and <code>\colalignX</code> hooks	40
Add <code>\Xendtwolines</code> , <code>\Xendmorethantwolines</code> , <code>\Xendtwolinesbutnotmore</code> and <code>\Xendtwolinesonlyinsamepage</code>	34
Add <code>\Xparindent</code> and <code>\hangindentX</code>	39
Add <code>\Xtwolinesbutnotmore</code> and <code>\Xtwolinesonlyinsamepage</code>	1
Add <code>nocritical</code> , <code>noend</code> , <code>nofamiliar</code> and <code>noledgroup</code> options.	1
Add <code>noeledsec</code> package option	1
Debug <code>\beforenotesX</code> <code>\maxhnotesX</code> <code>\noteswidthliketwocolumnsX</code> and <code>\afterruleX</code> with footnotes set in two and three columns.	1
Fix 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 bug with <code>\maxnotesX</code> when using <code>\foottwocolX</code> or <code>\footthreecolX</code>	1
Fix bug with space between columns with notes in two columns (bug added in v1.13.0).	1
Fix spurious space after first page number in <code>\doendnotes</code> . <code>oldprintnpnumspace</code> option allows to come back to previous setting	1
<code>parapparatus</code> option works now with familiar footnotes.	1
Provide <code>\@gobblefive</code>	76
v1.22.0.	
<code>\ledinnote</code> : <code>\ledinnote</code> takes a first optional argument, which is the label for hyperlinks.	265
General: Add <code>\doendnotesbysection</code> command.	24
Add option for lemma separator inside endnotes	38
Adds hyperlink for references to notes in indices.	1
Fix conflict between <code>noend</code> package option and <code>edtabularx</code> environments	1
Provides support for <code>xindy</code>	1
Standardize endnotes handbook.	24
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>	163
General: Fix 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>	115
<code>\Serefwithpage</code> : Debug <code>\Xendtwolines</code> , <code>\Xendmoreethantwolines</code> , <code>\Xendtwolinesbutnotmore</code> and <code>\Xendtwolinesonlyinsamepage</code> when using <code>\apprefwithpage</code>	247
<code>\lemma</code> : Fix spurious space after <code>\lemma</code> command	120
<code>\newseries@</code> : Prevent spurious spaces when <code>\Afootnote</code> and similar commands are followed by spaces (bug added on 1.0.0).	215
<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.	124
General: Add <code>\Xboxlinenumalign</code> and <code>\Xendboxlinenumalign</code>	36
Add <code>\Xboxstartlinenum</code> , <code>\Xendboxstartlinenum</code> , <code>\Xboxendlinenum</code> , <code>\Xendboxendlinenum</code>	36
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 bug with <code>\sameword</code> when the lemma overlaps multiple line.	26
Fix bug with <code>\sameword</code> when the same lemma is used for multiple notes or for nested <code>\edtexts</code>	26
Fix 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.	26
v1.23.1.	
General: Fix bug with <code>\lemma</code> command in the right side.	1
v1.23.2.	
General: Compatibility with \LaTeX 's release 2015.	1

v1.24.0.	
General: We can reinitialize <code>\AtEveryPstart</code> and <code>\AtEveryPend</code> providing to it an empty argument.	1
v1.24.1.	
General: <code>\lemma</code> is disabled when using ‘ <code>nocritical</code> ’ option.	1
v1.24.2.	
General: Fix incompatibility between ‘ <code>nofamiliar</code> ’ option and ‘ <code>memoir</code> ’ package.	1
v1.24.3.	
General: Restore marginal numbers and notes with sectioning command (bug introduced in v1.21.0)	1
v1.24.4.	
General: Fix spurious space with <code>\edindex</code> when using <code>xindy+hyperref</code> option.	1
v1.24.5.	
General: Fix bug of indent, when a added in 1.1.0, when a <code>\beginnumbering</code> immediately follow a sectioning command.	1
v2.0.0.	
<code>\@iiiminipage</code> : Patch <code>\@iiiminipage</code> instead of redefining it.	259
<code>\@xympar</code> : Patching <code>\@xympar</code> instead of redefining it	251
<code>\endminipage</code> : Patch <code>\endminipage</code> instead of redefining it.	260
<code>\initnumbering@quote</code> : <code>\initnumbering@sectcmd</code> becomes <code>\initnumbering@quote</code>	309
<code>\l@advance@parledgroup@beforenormalnotes</code> : Some code of <code>\l@dumboxmpfoot</code> moved to <code>\l@advance@parledegroupp@beforenormalnotes</code>	260
<code>\newseries@</code> : One endnotes file by series.	220
General: <code>\@makecol</code> , <code>\@reinserts</code> and <code>\@doclearpage</code> are patched instead of begin redefined	235
<code>\doxtrafeeti</code> becomes <code>\do@feetX</code> ; <code>\doxtrafeetii</code> becomes <code>\Xdo@feet</code> ; <code>\@opxtrafeeti</code> becomes <code>\@opfeetX</code> ; <code>\doreinxtrafeetii</code> becomes <code>\Xdoreinfeet</code> ; <code>\doreinxtrafeeti</code> becomes <code>\@doreinfeetX</code>	235
Add <code>\Xendinplaceofnumber</code> hook.	1
Add <code>\Xendnonumber</code> hook.	1
Add <code>nonum</code> option for endnotes.	1
Fix bug when printing only one series of endnotes, but wanted to keep endnotes for other series.	1
In order to have a more consistent name’s convention, many names has been changed.	1
Many \TeX ’s output macros are now patched and not override.	1
Package’s name becomes <code>reledmac</code>	1
Patch <code>\@footnotemark</code> instead of redefine it	181
Suppress indexing command specific to <code>memoir</code>	266
v2.0.1.	
General: Fix bug in <code>eledmac-compat</code> option	1
Fix incompatibility between optional argument of <code>\pstart</code> and <code>\numberpstarttrue</code>	1
v2.1.0.	
General: Fix bug with <code>\advanceline</code> at the beginning of a <code>\pstart</code>	1
Fix bug with <code>\chapter</code> in optional argument of <code>\pstart</code> in parallel typesetting with <code>scrbook</code>	1
Fix bug with <code>\eledchapter</code> in parallel typesetting with <code>scrbook</code>	1
Fix bug with <code>\setline</code> at the beginning of a <code>\pstart</code>	1
Fix spacing bug with <code>\Xhooknote</code> and <code>\hooknoteX</code> when using them to insert text and not to execute code.	1

New tools to number stanzas	1
v2.1.1.	
General: Fix bug with <code>\ledpbsetting{before}</code>	1
v2.1.2.	
General: Fix 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.	200
General: <code>\Xhangindent</code> and <code>\hangindentX</code> work now with all the paragraphs in the	
note.	1
<code>\Xnoindent</code> and <code>\noindentX</code> work now again (broken in 2.0.0).	1
Change some internal code in order to provide compatibility with \TeX release of	
october 2015	1
Fix bug which inserted double space before paragraphed familiar notes.	1
Fix bug with <code>\edindex</code> when using not-Latin characters without UTF-8 engines	1
v2.2.0.	
General: Fix bug with combination of <code>\onehalfspacing</code> and two columns and three	
columns notes typeset.	1
Fix bug with some setting command and optimization option.	1
Fix spurious space with paragraphed critical notes when using $\text{Lua}\TeX$	1
Increase line list version number to ensure compatibility with new options of	
<code>reledpar</code> package.	1
New setting tools for endnotes: <code>\Xendnumberonlyfirstinline</code> ,	
<code>\Xendnumberonlyfirstintwolines</code> , <code>\Xendsymmlinenumber</code> , <code>\Xendbeforenumber</code> ,	
<code>\Xendafternumber</code> , <code>\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 TeX format 2015/10/01.	1
v2.2.2.	
General: Fix bug in <code>\sethangingsymbol</code>	1
Fix bug with old version of <code>etex</code>	1
v2.3.0.	
General: Disable empty lines as paragraph in stanza.	1
Fix compatibility of paragraphed footnotes with <code>bidir</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.	149
General: <code>\Xbhooknote</code> and <code>\bhooknoteX</code> work with notes in columns.	1
Fix bug of <code>\parindentX</code> and <code>\Xparindent</code> with two columns and three columns	
notes.	1
Fix bug with <code>\sameword</code> in right side.	1
Fix spurious space in two columns and three columns notes.	1
Fix spurious space when using optional argument of <code>\stanza</code> (introduced in v2.3.0).	1
New hooks: <code>\Xlinerrangeseparator</code> and <code>\Xendlinerrangeseparator</code>	33

Option linerangesep for critical footnotes and endnotes.	33
v2.4.1.	
General: Fix bug with \appref and \apprefwithpage (introduced in v2.4.0).	1
Fix bug with tabular environments when using babel or polyglossia languages that override \LaTeX \roman command, like Greek language.	1
Fix bug with tabular environments when using babel or polyglossia languages that override \LaTeX \roman command, like Greek.	1
v2.5.0.	
\Serefwithpage: Debug \setapprefprefixsingle	247
\edlabel: Fix bug when calling \edlabel in a footnotes of the rightside	238
\l@d@section: \endnotes take six arguments.	205
\printlines: \printlines takes an eighth argument: the line flag	178
\xlineref: \xlineref does not include anymore the side flag. Use \xflagref to get it. Not that \edlineref still contains the flag.	243
General: \apprefwithpage and \appref print double quotation mark when the label was not defined.	1
\apprefwithpage and \appref work with right side crossref.	1
\apprefwithpage works also when noend option is enabled.	1
\appref and \apprefwithpage can take linerangesep optional argument.	1
\edlabel works now in \Xfootnote.	1
\lemma can be used even when the nocritical is enabled.	1
Compatibility with new hook and tools of reledpar 2.6.0.	1
Fix spurious vertical space in astanza environment (reledpar)	1
Log now states ‘There were undefined references’ when using wrong references in \edlineref or edpageref.	1
New hooks to customize page and line number appearance in endnotes.	1
New hooks: \Xhookgroup and \hookgroupX.	1
New tools to easily make cross-reference to a passage defined by a start and an end line	50
v2.6.0.	
General: Adds compatibility with innnote and notenumber options of indextools package.	1
Fix bug with footnote counter in ledgroup (added in v2.5.0).	1
Fix bug, introduced in v2.5.0, with footnote numbering in parallel typesetting when using perpage package.	1
v2.7.0.	
\@k: \rigidbalance is split in \Xrigidbalance and \rigidbalanceX.	164
\l@d@section: \endnotes take seven arguments.	205
General: Add dash as default page range separator for \SEonlypage	1
Debug \SErefonlypage when referring to only one page.	1
Delete parenthesis after \SErefonlypage.	1
Fix (again) bugs with footnote numbering in parallel typesetting while using ledgroup environments (bug added in v2.5.0).	1
Fix bug with \Serefwithpage.	1
Fix bugs in compatibility with innote and notenumber options of indextools package, when indexing outside of a ledgroup.	1
New commands to make glossaries connected to page and linenumber with the glossaries package	1
New hooks: \Xhsize and \hsizeX	43
New hooks: \Xlemmafont and \Xendlemmafont	38

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>	43
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.	205
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 bug concerning indent in a paragraph immediately following a sectioning command (bug NOT fixed on <code>reledpar</code>)	1
Fix bug with <code>\AtEveryPstart</code> added in version 2.0.0.	1
Fix 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 bug when using <code>\linenum{page}</code> and <code>\pausenumbering... \resumenumbering</code>	1
Fix 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 bug when notes start with “plus” or “minus”.	1
v2.9.2.	
General: Fix 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 bug with line number position and reset added by v. 2.9.0	1
v2.10.0.	
<code>\print@lemma</code> : Code refactoring between <code>\parafootfmt</code> , <code>\twocolfootfmt</code> , <code>\threecolfootfmt</code> and <code>\normalfootfmt</code>	171

General: Add \AtEveryStanza and \AtEveryStopStanza.	1
Fix bug in \ledlsnotefontsetup and \ledrsnotefontsetup which could not handle \color command properly.	1
More specific error messages.	1
New hooks: \Xwrapcontent, \Xendwrapcontent and \wrapcontentX.	39
New hooks: \Xwraplemma and \Xendwraplemma	39
v2.10.1.	
General: Add ‘nopenalties’ option.	1
Fix 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	139
General: Add new tools to produce an apparatus of manuscripts	1
Fix bug in \Xparafootsep in parallel typesetting	1
Make \parafootsepX work	1
Prevent \Xtxtbeforenotes hook from causing notes to go beyond the bottom margin	1