

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	10
1.1 Aim of the package	10
1.2 History	11
1.2.1 edmac	11
1.2.2 ledmac	13
1.2.3 eledmac	13

*This file (**reledmac.dtx**) has version number v2.8.2, last revised 2016/02/27.

[†]maieul at maieul dot net

1.2.4 <code>reledmac</code>	13
1.3 List of works edited with (r)(e)ledmac	13
2 How the package works	13
3 Options	14
3.1 Specific features	14
3.2 Optimizing package performance	14
4 Text lines and paragraphs numbering	15
4.1 Text lines numbering	15
4.2 Paragraphs	16
4.2.1 Basics	16
4.2.2 Automatically producing <code>\pstart ... \pend</code>	16
4.2.3 Content before specific <code>\pstart</code> and after specific <code>\pend</code>	17
4.2.4 Content before every <code>\pstart</code> and after every <code>\pend</code>	17
4.2.5 Numbering paragraphs (<code>\pstart</code>)	17
4.2.6 Languages written in Right to Left	18
4.2.7 Memory limits	18
4.3 Lineation commands	18
4.3.1 Disabling lineation	18
4.3.2 Setting lineation start and step	19
4.3.3 Setting lineation reset	19
4.3.4 Setting line number margin	19
4.3.5 Other settings	20
4.4 Changing the line numbers	20
4.4.1 Sublineation	20
4.4.2 Locking lineation	20
4.4.3 Setting and changing line number	20
4.4.4 Line number style	21
4.4.5 Skipping and hiding number	21
4.4.6 Execute code at each line	21
5 Apparatus commands	22
5.1 Terminology	22
5.2 Critical notes	22
5.2.1 The lemma	22
5.2.2 Footnotes	23
5.2.3 Endnotes	23
5.2.4 Paragraph in critical apparatus	24
5.2.5 Change lemma and line number	24
5.2.6 Changing the names of commands for critical apparatus	25
5.3 Disambiguation of identical words in the apparatus	26
5.3.1 Basic use	26
5.3.2 Notes about input encoding with UTF-8 processor	26
5.3.3 Use with <code>\lemma</code> command	27

5.3.4 Customizing	28
5.4 Familiar notes	28
5.4.1 Basic use	28
5.4.2 Customizing mark	28
5.4.3 Separator for multiple footnotes	29
5.5 Changing series	29
5.5.1 Create a new series	29
5.5.2 Delete series	29
5.5.3 Series order	29
5.6 Position of critical and familiar footnotes	29
6 Critical apparatus appearance	30
6.1 Notes arrangement in a series	30
6.2 Control line number printing	31
6.2.1 Print line number only at first time	31
6.2.2 Arbitrary text before line number	31
6.2.3 Separator for line range	31
6.2.4 Abbreviate line range	31
6.2.5 Disable line number	32
6.2.6 Printing pstart number	32
6.2.7 Printing stanza number	33
6.2.8 Separator between line and subline numbers	33
6.2.9 Space around number	33
6.2.10 Space around line symbol	33
6.2.11 Space in place of number	34
6.2.12 Boxing line number and line symbol	34
6.3 For endnotes	35
6.4 Arbitrary code around line number	35
6.5 Separator between the lemma and the note	35
6.5.1 For footnotes	35
6.5.2 For endnotes	36
6.6 Font style	36
6.6.1 For line number	36
6.6.2 For the lemma	36
6.6.3 For all notes	37
6.7 Indent of notes content	37
6.8 Arbitrary code at the beginning of notes	37
6.9 Options for footnotes in columns	37
6.9.1 Alignment	37
6.9.2 Size of the columns	38
6.10 Options for paragraphed footnotes	38
6.10.1 Mark separation of notes	38
6.10.2 Ragged text	38
6.11 Options for block of notes	39
6.11.1 Text before notes	39
6.11.2 Code before notes	39

6.11.3 Spacing	39
6.11.4 Rule	39
6.11.5 Maximum height	39
6.11.6 Width	40
6.12 Footnotes and the <code>reledpar</code> columns	40
6.13 Endnotes in one paragraph	40
7 Fonts	41
8 Verse	41
8.1 Basic	41
8.2 Define stanza indents	41
8.3 Repeating stanza indents	42
8.4 Manual stanza indent	43
8.5 Stanza breaking	43
8.6 Hanging symbol	43
8.7 Long verse and page break	44
8.8 Content before/after verses	44
8.9 Numbering stanza	44
8.10 Various tools	44
8.11 Notes on empty lines	45
9 Grouping	45
10 Cross referencing	45
10.1 Basic use	45
10.2 Cross-referencing to a critical note	46
10.3 Cross-referencing which return a number in any case	46
10.3.1 Cross-referencing in order to define line number of a critical note	47
10.4 Not automatic cross-referencing	47
10.5 Normal \LaTeX cross-referencing	47
10.6 References to start and end lines	47
10.6.1 Reference to main text lines	47
10.6.2 References to lines that are commented on in the apparatus	48
10.6.3 Settings	48
10.7 Compatibility with <code>xr</code> package	50
11 Side notes	50
11.1 Basics	50
11.2 Setting	50
11.2.1 Width	50
11.2.2 Vertical position	50
11.2.3 Distance to the main text	51
11.2.4 Separator between notes	51

12 Indexing	51
12.1 Basics	51
12.2 Referring to critical notes	51
12.3 Separator between page and line numbers	52
12.4 Using xindy	52
12.5 Advanced setting	53
13 Glossary	53
13.1 Preamble setting	53
13.2 Commands	53
14 Tabular material	54
15 Sectioning commands	57
15.1 Sectioning commands without line numbers or critical notes	57
15.2 Sectioning commands with line numbering and critical notes	57
15.3 Optimization	58
16 Quotation environments	58
17 Page breaks	58
17.1 Control page breaking	58
17.2 Prevent page break in a long verses	59
18 Miscellaneous	59
18.1 Known and suspected limitations	60
18.1.1 floatrow package compatibility	60
18.1.2 ‘No room for a new’	60
18.1.3 Marginal notes	60
18.1.4 Paragraph shape	60
18.1.5 Paragraphed footnotes	61
18.1.6 Use with other packages	61
18.1.7 Parallel typesetting	62
I Implementation overview	63
II Preliminaries	63
II.1 Links with original edmac	63
II.2 Package declaration	63
II.3 Package options	64
II.4 Loading packages	65
II.5 Compatibility with Lua \TeX	66
II.6 Boolean flags	66
II.7 Messages	67
II.8 Gobbling	72
II.9 Miscellaneous commands	72
II.10 Prepare reledpar	73

II.11 Booleans provided by other optional packages which are required in any case	74
III Sectioning commands	74
IV List macros	78
V Line counting	79
V.1 Choosing the system of lineation	79
V.2 Line number margin	81
V.3 Line number initialization and increment	82
V.4 Line number locking	83
V.5 Line number style	84
V.6 Line number printing	84
V.7 Line number counters and lists	85
V.8 Line number locking counter	86
V.9 Line number associated to lemma	87
V.10 Reading the line-list file	90
V.11 Commands within the line-list file	92
V.12 Writing to the line-list file	103
VI Marking text for notes	108
VI.1 <code>\edtext</code> itself	109
VI.2 Substitute lemma	116
VI.3 Substitute line numbers	116
VI.4 Lemma disambiguation	117
VII Paragraph decomposition and reassembly	123
VII.1 Boxes, counters, <code>\pstart</code> and <code>\pend</code>	123
VII.2 Processing one line	128
VII.2.1 General process	128
VII.2.2 Process for “normal” line	129
VII.2.3 Process for line containing <code>\eledsection</code> command	130
VII.2.4 Hooks	131
VII.2.5 Sidenotes and marginal line number initialization	131
VIII Line and page number computation	132
IX Line number printing	135
X Pstart number printing in side	139
XI Restoring footnotes and penalties	140
XI.1 Add insertions to the vertical list	140
XI.2 Penalties	141
XI.3 Printing leftover notes	142

XII Critical footnotes	143
XII.1 Fonts	143
XII.2 Individual note options	144
XII.3 Notes language	144
XII.4 General survey of the way we manage notes	145
XII.5 General setup	146
XII.6 Footnotes arrangement	146
XII.6.1 User level macro	146
XII.6.2 Normal footnote	147
XII.6.3 Paragraphed footnotes	151
XII.6.4 Columnar footnotes	158
XII.7 Critical notes presentation	164
XII.7.1 Font tools	164
XII.7.2 Pstart number in footnote	165
XII.7.3 Line number printing	166
XIII Familiar footnotes	174
XIII.1 Adjacent footnotes	174
XIII.2 Regular footnotes for numbered texts	176
XIII.3 Footnote formats	178
XIII.4 Footnote arrangement	178
XIII.4.1 User level macro	178
XIII.4.2 Normal footnotes	178
XIII.4.3 Two columns footnotes	184
XIII.4.4 Three columns footnotes	186
XIII.4.5 Paragraphed footnotes	188
XIII.5 Wrapping footnote marks in hyperlink	192
XIV Code common to both critical and familiar footnote in normal arrangement	193
XV Footnotes' width for two columns	194
XVI Footnotes' order	195
XVII Footnotes' rule	195
XVIII Specific skip for first series of footnotes	196
XVIII.0.1 Overview	196
XVIII.0.2 User level command	197
XVIII.0.3 Internal commands	197
XIX Endnotes	198

XX Generate series of notes	206
XX.1 Test if series is still existing	207
XX.2 Init specific to <code>reledpar</code>	207
XX.3 For critical footnotes	207
XX.3.1 Options	207
XX.3.2 Create inserts, needed to add notes in foot	208
XX.3.3 Create commands for critical apparatus, <code>\Afootnote</code> , <code>\Bfootnote</code> etc.	209
XX.3.4 Set standard display	211
XX.4 For familiar footnotes	211
XX.4.1 Options	211
XX.4.2 Create tools for familiar footnotes (<code>\footnoteX</code>)	212
XX.5 The endnotes	213
XX.5.1 The auxiliary file	213
XX.5.2 The main macro	214
XX.5.3 The options	215
XX.6 Init standards series (A,B,C,D,E)	216
XXI Setting series display	216
XXI.1 Change series order	216
XXI.2 Test series order	217
XXI.2.1 Get the first series	217
XXI.3 Series setting	217
XXI.3.1 General way of working	217
XXI.3.2 Tools to set options	217
XXI.3.3 Tools to generate options commands	219
XXI.3.4 Options for critical notes	220
XXI.3.5 Options for familiar notes	222
XXI.3.6 Options for endnotes	223
XXI.4 Hooks for a particular footnote	224
XXI.5 Alias	225
XXII Output routine	225
XXII.0.1 Page number management	225
XXII.0.2 Extra footnotes output	225
XXII.0.3 Standard output's commands patching	228
XXIII Cross referencing	230
XXIII.1 Compatibility with <code>xref</code>	243
XXIV Side notes	244
XXV Minipages and such	250

XXVI Indexing	255
XXVI.1 Looking on package order	255
XXVI.2 Auxiliary macros for <code>\edindex</code>	255
XXVI.3 Code specific to <code>\edindexin</code> critical footnotes	256
XXVI.4 Analysis of command in indexed text	257
XXVI.5 Code for the formatted index	258
XXVI.6 Main code	258
XXVI.7 Hyperlink	259
XXVI.8 ‘innote’ and ‘notenumber’ option of <code>indextols</code> package	262
XXVII Glossaries	263
XXVIII Verse	265
XXVIII.1 Hanging symbol management	265
XXVIII.2 Using <code>&</code> character	265
XXVIII.3 Code category setting	266
XXVIII.4 Stanza count and indent	266
XXVIII.5 Numbering stanza	267
XXVIII.6 Stanza number in note	268
XXVIII.7 Main work	269
XXVIII.8 Restore catcode and penalties	271
XXIX Arrays and tables	271
XXIX.1 Preamble: macro as environment	271
XXIX.2 Tabular environments	274
XXIX.2.1 Disabling and restoring commands	275
XXIX.2.2 Counters, boxes and lengths	278
XXIX.2.3 Tabular typesetting	282
XXIX.2.4 Environments	293
XXX Quotation’s commands	293
XXXI Section’s title commands	294
XXXI.1 Commands to disable some feature	294
XXXI.2 General overview	295
XXXI.3 <code>\beforeeledchapter</code> command	295
XXXI.4 Auxiliary commands	296
XXXI.5 Patching standard commands	297
XXXI.6 Main code of <code>\eledxxx</code> commands	301
XXXI.7 Macros written in the auxiliary file	304
XXXII Page breaking or no page breaking depending of specific lines	306
XXXIII Long verse: prevents being separated by a page break	308
XXXIV Tools for hyperref package	308

XXXV Compatibility with eledmac	309
Appendix A Things to do when changing versions	312
Appendix A.1 Migrating from edmac to ledmac	312
Appendix A.2 Migration from ledmac to eledmac	313
Appendix A.3 Migration to eledmac 1.5.1	314
Appendix A.4 Migration to eledmac 1.12.0	314
Appendix A.5 Migration to eledmac 1.7.1	315
Appendix A.6 Migration to eledmac 1.21.0	315
Appendix A.6.1 \Xledsetnormalparstuffand\ledsetnormalparstuffX	315
Appendix A.6.2 Endnotes	315
Appendix A.7 Migration to eledmac 1.22.0	315
Appendix A.8 Migration to eledmac 1.23.0	315
Appendix A.9 Migration from eledmac to reledmac	316
Appendix A.9.1 Risk of ‘no room for a new’	316
Appendix A.9.2 Multiple indices with memoir	316
Appendix A.9.3 Deprecated commands and options	316
Appendix A.9.4 \renewcommandreplaced by command	317
Appendix A.9.5 Commands the names of which have been changed . . .	317
Appendix A.9.6 Endnotes	319
Appendix A.9.7 Z Series	319
Appendix A.9.8 Internal commands	319
Appendix A.10 Migration to reledmac 2.1.0	319
Appendix A.11 Migration to reledmac 2.1.3	319
Appendix A.12 Migration to reledmac 2.3.0	319
Appendix A.13 Migration to reledmac 2.4.0	320
Appendix A.14 Migration to reledmac 2.5.0	320
Appendix A.15 Migration to reledmac 2.7.0	320
Appendix A.16 Migration to reledmac 2.7.2	320
Appendix A.17 Migration to reledmac 2.8.0	320
References	321
Index	321
Change History	365

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

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

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

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

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

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

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

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

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

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

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

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

¹²Being prepared at Poona and Lausanne Universities.

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 6 p. 30). 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. 313.

1.2.4 reledmac

`eledmac` has facilitated the creation of customized critical editions. However, the changes made to allow such customization were made in a non-systematic way. Many deprecated commands were kept and many technical ‘debts’ were accumulated, hindering the future evolution of the package.

For these reasons, Maïeul Rouquette decided on a spring cleaning of the code. As some commands name were changed, the resulting compatibility was broken (a little).

A new name was selected: `reledmac` (extended renewed `eledmac`). To migrate from `eledmac` to `reledmac`, please read Appendix A.9 p. 316.

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 L^AT_EX itself. Although your textual apparatus and line numbers will be printed on the first run, it takes two more passes through L^AT_EX 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.

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.

3 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.

3.1 Specific features

draft underlines lemmas in the main text.

eledmac-compatible help to migrate from `eledmac` to `reledmac` (see Appendix A.9.5 p. 317).

nopbinverse prevents page breaks inside verses.

noquotation by default, the quotation environment is redefined inside numbered text. You can disable this redefinition with `noquotation` (see 16 p. 58).

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

xindy and **xindy+hyperref** are for selecting `xindy` as the index processor (12.4 p. 52).

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.

3.2 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.

noeledsec disables tools for `\eledsection` and related commands (15.2 p. 57).

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 9 p. 45). 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.

4 Text lines and paragraphs numbering

4.1 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).

4.2 Paragraphs

4.2.1 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
```

4.2.2 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

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

Both `\pstart` and `\pend` can take a optional argument in brackets. Its content will be printed before the beginning of `\pstart` / after the end of `\pend` instead of the argument of `\AtEveryPstart` / `\AtEveryPend`. If you need to start a `\pstart` with brackets, or to add brackets after a `\pend`, just add a `\relax` between `\pstart ... \pend` and the brackets.

This feature is also useful when typesetting verses (see 8 p. 41) or `reledpar` (see 18.1.7 p. 62).

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

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

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

4.2.5 Numbering paragraphs (`\pstart`)

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

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

4.2.6 Languages written in Right to Left

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

4.2.7 Memory limits

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

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

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

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

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

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

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

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

4.3 Lineation commands

4.3.1 Disabling lineation

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

4.3.2 Setting lineation start and step

`\firstlinenum`
`\linenumincrement`

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

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

`\firstsublinenum`
`\sublinenumincrement`
`\linenumberlist`

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

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

```
\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.

4.3.3 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.

4.3.4 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).

4.3.5 Other settings

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

4.4 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.

4.4.1 Sublineation

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

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

You can change the separator between line number and subline number or using `\Xsublinesep` without any option argument (6.2.8 p. 33 or using `\Xsublinesepside`. But in the second case, it will change the separator only for line number in side, not for the footnotes.

4.4.2 Locking lineation

`\startlock` The `\startlock` command, used in running text, locks the line number at its current
`\endlock` value, until you insert `\endlock`. It can tell for itself whether you are in a patch of line or sub-line numbering. One use for line-number locking is in printing poetry: there the line numbers should be those of verse lines rather than of printed lines, even when a verse line requires several printed lines. But in this case you may use the `\stanza` mechanism, see 8 p. 41.

`\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}`.

4.4.3 Setting and changing line number

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

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

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

4.4.4 Line number style

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

Alph Uppercase letters (A ... Z).

alph Lowercase letters (a ... z).

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

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

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

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

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

4.4.5 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.

4.4.6 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.

5 Apparatus commands

5.1 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.

5.2 Critical notes

5.2.1 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		
July 16, 1954.}		1 Smith] Jones C, D.
}		
		1-2 I saw my friend Smith on Tuesday.] The
		date was July 16, 1954.

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

5.2.2 Footnotes

The second argument of the `\edtext` macro, $\langle commands \rangle$, 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
`\Bfootnote` like `\Afootnote{\langle text \rangle}`. When all of the six are used, the A notes appear in a layer
`\Cfootnote` just below the main text, followed by the rest in turn, down to the E notes at the bottom.
`\Dfootnote` These are the main macros that you will use to construct the critical apparatus of your
`\Efootnote` text.

If you need more series of critical notes, please look at 5.5.1 p. 29.

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. 6.2.4 p. 31).
- `nonum` disables line numbering for this note. A horizontal blank space is added instead. You can use `\Xinplaceoflemmaseparator` to set it (6.5.1 p. 35).
- `nosep` to disable the lemma separator for this note.
- `linangesep= $\langle c \rangle$` to change to $\langle c \rangle$ the separator between start line and end line for this particular note.

Example: `\Afootnote[nonum]{\langle text \rangle}`.

5.2.3 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
`\Cendnote` `reledmac`.

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

`\Aendnote[\langle option \rangle]{\langle text \rangle}`.

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

- `fulllines` to disable `\Xendtwolines` and `\Xendmorethantwolines` features for this particular note (cf. 6.2.4 p. 31).
- `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 (6.5.2 p. 36).

- `linrangesep=<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 (6.5.2 p. 36).

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

5.2.4 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!

5.2.5 Change lemma and line number

`\lemma`

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


```

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

```

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

1 Smith] Jones C, D.

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

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

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

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

This command does not change the marginal line numbers in any way; it just changes the numbers passed to the notes. Its use comes in situations that `\edtext` has trouble dealing with for whatever reason. If you need notes for overlapping passages that aren't nested, for instance, you can use `\lemma` and `\linenum` to generate such notes despite the limitations of `\edtext`. If the `⟨lemma⟩` argument to `\edtext` is extremely long, you may run out of memory; here again you can specify a note with an abbreviated lemma using `\lemma` and `\linenum`. The numbers used in `\linenum` need not be entered manually; you can use the 'x-' symbolic cross-referencing commands below (10 p. 45) 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.

5.2.6 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

```
\newcommand{\variant}[2][1,usedefault]{\Afootnote[#1]{#2}}
```

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`.

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

5.3 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.

5.3.1 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.

5.3.2 Notes about input encoding with UTF-8 processor

If you use UTF-8 processor, like `XYTeX` 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 `XYTeX`, 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.

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

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

5.3.3 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`:

- 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`.

1 _{inlemma}	2	3 ₂	1...3	4	5 ₁	1...5
----------------------	---	----------------	-------	---	----------------	-------

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

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

The `\sameword` number 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.

5.3.4 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}%
}
```

5.4 Familiar notes

5.4.1 Basic use

<code>\footnoteA</code> <code>\footnoteB</code> <code>\footnoteC</code> <code>\footnoteD</code> <code>\footnoteE</code>	<p>As well as the standard L^AT_EX footnotes generated via <code>\footnote</code>, the package also provides five series of additional footnotes called <code>\footnoteA</code> through <code>\footnoteE</code>. These have the familiar marker in the text, and the marked text at the foot of the page can be formatted using any of the styles described for the critical footnotes. Note that the ‘regular’ footnotes have the series letter at the end of the macro name whereas the critical footnotes have the series letter at the start of the name.</p>
---	---

5.4.2 Customizing mark

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

5.4.3 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.

5.5 Changing series

5.5.1 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}`.

5.5.2 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}
```

5.5.3 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.

5.6 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}
```

6 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 \LaTeX 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.

6.1 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 18.1.5 p. 61.
- `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.

6.2 Control line number printing

6.2.1 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`

`\Xendnumberonlyfirstintwolines`

`\Xendsymlinenum`

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

6.2.2 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.

6.2.3 Separator for line range

`\Xlinerangeseparator`

`\Xendlinerangeseparator`

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.

6.2.4 Abbreviate line range

`\Xtwolines`

`\Xmorethantwolines`

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

line numbers. Instead of this, it is also possible to print an abbreviation which stands for “line 1 and subsequent line(s)”.

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

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

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

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

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

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

`\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. 5.2.2 p. 23.

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

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

6.2.5 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.

6.2.6 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.

6.2.7 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.

6.2.8 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{.}`.

6.2.9 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.

6.2.10 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.

6.2.11 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.

6.2.12 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>	

6.3 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`.

6.4 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 `\Xendbeforeasymlinenumber` 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 `\Xendafterasymlinenumber` space.

6.5 Separator between the lemma and the note

6.5.1 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.

6.5.2 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.

6.6 Font style

6.6.1 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`.

6.6.2 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 T_EX font command to the lemma. For example, to have boldface lemma:

`\Xendlemmafont`

`\Xlemmafont{\bfseries}`

`\Xendlemmafont⟨arg⟩⟨cmd⟩` is the same for endnotes.

6.6.3 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> .

6.7 Indent of notes content

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

6.8 Arbitrary code at the beginning of notes

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

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

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

6.9 Options for footnotes in columns

6.9.1 Alignment

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

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

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

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

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

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

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

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

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

6.9.2 Size of the columns

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

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

6.10 Options for paragraphed footnotes

6.10.1 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.

6.10.2 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.

6.11 Options for block of notes

6.11.1 Text before notes

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

6.11.2 Code before notes

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

6.11.3 Spacing

`\Xbeforenotes` You can change the vertical space before the rule of the critical notes with `\Xbeforenotes[$\langle s \rangle$]{ $\langle l \rangle$ }`. 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[$\langle s \rangle$]{ $\langle l \rangle$ }`. 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{ $\langle l \rangle$ }`. 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{ $\langle l \rangle$ }`. You can disable this feature by setting the length to 0pt.

6.11.4 Rule

`\Xafterrule` You can change the vertical space printed after the rule of the critical notes with `\Xafterrule[$\langle s \rangle$]{ $\langle l \rangle$ }`. 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[$\langle s \rangle$]{ $\langle l \rangle$ }`. 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.

6.11.5 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[$\langle s \rangle$]{ $\langle l \rangle$ }`. 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.

6.11.6 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 with of the block of notes. If you want to change the width of each column when typesetting notes in columns, use `\Xhsizetwocol`, `\Xhsizethreecol`, `\hsizetwocolX`, `\hsizethreecolX`, see 6.9.2 p. 38.

6.12 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.

6.13 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}`.

7 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.

8 Verse

8.1 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.

8.2 Define stanza indents

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

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

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

To specify these multiples one invokes, for example
`\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. 8.6 p. 43.

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.

8.3 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. 314.

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.

8.4 Manual stanza indent

`\stanzaindent` You can set the indent of some specific verse by calling `\stanzaindent{⟨value⟩}` at the beginning of the verse, before any other character. In this case, the indent defined by `\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`.

8.5 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.

8.6 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}
```

8.7 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 17.2 p. 59 for further details.

8.8 Content before/after verses

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

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

8.9 Numbering stanza

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

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

You can use this feature in combination with `\Xstanza` (6.2.7 p. 33).

`thestanza` . You can redefine `\thestanza` to change the aspect of stanza number. Default value is:

```
\renewcommand{\thestanza}{%
\textbf{\arabic{stanza}}%
}
```

You can change the value of the `stanza` counter with the usual commands of \LaTeX .

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

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

8.10 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[$\langle len \rangle$]{ $\langle text \rangle$ }` at the start of a line in a stanza (or elsewhere) will typeset $\langle text \rangle$ at a distance $\langle len \rangle$ before the line. The default $\langle len \rangle$ is `\stanzaindentbase`.

8.11 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 (8.8 p. 44).

9 Grouping

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

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

You can also put familiar footnotes (see section 5.4) 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}`

10 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.

10.1 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{<lab>}`, or `\edlineref{<lab>}` will produce, respectively, the page, line, sub-line and pstart on which the `\edlabel{<lab>}` command occurred.
`\edlineref`
`\sublineref`
`\pstartref`

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

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

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

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

10.2 Cross-referencing to a critical note

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

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

If you want to refer to starting and ending lines, you should use `\appref` and related tools (10.6.2 p. 48).

10.3 Cross-referencing which return a number in any case

`\xpageref` Where #1 stands for the reference.
`\xlineref`
`\xsublineref`
`\xpstartref`

However, there are situations in which you will want `reledmac` to return a number without displaying any warning messages about undefined labels or the like: if you want to use the reference in a context where \TeX is looking for a number, such a warning will lead to a complaint that the number is missing. This is the case for references used within the argument to `\linenum`, for example (see 5.2.5 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.

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

- When `hyperref` is loaded, the `hyperref` link will not be added. (Indeed, it is not a limitation, but a feature.)
- With `reledpar`, the `\xlineref` does not insert the right side flag, in order to obtain a line number. Use `\xflagref` to obtain the side flag, depending of your flag.

10.3.1 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., 5.2.5 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.

10.4 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.

10.5 Normal \TeX cross-referencing

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

10.6 References to start and end lines

10.6.1 Reference to main text lines

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

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

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

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

`\SEref` The main utility is to use them with three other commands. `\SEref{<label>}` will

make a cross-reference printed as a reference in critical footnotes.

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

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

10.6.2 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` (5.2.5 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

`\apprefwithpage` the second will print the lines as they are printed in endnotes.

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

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

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

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

Also note that `\setSerefonlypageprefixsingle` is only a shortcut for `\Xendbeforepagenumber` (see 10.6.3 p. 49). 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 commandes 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 (5.2.2 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 (5.2.3 p. 23).

10.7 Compatibility with xr package

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

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

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

11 Side notes

11.1 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.

11.2 Setting

11.2.1 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`.

11.2.2 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).

11.2.3 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.

11.2.4 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>}`.

12 Indexing

12.1 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 & linenumber should be added to the raw index file.

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

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`.

12.2 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}}
```

12.3 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.

12.4 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

12.5 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.

13 Glossary

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

13.1 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 will have consequences on your use of `\edindex`, and you should set your `.ist` file (?? p. ??).

13.2 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`.

14 Tabular material

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

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

```

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

```

1	2	3
a	bb	ccc
AAA	BB	C

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

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

For example:

```

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

1	2	3	4
a	\sqcup		d
A	B	C	D

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

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

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

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

For example:

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

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

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

`\edvertdots`

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



```

m & n & o & p & \\\
k & & L & cvb & \edvertline{4pc}
\end{edarrayr}

```

$$\begin{array}{cccc}
 a & b & C & d \\
 v & w & x & y \\
 m & n & o & p \\
 k & & L & cvb
 \end{array}
 \left|
 \right.$$

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

15 Sectioning commands

15.1 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` (4.2.3 p. 17):

```

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

```

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

15.2 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.

15.3 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.

16 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.

17 Page breaks

17.1 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.

17.2 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.

18 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
```

18.1 Known and suspected limitations

18.1.1 floatrow package compatibility

The floatrow package must be loaded before the reledmac.

18.1.2 ‘No room for a new’

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

The first thing in order to prevent such problem is to use the options to optimize reledmac. For example, if you need only two series of notes, use `series={A,B}` option. Read 15.3 p. 58 in order to know which are there options.

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

‘**no room for a new count**’ is often caused by a conjunction with biblatex. Load reledmac (and reledpar) *before* biblatex.

‘**no room for a new write**’ can be caused by with 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.

18.1.3 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.

18.1.4 Paragraph shape

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

`\ballast` \LaTeX is a three-pass system, but even after a document has been processed three times, there are some tricky situations in which the page breaks decided by \TeX never settle down. At each successive run, reledmac may oscillate between two different sets of page decisions. To stop this happening, should it arise, Wayne Sullivan suggested the inclusion of the quantity `\ballast`. The amount of `\ballast` will be subtracted from the penalties which apply to the page breaks calculated on the *previous* run through \TeX ,

thus reinforcing these breaks. So if you find your page breaks oscillating, insert `\setcounter{ballast}{100}` or some such figure, and with any luck the page breaks will settle down. Luckily, this problem does not crop up at all often.

18.1.5 Paragraphed footnotes

The restriction on explicit line-breaking in paragraphed footnotes, mentioned in a footnote 6.1 p. 30, and described in more detail on XII.6.3 p. 154, 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.

`\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` (6.11.5 p. 39), 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.

18.1.6 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
```

²²Reported by Dirk-Jan Dekker in the CTT thread ‘Incompatibility of “color” package’ on 2003/08/28.

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

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

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

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

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

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

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

18.1.7 Parallel typesetting

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

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

I Implementation overview

We present the `reledmac` code in roughly the order in which it is used during a run of \TeX . The order is *exactly* that in which it is read when you load The `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}[2016/02/27 v2.8.2 typeset critical edition]%
4 %

```

II.3 Package options

<pre> \ifledfinal \ifnocritical@ \if@noeled@sec \ifnoend@ \ifnofamiliar@ \ifnoledgroup@ \ifparapparatus@ \ifnoquotation@ \iflednopbinverse \ifparledgroup \ifwidthliketwocolumns \ifxindy@ \ifxindyhyperref@ \ifeledmaccompat@ </pre>	<p>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.</p> <pre> 5 \RequirePackage{xkeyval} 6 % </pre> <p>The parledgroup option is for reledpar. However, it has consequence on reledmac internal command. So we need to define the boolean now.</p> <pre> 7 \newif\ifparledgroup 8 % </pre> <p>And now, the options of reledmac.</p> <pre> 9 \DeclareOptionX{series}[A,B,C,D,E]{\xdef\default@series{#1}} 10 \ExecuteOptionsX{series}% 11 12 \newif\if@noeled@sec% 13 \DeclareOptionX{noeledsec}{\@noeled@sectrue} 14 15 \newif\ifnocritical@% 16 \DeclareOptionX{nocritical}{\nocritical@true}% 17 18 \newif\ifnofamiliar@% 19 \DeclareOptionX{nofamiliar}{\nofamiliar@true}% 20 21 \newif\ifnoledgroup@% 22 \DeclareOptionX{noledgroup}{\noledgroup@true}% 23 24 \newif\ifnoend@% 25 \DeclareOptionX{noend}{% 26 \let\l@dend@open\@gobble% 27 \let\l@dend@close\relax% 28 \global\let\l@dend@stuff=\relax% 29 \noend@true% 30 }% 31 32 \newif\ifnoquotation@ 33 \DeclareOptionX{noquotation}{\noquotation@true} 34 35 36 \newif\ifledfinal 37 \DeclareOptionX{final}{\ledfinaltrue} 38 \DeclareOptionX{draft}{\ledfinalfalse} </pre>
---	---


```

39 \ExecuteOptionsX{final}
40
41 \newif\ifparapparatus@
42 \DeclareOptionX{parapparatus}{\parapparatus@true}
43
44 \newif\iflednopbinverse
45 \DeclareOptionX{nopbinverse}{\lednopbinversetrue}
46
47 \newif\ifwidthliketwocolumns%
48 \DeclareOptionX{widthliketwocolumns}{\widthliketwocolumnstrue}%
49
50 \newif\ifxindy@
51 \DeclareOptionX{xindy}[eledmac-markup-attr.xdy]{%
52   \AtBeginDocument{\immediate\openout\eledmac@xindy@out=#1}%
53   \newwrite\eledmac@xindy@out%
54   \xindy@true%
55   \gdef\eledmacmarkuplocdepth{:depth 1}%
56   \AtEndDocument{\immediate\closeout\eledmac@xindy@out}%
57 }%
58
59 \newif\ifxindyhyperref@
60 \DeclareOptionX{xindy+hyperref}{%
61   \xindyhyperref@true%
62 }%
63
64 \newif\ifeledmaccompat@%
65 \DeclareOptionX{eledmac-compat}{%
66   \eledmaccompat@true%
67 }%
68 %

```

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.

```

69 \ProcessOptionsX*\relax
70
71 %

```

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

```

72 \RequirePackage{xargs}
73 \RequirePackage{etoolbox}

```

```

74 \ifl@t@r\fmtversion{2015/10/01}
75 {}%
76 {\RequirePackage{etex}%
77 \csname reserveinserts\endcsname{32}%
78 }
79 \RequirePackage{suffix}
80 \RequirePackage{xstring}
81 \RequirePackage{ifluatex}
82 \RequirePackage{ragged2e}
83 \RequirePackage{ifxetex}%
84 %

```

II.5 Compatibility with Lua_T_EX

Here, we enable some primitives for Lua_T_EX.

```

85 \ifx\directlua\undefined\else%
86 \directlua{tex.enableprimitives("",{"texdir","pdir","bodydir"})}
87 \fi
88 %

```

II.6 Boolean flags

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

```

89 \newif\ifl@dmemoir
90 \@ifclassloaded{memoir}{\l@dmemoirtrue}{\l@dmemoirfalse}
91
92 %

```

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

```

93 \newif\if@ledgroup%
94 %

```

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

```

95 \newif\ifl@imakeidx
96 \@ifpackageloaded{imakeidx}{\l@imakeidxtrue}{}%False is the default value
97 %

```

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

```

98 \newif\ifl@indextools%
99 \@ifpackageloaded{indextools}{%
100 \l@indextoolstrue%
101 \l@imakeidxtrue%
102 \let\imki@wrindexentry\indtl@wrindexentry%
103 }{}%
104 %

```

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.

```
105 \ifdef{\if@RTL}{-}{\newif\if@RTL}
106 %
```

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.

```
107 \newcommand{\reledmac@warning}[1]{\PackageWarning{reledmac}{#1}}
108 %
```

`\reledmac@error` Write an error message.

```
109 \newcommand{\reledmac@error}[2]{\PackageError{reledmac}{#1}{#2}}
110 %
```

```
\led@err@NumberingStarted11 \newcommand*{\led@err@NumberingStarted}{%
d@err@NumberingNotStarted12 \reledmac@error{Numbering has already been started}{\@ehc}}
NumberingShouldHaveStarted13 \newcommand*{\led@err@NumberingNotStarted}{%
114 \reledmac@error{Numbering was not started}{\@ehc}}
115 \newcommand*{\led@err@NumberingShouldHaveStarted}{%
116 \reledmac@error{Numbering should already have been started}{\@ehc}}
117 %
```

```
d@err@edtextoutsidestart18 \newcommand*{\led@err@edtextoutsidestart}{%
119 \reledmac@error{\string\edtext\space outside numbered paragraph (\pstart\
ldots\pend)}{\@ehc}}%
120 %
```

```
\led@mess@NotesChanged21 \newcommand*{\led@mess@NotesChanged}{%
122 \typeout{reledmac reminder: }%
123 \typeout{ The number of the footnotes in this section
124 has changed since the last run.}%
125 \typeout{ You will need to run LaTeX two more times
126 before the footnote placement}%
127 \typeout{ and line numbering in this section are
128 correct.}}
129 %
```

```

\led@mess@SectionContinued30 \newcommand*{\led@mess@SectionContinued}[1]{%
131 \message{Section #1 (continuing the previous section)}}
132 %

\led@err@LineationInNumbered33 \newcommand*{\led@err@LineationInNumbered}{%
134 \reledmac@error{You can't use \string\lineation\space within
135 a numbered section}{\@ehc}}
136 %

\led@warn@BadLineation37 \newcommand*{\led@warn@BadLineation}{%
\led@warn@BadLinenummargin38 \reledmac@warning{Bad \string\lineation\space argument}}
\led@warn@BadLockdisp39 \newcommand*{\led@warn@BadLinenummargin}{%
\led@warn@BadSublockdisp40 \reledmac@warning{Bad \string\linenummargin\space argument}}
141 \newcommand*{\led@warn@BadLockdisp}{%
142 \reledmac@warning{Bad \string\lockdisp\space argument}}
143 \newcommand*{\led@warn@BadSublockdisp}{%
144 \reledmac@warning{Bad \string\sublockdisp\space argument}}
145 %

\led@warn@NoLineFile46 \newcommand*{\led@warn@NoLineFile}[1]{%
147 \reledmac@warning{Can't find line-list file #1}}
148 %

\led@warn@LineFileObsolete49 \newcommand*{\led@warn@Obsolete}[1]{%
150 \reledmac@warning{Line-list file #1 was obsolete. We have not read it.
Please run LaTeX again.}}
151 %

\led@warn@BadAdvancelineSubline52 \newcommand*{\led@warn@BadAdvancelineSubline}{%
\led@warn@BadAdvancelineLine53 \reledmac@warning{\string\advanceline\space produced a sub-line
154 number less than zero.}}
155 \newcommand*{\led@warn@BadAdvancelineLine}{%
156 \reledmac@warning{\string\advanceline\space produced a line
157 number less than zero.}}
158 %

\led@warn@BadSetline59 \newcommand*{\led@warn@BadSetline}{%
\led@warn@BadSetlinenum60 \reledmac@warning{Bad \string\setline\space argument}}
161 \newcommand*{\led@warn@BadSetlinenum}{%
162 \reledmac@warning{Bad \string\setlinenum\space argument}}
163 %

```

```

\led@err@PstartNotNumbered\newcommand*{\led@err@PstartNotNumbered}{%
\led@err@PstartInPstart\reledmac@error{\string\pstart\space must be used within a
\led@err@PendNotNumberednumbered section}{\@ehc}}
\led@err@PendNoPstart\newcommand*{\led@err@PstartInPstart}{%
\reledmac@error{\string\pstart\space encountered while another
\string\pstart\space was in effect}{\@ehc}}
\led@err@AutoparNotNumbered\newcommand*{\led@err@PendNotNumbered}{%
\reledmac@error{\string\pend\space must be used within a
\rr@NumberingWithoutPstartnumbered section}{\@ehc}}
\newcommand*{\led@err@PendNoPstart}{%
\reledmac@error{\string\pend\space must follow a \string\pstart}{\@ehc}}
\newcommand*{\led@err@AutoparNotNumbered}{%
\reledmac@error{\string\autopar\space must be used within a
numbered section}{\@ehc}}
\newcommand*{\led@err@NumberingWithoutPstart}{%
\reledmac@error{\string\beginnumbering...\string\endnumbering\space
without \string\pstart}{\@ehc}}%
%

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

\led@warn@DuplicateLabel\newcommand*{\led@warn@DuplicateLabel}[1]{%
\reledmac@warning{Duplicate definition of label `#1'\@gobble}%
\led@warn@AppLabelOutEdtext\@latex@warning{\no@line{Label `#1' multiply defined}}%
\led@warn@RefUndefined}%
\newcommand*{\led@warn@AppLabelOutEdtext}[1]{%
\reledmac@warning{\string\applabel\space outside of \string\edtext\space
`#1' on page \the\pageno.}}%
\newcommand*{\led@warn@RefUndefined}[1]{%
\G@refundefinedtrue%
\reledmac@warning{Reference `#1' on page \the\pageno\space undefined.%
Using `000'.}%
\@latex@warning{Reference `#1' undefined\on@line}%
}%
\newcommand*{\led@warn@pairRefUndefined}[1]{%
\G@refundefinedtrue%
\reledmac@warning{Reference `#1:start' and/or `#1:end' on page \the\
pageno\space undefined.
Using `??'.}%
\@latex@warning{Reference `#1:start' and/or `#1:end' undefined\on@line}%
}%
%

\led@warn@NoMarginpars\newcommand*{\led@warn@NoMarginpars}{%

```

```

204 \reledmac@warning{You can't use \string\marginpar\space in numbered text
    }}
205 %

```

```

\led@warn@BadSidenotemargin06 \newcommand*{\led@warn@BadSidenotemargin}{%
207 \reledmac@warning{Bad \string\sidenotemmargin\space argument}}
208 %

```

```

\led@warn@NoIndexFile09 \newcommand*{\led@warn@NoIndexFile}[1]{%
210 \reledmac@warning{Undefined index file #1}}
211 %

```

```

\led@warn@SeriesStillExist12 \newcommand{\led@warn@SeriesStillExist}[1]{%
213 \reledmac@warning{Series #1 is still existing !}%
214 }%
215 %

```

```

\led@err@ManySidenotes16 \newcommand{\led@err@ManySidenotes}{%
\led@err@ManyLeftnotes17 \ifledRcol{%
\led@err@ManyRightnotes18 \reledmac@warning{\itemcount@\space sidenotes on line \the\line@numR\
space p. \the\page@numR}%
219 \else%
220 \reledmac@warning{\itemcount@\space sidenotes on line \the\line@num\
space p. \the\page@num}%
221 \fi%
222 }%
223 \newcommand{\led@err@ManyLeftnotes}{%
224 \ifledRcol{%
225 \reledmac@warning{\itemcount@\space leftnotes on line \the\line@numR\
space p. \the\page@numR}%
226 \else%
227 \reledmac@warning{\itemcount@\space leftnotes on line \the\line@num\
space p. \the\page@num}%
228 \fi%
229 }%
230 \newcommand{\led@err@ManyRightnotes}{%
231 \ifledRcol{%
232 \reledmac@warning{\itemcount@\space rightnotes on line \the\line@numR\
space p. \the\page@numR}%
233 \else%
234 \reledmac@warning{\itemcount@\space rightnotes on line \the\line@num\
space p. \the\page@num}%
235 \fi%
236 }%
237 %

```

```

\led@err@TooManyColumns38 \newcommand*\led@err@TooManyColumns}{%
\led@err@UnequalColumns39 \reledmac@error{Too many columns}{\@ehc}}
\led@err@LowStartColumn40 \newcommand*\led@err@UnequalColumns}{%
\led@err@HighEndColumn41 \reledmac@error{Number of columns is not equal to the number
\led@err@ReverseColumns42 in the previous row (or \protect\\ \space forgotten?)}{\@ehc}}
243 \newcommand*\led@err@LowStartColumn}{%
244 \reledmac@error{Start column is too low}{\@ehc}}
245 \newcommand*\led@err@HighEndColumn}{%
246 \reledmac@error{End column is too high}{\@ehc}}
247 \newcommand*\led@err@ReverseColumns}{%
248 \reledmac@error{Start column is greater than end column}{\@ehc}}
249 %

err@EdtextWithoutFootnote50 \newcommand*\led@err@EdtextWithoutFootnote}{%
251 \reledmac@error{edtext without Xfootnote. Check syntaxis.}{\@ehc}%
252 }%
253 %

err@FootnoteWithoutEdtext54 \newcommand*\led@err@FootnoteWithoutEdtext}{%
255 \reledmac@error{Xfootnote without edtext. Check syntax.}{\@ehc}%
256 }%
257 %

error@ImakeidxAfterEledmac58 \newcommand*\led@error@ImakeidxAfterEledmac}{%
259 \reledmac@error{Imakeidx must be loaded before reledmac.}{\@ehc}%
260 }%
261 %

or@IndextoolsAfterEledmac62 \newcommand*\led@error@IndextoolsAfterEledmac}{%
263 \reledmac@error{Indextools must be loaded before reledmac.}{\@ehc}%
264 }%
265 %

error@fail@patch@@makecol66 \newcommand*\led@error@fail@patch@@makecol}{%
267 \reledmac@error{Fail to patch \string\@makecol\space command.}{\@ehc}%
268 }%
269 %

error@fail@patch@@reinserts70 \newcommand*\led@error@fail@patch@@reinserts}{%
271 \reledmac@error{Fail to patch \string\@reinserts\space command.}{\@ehc}%
272 }%
273 %

```

```

\led@error@fail@patch@@doclearpage74 \newcommand{\led@error@fail@patch@@doclearpage}{%
275 \reledmac@error{Fail to patch \string\@doclearpage\space command.}\@ehc}
%
276 }%
277 %

```

```

\led@error@fail@patch@@iiiminipage78 \newcommand{\led@error@fail@patch@@iiiminipage}{%
279 \reledmac@error{Fail to patch \string\@iiiminipage\space command.}\@ehc}
%
280 }%
281 %

```

```

\led@error@fail@patch@endminipage82 \newcommand{\led@error@fail@patch@endminipage}{%
283 \reledmac@error{Fail to patch \string@endminipage\space command.}\@ehc}%
284 }%
285 %

```

```

\led@warning@hsizeX@deprecated86 \newcommand{\led@warning@hsizeX@deprecated}{%
287 \reledmac@warning{\string\hsizeX\space command deprecated, use \string\
widthX\space instead.}%
288 }%
289 %

```

```

\led@warning@Xhsize@deprecated90 \newcommand{\led@warning@Xhsize@deprecated}{%
291 \reledmac@warning{\string\Xhsize\space command deprecated, use \string\
Xwidth\space instead.}%
292 }%
293 %

```

II.8 Gobbling

Here, we define some commands which gobble their arguments.

```

\@gobblethree94 \providecommand*\@gobblethree}[3]{}
\@gobblefour95 \providecommand*\@gobblefour}[4]{}
\@gobblefive96 \providecommand*\@gobblefive}[5]{}
297 %

```

II.9 Miscellaneous commands

```

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

```



```

298 \ifledfinal
299   \newcommand*{\showlemma}[1]{#1}
300 \else
301   \newcommand*{\showlemma}[1]{\underline{#1}}
302 \fi
303
304 %

```

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

Initialize it as `\empty`.

```

305 \let\linenumberlist=\empty
306
307 %

```

`\@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).

```

308 \newcount\@l@tempcnta \newcount\@l@tempcntb
309 %

```

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@dpageing`

`\ifl@dprintingpages`

`\ifl@dprintingcolumns`

`\ifpst@rtedL`

`\l@dnumpstartsl`

```

310 \newif\ifl@dpairing
311 \newif\ifl@dpageing%
312 \newif\ifl@dprintingpages%
313 \newif\ifl@dprintingcolumns%
314 \newif\ifpst@rtedL
315 \newcount\l@dnumpstartsl
316 %

```

`\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@`

```

317 \newif\ifledRcol
318 \newif\ifledRcol@
319 %

```

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

```

320 \newif\ifnumberingR
321 %

```

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

```
322 \newif\ifXnote@%
323 %
```

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.

```
324 \providebool{indtl@innote}%
325 \providebool{indtl@notenumber}%
326 %
```

III Sectioning commands

`\section@num` You use `\beginnumbering` and `\endnumbering` to begin and end a line-numbered section of the text; the pair of commands may be used as many times as you like within one document to start and end multiple, separately line-numbered sections. \TeX will maintain and display a ‘section number’ as a count named `\section@num` that counts how many `\beginnumbering` and `\resumenumbering` commands have appeared; it need not be related to the logical divisions of your text.

`\extensionchars` Each section will read and write an associated ‘line-list file’, containing information used to do the numbering; the file will be called `\jobname\section@num`, 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.

```
327 \newcount\section@num
328 \section@num=0
329 \let\extensionchars=\empty
330 %
```

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

```
331 \newif\ifnumbering
332 %
```

`\beginnumbering` `\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.

```

333 \newcommand*{\beginnumbering}{%
334   \ifnumbering
335     \led@err@NumberingStarted
336   \endnumbering
337 \fi
338 \global\numberingtrue
339 \global\advance\section@num \@ne
340 \initnumbering@reg
341 \message{Section \the\section@num }%
342 \line@list@stuff{\jobname.\extensionchars\the\section@num}%
343 \l@dend@stuff
344 \setcounter{pstart}{1}
345 \ifl@dpairing
346   \global\l@dnumpstartsL \z@
347   \global\pst@rtedLfalse
348 %

```

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.

```

349 \else
350   \begingroup
351   \global\@afterindenttrue%In order to reestablish normal feature if the \
begingroup was not here
352   \initnumbering@quote
353   \ifwidthliketwocolumns%
354     \csuse{setwidthliketwocolumns@\columns@position}%
355     \csuse{setpositionliketwocolumns@\columns@position}%
356   \fi%

```

```

357 \fi
358 \gdef\eled@sections@@{}%
359 \if@noeled@sec\else%
360 \makeatletter\inputIfFileExists{\jobname.eledsec\the\section@num}{-}{-}\
makeatother%
361 \immediate\openout\eled@sectioning@out=\jobname.eledsec\the\section@num
\relax%
362 \fi%
363 }
364 \newcommand*\initnumbering@reg{%
365 \global\pst@rtedLfalse
366 \global\l@dnumpststartsL \z@
367 \global\absline@num \z@
368 \gdef\normal@page@break{}
369 \gdef\l@prev@pb{}
370 \gdef\l@prev@nopb{}
371 \global\line@num \z@
372 \global\subline@num \z@
373 \global\@lock \z@
374 \global\sub@lock \z@
375 \global\sublines@false
376 \global\let\next@page@num=\relax
377 \global\let\sub@change=\relax
378 \resetprevline@
379 \resetprevpage@num
380 }
381
382 %

```

`\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.

```

383 \def\endnumbering{%
384 \ifnumbering
385 \global\numberingfalse
386 \normal@pars
387 \ifnum\l@dnumpststartsL=0%
388 \led@err@NumberingWithoutPstart%
389 \fi%
390 \ifl@dpairing
391 \global\pst@rtedLfalse
392 \else
393 \ifx\insertlines@list\empty\else
394 \global\noteschanged@true
395 \fi
396 \ifx\line@list\empty\else
397 \global\noteschanged@true
398 \fi
399 \fi

```

```

400 \ifnoteschanged@
401 \led@mess@NotesChanged
402 \fi
403 \else
404 \led@err@NumberingNotStarted
405 \fi
406 \autoparfalse
407 \if@noeled@sec\else%
408 \immediate\closeout\eled@sectioning@out%
409 \fi%
410 \ifl@dpairing\else
411 \global\l@dnumstartsL=\z@%
412 \endgroup
413 \fi
414 }
415 %

```

`\pausenumbering` The `\pausenumbering` macro is just the same as `\endnumbering`, but with the `\ifnumbering` flag set to true, to show that numbering continues across the gap.²³

`\resumenumbering`

```

416 \newcommand{\pausenumbering}{%
417 \ifautopar\global\autopar@pausetrue\fi%
418 \endnumbering\global\numberingtrue}
419 %

```

The `\resumenumbering` macro is a bit more involved, but not much. It does most of the same things as `\beginnumbering`, but without resetting the various counters. Note that no check is made by `\resumenumbering` to ensure that `\pausenumbering` was actually invoked.

```

420 \newcommand*{\resumenumbering}{%
421 \ifnumbering
422 \ifautopar@pause\autopar\fi
423 \global\pst@rtedLtrue
424 \global\advance\section@num \@ne
425 \led@mess@SectionContinued{\the\section@num}%
426 \line@list@stuff{\jobname.\extensionchars\the\section@num}%
427 \l@dend@stuff
428 \ifl@dpairing\else%
429 \begingroup%
430 \initnumbering@quote%
431 \ifwidthliketwocolumns%
432 \csuse{setwidthliketwocolumns@\columns@position}%
433 \csuse{setpositionliketwocolumns@\columns@position}%
434 \fi%
435 \fi%
436 \else
437 \led@err@NumberingShouldHaveStarted
438 \endnumbering

```

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

```

439 \beginnumbering
440 \fi}
441
442
443 %

```

IV List macros

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

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

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

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

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

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

```

444 \newcommand*\list@create}[1]{%
445   \global\let#1=\empty%
446 }%
447 %

```

\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 .

```

448 \newcommand*\list@clear}[1]{%
449   \global\let#1=\empty%
450 }
451 %

```

\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`.

```

452 \newtoks\led@toksa \newtoks\led@toksb
453 \global\led@toksa={\}
454 \long\def\xright@appenditem#1\to#2{%
455   \global\led@toksb=\expandafter{#2}%

```

```

456 \xdef#2{\the\led@toksb\the\led@toksa\expandafter{#1}}%
457 \global\led@toksb={}}
458 %

```

\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.

```

459 \long\def\xleft@appenditem#1\to#2{%
460 \global\led@toksb=\expandafter{#2}%
461 \xdef#2{\the\led@toksa\expandafter{#1}\the\led@toksb}%
462 \global\led@toksb={}}
463 %

```

\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.

```

464 \def\gl@p#1\to#2{\expandafter\gl@poff#1\gl@poff#1#2}
465 \long\def\gl@poff\#1#2\gl@poff#3#4{\gdef#4{#1}\gdef#3{#2}}
466
467 %

```

V Line counting

V.1 Choosing the system of lineation

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

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

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

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

```

468 \newif\ifbypage@
469 \newif\ifbypstart@
470 %

```

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@R71 \newif\ifbypage@R
\ifbypstart@R72 \newif\ifbypstart@R
473 %

```

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

```
474 \newcommand*{\lineation}[1]{%
475 %
```

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

```
476 \ifnumbering
477 \led@err@LineationInNumbered
478 \else
479 %
```

If the argument is page.

```
480 \def\@tempa{#1}\def\@tempb{page}%
481 \ifx\@tempa\@tempb
482 \global\bypage@true
483 \global\bypstart@false
484 \unless\ifnocritical@%
485 \Xpstart[] [false]%
486 \fi%
487 %
```

If the argument is pstart.

```
488 \else
489 \def\@tempb{pstart}%
490 \ifx\@tempa\@tempb
491 \global\bypage@false
492 \global\bypstart@true
493 \unless\ifnocritical@%
494 \Xpstart%
495 \fi%
496 %
```

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

```
497 \else
498 \def\@tempb{section}
499 \ifx\@tempa\@tempb
500 \global\bypage@false
501 \global\bypstart@false
502 \unless\ifnocritical@%
503 \Xpstart[] [false]%
504 \fi%
505 %
```

In other case, it is an error.

```
506 \else
507 \led@warn@BadLineation
508 \fi
509 \fi
510 \fi
```



```

511 \fi}}
512 %

```

V.2 Line number margin

`\linenummargin` `\linenummargin{<word>}` specify which margin line numbers are in; it takes one argument, a string, which value can be left ; right; inner or outer.

`\line@margin` The selection is recorded in the count `\line@margin`: 0 for left, 1 for right, 2 for outer, and 3 for inner.

`\l@dgetline@margin`

```

513 \newcount\line@margin
514
515 \newcommand*{\linenummargin}[1]{%
516   \l@dgetline@margin{#1}%
517   \ifnum\@l@tempcntb>\m@ne
518     \ifledRcol
519       \global\line@marginR=\@l@tempcntb
520       \led@warn@setting@in@rightside{\linenummargin}%
521     \else
522       \global\line@margin=\@l@tempcntb
523     \fi
524   \fi}}
525
526 \newcommand*{\l@dgetline@margin}[1]{%
527   \def\@tempa{#1}\def\@tempb{left}%
528   \ifx\@tempa\@tempb
529     \@l@tempcntb \z@
530   \else
531     \def\@tempb{right}%
532     \ifx\@tempa\@tempb
533       \@l@tempcntb \@ne
534     \else
535       \def\@tempb{outer}%
536       \ifx\@tempa\@tempb
537         \@l@tempcntb \tw@
538       \else
539         \def\@tempb{inner}%
540         \ifx\@tempa\@tempb
541           \@l@tempcntb \thr@@
542         \else
543           \led@warn@BadLinenummargin
544           \@l@tempcntb \m@ne
545         \fi
546       \fi
547     \fi
548   \fi}
549
550 %

```

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.

```
551 \newcounter{firstlinenum}
552   \setcounter{firstlinenum}{5}
553 \newcounter{linenumincrement}
554   \setcounter{linenumincrement}{5}
555 %
```

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

```
556 \newcounter{firstsublinenum}
557   \setcounter{firstsublinenum}{5}
558 \newcounter{sublinenumincrement}
559   \setcounter{sublinenumincrement}{5}
560
561 %
```

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

```
562 \newcommand*{\firstlinenum}[1]{%
563   \ifledRcol%
564     \setcounter{firstlinenumR}{#1}%
565     \led@warn@setting@in@rightside{\firstlinenum}%
566   \else%
567     \setcounter{firstlinenum}{#1}%
568   \fi%
569 }
570
571 \newcommand*{\linenumincrement}[1]{%
572   \ifledRcol%
573     \setcounter{linenumincrementR}{#1}%
574     \led@warn@setting@in@rightside{\linenumincrement}%
575   \else%
576     \setcounter{linenumincrement}{#1}%
577   \fi%
578 }
579
580 \newcommand*{\firstsublinenum}[1]{%
581   \ifledRcol%
582     \setcounter{firstsublinenumR}{#1}%
583     \led@warn@setting@in@rightside{\firstsublinenum}%
584   \else%
585     \setcounter{firstsublinenum}{#1}%
586   \fi%
```

```

586 }
587 \newcommand*{\sublinenumincrement}[1]{%
588   \ifledRcol%
589     \setcounter{sublinenumincrementR}{#1}%
590     \led@warn@setting@in@rightside{\sublinenumincrement}%
591   \else%
592     \setcounter{sublinenumincrement}{#1}%
593   \fi%
594 }
595
596 %

```

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.

```

597 \newcount\lock@disp
598 \newcommand{\lockdisp}[1]{%
599   \l@dgetlock@disp{#1}%
600   \ifnum\l@dtempcntb>\m@ne
601     \global\lock@disp=\l@dtempcntb
602   \else
603     \led@warn@BadLockdisp
604   \fi}}
605 \newcommand*{\l@dgetlock@disp}[1]{
606   \def\@tempa{#1}\def\@tempb{first}%
607   \ifx\@tempa\@tempb
608     \l@dtempcntb \z@
609   \else
610     \def\@tempb{last}%
611     \ifx\@tempa\@tempb
612       \l@dtempcntb \@ne
613     \else
614       \def\@tempb{all}%
615       \ifx\@tempa\@tempb
616         \l@dtempcntb \tw@
617       \else
618         \l@dtempcntb \m@ne
619       \fi
620     \fi
621   \fi}
622
623 %

```

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

```

624 \newcount\sublock@disp
625 \newcommand{\sublockdisp}[1]{\%
626   \l@getlock@disp{#1}%
627   \ifnum\l@dttempcntb>\m@ne
628     \global\sublock@disp=\l@dttempcntb
629   \else
630     \led@warn@BadSublockdisp
631   \fi}}
632
633 %

```

V.5 Line number style

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

```

634 \newcommand*{\linenumberstyle}[1]{\%
635   \def\linenumrep##1{\@nameuse{#1}{##1}}}
636 \newcommand*{\sublinenumberstyle}[1]{\%
637   \def\sublinenumrep##1{\@nameuse{#1}{##1}}}
638 %

```

Initialise the number styles to arabic.

```

639 \linenumberstyle{arabic}
640 \let\linenumr@p\linenumrep
641 \sublinenumberstyle{arabic}
642 \let\sublinenumr@p\sublinenumrep
643
644 %

```

V.6 Line number printing

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

Whatever these macros output gets printed in a box that will be put into the appropriate margin without any space between it and the line of text. You will generally want a kern between a line number and the text, and `\linenumsep` is provided as a

standard way of storing its size. Line numbers are usually printed in a smaller font, and `\numlabfont` is provided as a standard name for that font. When called, these macros will be executed within a group, so font changes and the like will remain local.

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

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

```

645 \newlength{\linenumsep}
646 \setlength{\linenumsep}{1pc}
647 \newcommand*\numlabfont{\normalfont\scriptsize}
648 \newcommand*\ledlinenum{%
649   \bgroup%
650   \ifluatex%
651     \texdir TLT%
652   \fi%
653   \numlabfont\linenumrep{\line@num}%
654   \ifsublines@
655     \ifnum\subline@num>0\relax
656       \unskip%
657       \Xsublinesep@side%
658       \sublinenumrep{\subline@num}%
659     \fi
660     \fi%
661     \egroup%
662 }%
663
664 \newcommand*\leftlinenum{%
665   \ledlinenum
666   \kern\linenumsep}
667 \newcommand*\rightlinenum{%
668   \kern\linenumsep
669   \ledlinenum}
670
671 %

```

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`.

```
672 \newcount\line@num
673 %
```

`\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.

```
674 \newcount\subline@num
675 %
```

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

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

```
676 \newif\ifsublines@
677 %
```

`\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.

```
678 \newcount\absline@num
679 %
```

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.

```

680 \newcount\@lock
681 \newcount\sub@lock
682 %

```

V.9 Line number associated to lemma

```

\line@list
\insertlines@list
\actionlines@list
\actions@list

```

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

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

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

```
23|35|0|24|3|0|OT1/cm/r/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:


```

683 \list@create{\line@list}
684 \list@create{\insertlines@list}
685 \list@create{\actionlines@list}
686 \list@create{\actions@list}
687
688 %

```

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

```

\endpage@num
\endline@num
\endsubline@num
689 \newcount\page@num
690 \newcount\endpage@num
691 \newcount\endline@num
692 \newcount\endsubline@num
693 %

```

`\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.

```

694 \newif\ifnoteschanged@
695 %

```

`\resetprevline@` Inside the apparatus, at each note, the line number is stored in a macro called `\prevlineX`, where X is the letter of the current series. This macro is called when using `\Xnumberonlyfirstinline`. This macro must be reset at the same time as the line number. The `\resetprevline@` does this resetting for every series.

```

\resetprevline@% \newcommand*{\resetprevline@}{%
697 \def\do##1{\global\csundef{prevline##1}}%
698 \dolistloop{\@series}%
699 }
700 %

```

`\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@01 \newcommand*\resetprevpage@num}{%
702   \def\do##1{\ifcsdef{prevpage##1@num}{\global\csname prevpage##1@num\
endcsname=0}{}}%
703   \dolistloop{\@series}%
704 }
705 %

```

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.

```

706 \newread\@inputcheck
707 \newcommand*\read@linelist}[1]{%
708   \ifledRcol%
709     \list@clearing@regR%
710   \else%
711     \list@clearing@reg%
712   \fi%
713 %

```

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).

```

714   \list@clear{\maxlinesinpar@list}
715 %

```

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

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

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

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

```

716 \get@linelistfile{#1}%
717 \endgroup
718 %

```

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.

```

719 \ifledRcol
720   \global\page@numR=\m@ne
721   \ifx\actionlines@listR\empty
722     \gdef\next@actionlineR{1000000}%
723   \else
724     \gl@p\actionlines@listR\to\next@actionlineR
725     \gl@p\actions@listR\to\next@actionR
726   \fi
727 \else
728   \global\page@num=\m@ne
729   \ifx\actionlines@list\empty
730     \gdef\next@actionline{1000000}%
731   \else
732     \gl@p\actionlines@list\to\next@actionline
733     \gl@p\actions@list\to\next@action
734   \fi
735 \fi
736 }
737 %

```

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

```

738 \newcommand*{\list@clearing@reg}{%
739   \list@clear{\line@list}%
740   \list@clear{\insertlines@list}%
741   \list@clear{\actionlines@list}%
742   \list@clear{\actions@list}%
743   \list@clear{\linesinpar@listL}%
744   \list@clear{\linesonpage@listL}%
745   }%
746 %

```

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

```

747 \newcommand*{\get@linelistfile}[1]{%
748   \InputIfFileExists{#1}{%
749     \global\noteschanged@false
750     \begingroup
751       \catcode`\[=1 \catcode`\]=2
752       \makeatletter \catcode`\^^M=9}{%
753       \led@warn@NoLineFile{#1}%
754       \global\noteschanged@true
755       \begingroup}%
756   }
757 %
758 %

```

This version of `\read@linelist` creates list macros containing data for the entire section, so they could get rather large. It would be no more difficult to read the line-list file incrementally rather than all at once: we could read, at the start of each paragraph, only the commands relating to that paragraph. But this would require that we have two line-lists open at once, one for reading, one for writing, and on systems without version numbers we would have to do some file renaming outside of \TeX for that to work. We have retained this slower approach to avoid that sort of hacking about, but have provided the `\pausenumbering` and `\resumenumbering` macros to help you if you run into macro memory limitations (see 4.2.7 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.

```

759 \newcommand{\line@list@version}[1]{%
760   \IfStrEq{#1}{\this@line@list@version}%
761   {}%
762   {\ifledRcol%
763     \led@warn@Obsolete{\jobname.\extensionchars\the\section@num}%
764     \else%
765     \led@warn@Obsolete{\jobname.\extensionchars\the\section@num}%
766     \fi%
767     \endinput%
768   }%
769 }%
770 %

```

`\@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`.

```

771
772 \newcommand*{\@nl}[2]{%
773   \fix@page{#1}%
774   \ifledRcol%
775     \@nl@regR%
776   \else%
777     \@nl@reg%
778   \fi%
779 }
780 \newcommand*{\@nl@reg}{%
781   \ifx\l@dchset@num\relax \else
782     \advance\absline@num \@ne
783     \set@line@action
784     \let\l@dchset@num=\relax
785     \advance\absline@num \m@ne
786     \advance\line@num \m@ne
787   \fi
788   %

```

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

```

789   \advance\absline@num \@ne
790     \ifx\next@page@num\relax \else
791       \page@action
792       \let\next@page@num=\relax
793     \fi
794     \ifx\sub@change\relax \else
795       \ifnum\sub@change>\z@
796         \sublines@true
797       \else
798         \sublines@false
799       \fi
800       \sub@action
801       \let\sub@change=\relax
802     \fi
803   %

```

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

```

804     \ifcase\@lock
805         \or
806             \@lock \tw@
807         \or \or
808             \@lock \z@
809     \fi
810     \ifcase\sub@lock
811         \or
812             \sub@lock \tw@
813         \or \or
814             \sub@lock \z@
815     \fi
816 %

```

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

```

817     \ifsublines@
818         \ifnum\sub@lock<\tw@
819             \advance\subline@num \@ne
820         \fi
821     \else
822         \ifnum\@lock<\tw@
823             \advance\line@num \@ne \subline@num \z@
824         \fi
825     \fi}
826
827 %

```

`\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`.

```

828 \newcount\last@page@num
829 \last@page@num=-10000
830
831 \newcommand*{\fix@page}[1]{%
832     \ifledRcol
833         \ifnum #1=\last@page@numR
834         \else
835             \ifbypage@R
836                 \line@numR \z@ \subline@numR \z@
837             \fi
838             \page@numR=#1\relax
839             \last@page@numR=#1\relax
840             \def\next@page@numR{#1}%
841         \fi
842     \else
843         \ifnum #1=\last@page@num
844         \else
845             \ifbypage@
846                 \line@num \z@ \subline@num \z@
847             \fi

```

```

848 \page@num=#1\relax
849 \last@page@num=#1\relax
850 \def\next@page@num{#1}%
851 \listxadd{\normal@page@break}{\the\absline@num}
852 \fi
853 \fi}
854 %

```

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

```

855 \newcommand*{\@pend}[1]{}
856 \newcommand*{\@pendR}[1]{}
857 \newcommand*{\@lopL}[1]{}
858 \newcommand*{\@lopR}[1]{}
859
860 %

```

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

```

861 \newcommand*{\sub@on}{\ifsublines@
862 \let\sub@change=\relax
863 \else
864 \def\sub@change{1}%
865 \fi}
866 \newcommand*{\sub@off}{\ifsublines@
867 \def\sub@change{-1}%
868 \else
869 \let\sub@change=\relax
870 \fi}
871
872 %

```

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

```

873
874 \newcommand*{\@adv}[1]{%
875 \ifsublines@
876 \ifledRcol
877 \advance\subline@numR by #1\relax
878 \ifnum\subline@numR<\z@
879 \led@warn@BadAdvancelineSubline
880 \subline@numR \z@
881 \fi
882 \else
883 \advance\subline@num by #1\relax

```

```

884     \ifnum\subline@num<\z@
885         \led@warn@BadAdvancelineSubline
886         \subline@num \z@
887     \fi
888 \fi
889 \else
890     \ifledRcol
891         \advance\line@numR by #1\relax
892         \ifnum\line@numR<\z@
893             \led@warn@BadAdvancelineLine
894             \line@numR \z@
895         \fi
896     \else
897         \advance\line@num by #1\relax
898         \ifnum\line@num<\z@
899             \led@warn@BadAdvancelineLine
900             \line@num \z@
901         \fi
902     \fi
903 \fi
904 \set@line@action}
905
906 %

```

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

```

907
908 \newcommand*{\@set}[1]{%
909     \ifledRcol
910         \ifsublines@
911             \subline@numR=#1\relax
912         \else
913             \line@numR=#1\relax
914         \fi
915         \set@line@action
916     \else
917         \ifsublines@
918             \subline@num=#1\relax
919         \else
920             \line@num=#1\relax
921         \fi
922         \set@line@action
923     \fi}
924
925 %

```

\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-ber change is to be done.

```

926
927 \newcommand*{\l@d@set}[1]{%
928   \ifledRcol
929     \line@numR=#1\relax
930     \advance\line@numR \@ne
931     \def\l@dchset@num{#1}
932   \else
933     \line@num=#1\relax
934     \advance\line@num \@ne
935     \def\l@dchset@num{#1}
936   \fi}
937 \let\l@dchset@num\relax
938
939 %

```

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

```

940
941 \newcommand*{\page@action}{%
942   \ifledRcol
943     \xright@appenditem{\the\absline@numR}\to\actionlines@listR
944     \xright@appenditem{\next@page@numR}\to\actions@listR
945   \else
946     \xright@appenditem{\the\absline@num}\to\actionlines@list
947     \xright@appenditem{\next@page@num}\to\actions@list
948   \fi}
949 %

```

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

```

950
951 \newcommand*{\set@line@action}{%
952   \ifledRcol
953     \xright@appenditem{\the\absline@numR}\to\actionlines@listR
954     \ifsublines@
955       \@l@tempcnta=-\subline@numR
956     \else
957       \@l@tempcnta=-\line@numR
958     \fi
959     \advance\@l@tempcnta by -5000\relax
960     \xright@appenditem{\the\@l@tempcnta}\to\actions@listR
961   \else
962     \xright@appenditem{\the\absline@num}\to\actionlines@list
963     \ifsublines@
964       \@l@tempcnta=-\subline@num
965     \else

```

```

966     \@l@tempcnta=-\line@num
967     \fi
968     \advance\@l@tempcnta by -5000\relax
969     \xright@appenditem{\the\@l@tempcnta}\to\actions@list
970   \fi}
971 %

```

\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.

```

972
973 \newcommand*{\sub@action}{%
974   \ifledRcol
975     \xright@appenditem{\the\absline@numR}\to\actionlines@listR
976     \ifsublines@
977       \xright@appenditem{-1001}\to\actions@listR
978     \else
979       \xright@appenditem{-1002}\to\actions@listR
980     \fi
981   \else
982     \xright@appenditem{\the\absline@num}\to\actionlines@list
983     \ifsublines@
984       \xright@appenditem{-1001}\to\actions@list
985     \else
986       \xright@appenditem{-1002}\to\actions@list
987     \fi
988   \fi}
989 %

```

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

Adding commands to the action list is slow, and it is very often the case that a lock-on command is immediately followed by a lock-off command in the line-list file, and therefore really does nothing. We use a look-ahead scheme here to detect such pairs, and add nothing to the line-list in those cases.

```

990 \newcommand*{\lock@on}{\futurelet\next\do@lockon}
991
992 \newcommand*{\do@lockon}{%
993   \ifx\next\lock@off
994     \global\let\lock@off=\skip@lockoff
995   \else
996     \ifledRcol
997       \do@lockonR
998     \else
999       \do@lockonL
1000   \fi
1001 \fi}

```

```

1002
1003
1004 \newcommand*{\do@lockonL}{%
1005   \xright@appenditem{\the\absline@num}\to\actionlines@list
1006   \ifsublines@
1007     \xright@appenditem{-1005}\to\actions@list
1008     \ifnum\sub@lock=\z@
1009       \sub@lock \@ne
1010     \else
1011       \ifnum\sub@lock=\thr@@
1012         \sub@lock \@ne
1013       \fi
1014     \fi
1015   \else
1016     \xright@appenditem{-1003}\to\actions@list
1017     \ifnum\@lock=\z@
1018       \@lock \@ne
1019     \else
1020       \ifnum\@lock=\thr@@
1021         \@lock \@ne
1022       \fi
1023     \fi
1024   \fi}
1025
1026 %

```

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

```

1027 \do@lockoff
1028 \do@lockoffL
1029 \skip@lockoff
1030 \newcommand*{\do@lockoffL}{%
1031   \xright@appenditem{\the\absline@num}\to\actionlines@list
1032   \ifsublines@
1033     \xright@appenditem{-1006}\to\actions@list
1034     \ifnum\sub@lock=\tw@
1035       \sub@lock \thr@@
1036     \else
1037       \sub@lock \z@
1038     \fi
1039   \else
1040     \xright@appenditem{-1004}\to\actions@list
1041     \ifnum\@lock=\tw@
1042       \@lock \thr@@
1043     \else
1044       \@lock \z@
1045     \fi
1046   \fi}
1047
1048 \newcommand*{\do@lockoff}{%
1049   \ifledRcol
1050     \do@lockoffR
1051   \else

```

```

1049 \do@lockoffL
1050 \fi}
1051 \newcommand*{\skip@lockoff}{\global\let\lock@off=\do@lockoff}
1052 \global\let\lock@off=\do@lockoff
1053
1054 %

```

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

```

1055 \newcommand*{\n@num}{%
1056 \ifledRcol%
1057 \xright@appenditem{\the\absline@numR}\to\actionlines@listR
1058 \xright@appenditem{-1007}\to\actions@listR
1059 \else%
1060 \xright@appenditem{\the\absline@num}\to\actionlines@list%
1061 \xright@appenditem{-1007}\to\actions@list%
1062 \fi%
1063 }%
1064
1065 %

```

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

```

1066 \newcommand*{\n@num@stanza}{%
1067 \ifledRcol%
1068 \xright@appenditem{\the\absline@numR}\to\actionlines@listR%
1069 \xright@appenditem{-1008}\to\actions@listR%
1070 \else%
1071 \xright@appenditem{\the\absline@num}\to\actionlines@list%%
1072 \xright@appenditem{-1008}\to\actions@list%
1073 \fi%
1074 }
1075 %

```

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

\hidenum

```

1076 \newif\ifl@dhidenumber
1077 \newcommand*{\hidenum}{%
1078 \ifledRcol%
1079 \write\linenum@outR{\string\hide@num}%
1080 \else%
1081 \write\linenum@out{\string\hide@num}%
1082 \fi%
1083 }%
1084 \newcommand*{\hide@num}{%
1085 \ifledRcol%
1086 \xright@appenditem{\the\absline@numR}\to\actionlines@listR%

```

```

1087 \xright@appenditem{-1009}\to\actions@listR%
1088 \else%
1089 \xright@appenditem{\the\absline@num}\to\actionlines@list%%
1090 \xright@appenditem{-1009}\to\actions@list%
1091 \fi%
1092 }
1093 %

```

\@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.

```

1094 \newcount\insert@count
1095 %

```

- #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.

```

1096 \newcommand*\dummy@ref}[2]{#2}
1097 %

```

\@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.

```

1098 \newcommand*\@ref}[2]{%
1099 \ifledRcol%
1100 \@ref@regR{#1}{#2}%
1101 \else%
1102 \@ref@reg{#1}{#2}%
1103 \fi%
1104 }%
1105 \newcommand*\@ref@reg}[2]{%
1106 \global\insert@count=#1\relax
1107 \global\advance\@edtext@level by 1%
1108 \loop\ifnum\insert@count>\z@
1109 \xright@appenditem{\the\absline@num}\to\insertlines@list
1110 \global\advance\insert@count \m@ne
1111 \repeat
1112 %

```

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.

```

1113 \begingroup
1114   \let\@ref=\dummy@ref
1115   \let\@lopL\@gobble
1116   \let\page@action=\relax
1117   \let\sub@action=\relax
1118   \let\set@line@action=\relax
1119   \let\@lab=\relax
1120   \let\@lemma=\relax%
1121   \let\@sw\@gobblethree%
1122   #2
1123   \global\endpage@num=\page@num
1124   \global\endline@num=\line@num
1125   \global\endsubline@num=\subline@num
1126 \endgroup
1127 %

```

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

```

1128   \xright@appenditem%
1129     {\the\page@num|\the\line@num|%
1130     \ifsublines@ \the\subline@num \else 0\fi|%
1131     \the\endpage@num|\the\endline@num|%
1132     \ifsublines@ \the\endsubline@num \else 0\fi}\to\line@list
1133 %

```

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.

```

1134   \expandafter\list@create\expandafter{\csname sw@list@edtext@tmp@\the\
1135   @edtext@level\endcsname}%
1136   \providebool{lemmacommand@\the\@edtext@level}%
1137   \boolfalse{lemmacommand@\the\@edtext@level}%
1138 %

```

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

```

1138   #2%
1139 %

```

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.

```

1140   \ifnum\@edtext@level>0%
1141     \def\create@this@edtext@level{\expandafter\list@create\expandafter{\
1142     csname sw@list@edtext@\the\@edtext@level\endcsname}}%
1143     \ifcsundef{sw@list@edtext@\the\@edtext@level}{\create@this@edtext@level
1144     }{}%
1145     \letcs{\@tmp}{sw@list@edtext@\the\@edtext@level}%
1146     \letcs{\@tmpp}{sw@list@edtext@tmp@\the\@edtext@level}%
1147     \xright@appenditem{\expandonce\@tmpp}\to\@tmp%
1148     \global\cslet{sw@list@edtext@\the\@edtext@level}{\@tmp}%

```

```

1147 \fi%
1148 %

Decrease edtext level counter.

1149 \global\advance\@edtext@level by -1%
1150 %

1151 }
1152
1153 %

```

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`.

```

1154 \newwrite\linenum@out
1155 %

```

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

We can manage pretty well by means of the `\iffirst@linenum@out@` flag; its in-
elegant name suggests the nature of the problem that made its creation necessary. It is
set to be `true` before any `\linenum@out` file is opened. When such a file is opened for
the first time, it is done using `\immediate`, so that it will at once be safe for the output
routine to write to it; we then set this flag to `false`.

```

1156 \newif\iffirst@linenum@out@
1157 \first@linenum@out@true
1158 %

```

`\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.

```

1159 \newcommand{\this@line@list@version}{5}%
1160 %

```

\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.

```
1161 \newcommand*{\line@list@stuff}[1]{%
1162 %
```

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

```
1163 \read@linelist{#1}%
1164 %
```

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.

```
1165 \iffirst@linenum@out@
1166 \immediate\closeout\linenum@out%
1167 \global\first@linenum@out@false%
1168 \immediate\openout\linenum@out=#1\relax%
1169 \immediate\write\linenum@out{\string\line@list@version{\
this@line@list@version}}%
1170 \ifl@dpaging%
1171 \immediate\write\linenum@out{\string\@par@sync@option{\
@par@this@sync@option}}%
1172 \fi%
1173 \else
1174 %
```

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.

```
1175 \if@minipage%
1176 \leavevmode%
1177 \fi%
1178 \closeout\linenum@out%
1179 \openout\linenum@out=#1\relax%
1180 \fi}
1181
1182 %
```

\new@line The `\new@line` macro sends the `\@n1` command to the line-list file, to mark the start of a new text line, and its page number.

```
1183 \newcommand*{\new@line}{%
1184 \IfStrEq{\led@pb@setting}{after}%
1185 {\xifinlist{\the\absline@num}{\l@prev@nopb}%
1186 {\xifinlist{\the\absline@num}{\normal@page@break}%
1187 {\numgdef{\@next@page}{\c@page+1}%
1188 \write\linenum@out{\string\@n1[\@next@page][\@next@page]}%
1189 }%
}
```



```

1190     {\write\linenum@out{\string\@nl[\the\c@page][\thepage]}}}%
1191   }%
1192   {\write\linenum@out{\string\@nl[\the\c@page][\thepage]}}}%
1193 }%
1194 \IfStrEq{\led@pb@setting}{before}%
1195   {\numdef{\next@absline}{\the\absline@num+1}%
1196   \xifinlist{\next@absline}{\l@prev@nopb}%
1197     {\xifinlist{\the\absline@num}{\normal@page@break}%
1198     {\numgdef{\nc@page}{\c@page+1}%
1199     \write\linenum@out{\string\@nl[\nc@page][\nc@page]}}%
1200     }%
1201     {\write\linenum@out{\string\@nl[\the\c@page][\thepage]}}}%
1202   }%
1203   {\write\linenum@out{\string\@nl[\the\c@page][\thepage]}}}%
1204 }%
1205 {}%
1206 \IfStrEqCase{\led@pb@setting}{{before}{\relax}{after}{\relax}}{\write\linenum@out{\string\@nl[\the\c@page][\thepage]}}%
1207 }
1208
1209 %

```

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

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

\flag@end

```

1210 \newif\if@noneed@Footnote%
1211
1212 \newcommand*{\flag@start}{%
1213   \ifledRcol%
1214     \edef\next{\write\linenum@outR{%
1215       \string\@ref[\the\insert@countR] []}}%
1216     \next%
1217     \ifnum\insert@countR<1%
1218       \if@noneed@Footnote\else%
1219         \led@err@EdtextWithoutFootnote%
1220       \fi%
1221     \fi%
1222   \else%
1223     \edef\next{\write\linenum@out{%
1224       \string\@ref[\the\insert@count] []}}%
1225     \next%
1226     \ifnum\insert@count<1%
1227       \if@noneed@Footnote\else%
1228         \led@err@EdtextWithoutFootnote%
1229       \fi%
1230     \fi%

```

```

1231 \fi}%
1232
1233 %

```

\startsub **\endsub** **\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.

```

1234
1235
1236 \newcommand*{\startsub}{\dimen0\lastskip
1237 \ifdim\dimen0>Opt \unskip \fi
1238 \ifledRcol \write\linenum@outR{\string\sub@on}%
1239 \else \write\linenum@out{\string\sub@on}%
1240 \fi
1241 \ifdim\dimen0>Opt \hskip\dimen0 \fi}
1242 \def\endsub{\dimen0\lastskip
1243 \ifdim\dimen0>Opt \unskip \fi
1244 \ifledRcol \write\linenum@outR{\string\sub@off}%
1245 \else \write\linenum@out{\string\sub@off}%
1246 \fi
1247 \ifdim\dimen0>Opt \hskip\dimen0 \fi}
1248
1249 %

```

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

```

1250 \newcommand*{\advanceline}[1]{\leavevmode%
1251 \ifledRcol \write\linenum@outR{\string\@adv[#1]}%
1252 \else \write\linenum@out{\string\@adv[#1]}%
1253 \fi%
1254 }
1255 %

```

\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.

```

1256
1257 \newcommand*{\setline}[1]{%

```

```

1258 \leavevmode%
1259 \ifnum#1<\z@
1260 \led@warn@BadSetline
1261 \else
1262 \ifledRcol \write\linenum@outR{\string\@set[#1]}%
1263 \else \write\linenum@out{\string\@set[#1]}%
1264 \fi
1265 \fi}
1266
1267 %

```

\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.

```

1268
1269 \newcommand*{\setlinenum}[1]{%
1270 \ifnum#1<\z@
1271 \led@warn@BadSetlinenum
1272 \else
1273 \ifledRcol \write\linenum@outR{\string\l@d@set[#1]}
1274 \else \write\linenum@out{\string\l@d@set[#1]} \fi
1275 \fi}
1276
1277 %

```

\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.

```

1278
1279 \newcommand*{\startlock}{%
1280 \ifledRcol \write\linenum@outR{\string\lock@on}%
1281 \else \write\linenum@out{\string\lock@on}%
1282 \fi}
1283 \def\endlock{%
1284 \ifledRcol \write\linenum@outR{\string\lock@off}%
1285 \else \write\linenum@out{\string\lock@off}%
1286 \fi}
1287 %

```

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

```

1288 \l@dskipnumbertrue
1289 \l@dskipnumberfalse
1290 \skipnumbering
1291 \leavevmode%
1292 \ifledRcol%
1293 \ifinstanza%
1294 \write\linenum@outR{\string\n@num@stanza}%

```

```

1295 \else%
1296   \write\linenum@outR{\string\n@num}%
1297   \fi%
1298   \advanceline{-1}%
1299 \else%
1300   \ifistanza%
1301     \write\linenum@out{\string\n@num@stanza}%
1302   \else%
1303     \write\linenum@out{\string\n@num}%
1304   \fi%
1305   \advanceline{-1}%
1306 \fi%
1307 }%
1308
1309 %

```

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

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

VI.1 `\edtext` itself

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

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

`\end@lemmas` is intended for the few things that need to be associated with the end of the lemma, like footnote numbers. Such numbers are not implemented in the current version, and indeed no use is currently made of `\end@lemmas` or of the `\aftergroup` trick. The general approach would be to define a macro to be used within the second argument of `\edtext` that would add the appropriate command to `\end@lemmas`.

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

```
1310 \list@create{\end@lemmas}
1311 %
```

`\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.

```

1312 \newcommand{\dummy@edtext}[2]{#1}
1313 %

```

`\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`.

```

1314 \newcommand{\dummy@edtext@showlemma}[2]{\showlemma{#1}}%
1315 %

```

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 `\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

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

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.)

```

1316 \newcommand*{\no@expands}{%
1317   \let\select@lemmafnt=0%
1318   \let\startsub=\relax \let\endsub=\relax
1319   \let\startlock=\relax \let\endlock=\relax
1320   \let\edlabel=\@gobble
1321   \let\setline=\@gobble \let\advanceline=\@gobble
1322   \let\sameword\sameword@inedtext%
1323   \let\edtext=dummy@edtext
1324   \l@dtabnoexpands
1325   \morenoexpands}
1326 \let\morenoexpands=\relax
1327
1328 %

```

\@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.

```

1329 \newcommand{\@tag}{}
1330 %

```

\@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.

```

1331 \newcount\@edtext@level%
1332 \@edtext@level=0%
1333 %

```

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

```

1334 \newcounter{edtext}
1335 \renewcommand{\theedtext}{edtxt@arabic{edtext}}%
1336 %

```

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

```

1337 \newcommand{\edtext}[2]{\leavevmode%
1338 %

```

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

```

1339   \ifnumberedpar%
1340   %

```

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

```
1341 \global\advance\@edtext@level by 1%
1342 %
```

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

```
1343 \stepcounter{edtext}%
1344 %
```

By default, we do not use `\lemma`

```
1345 \global\@lemmacommand@false%
1346 %
```

```
1347 \begingroup%
1348 %
```

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`.

```
1349 \ifledRcol%
1350 \ifcsvoid{sw@list@edtextR@the\@edtext@level}%
1351 {\global\let\sw@inthisedtext\empty}%
1352 {\expandafter\gl@p\csname sw@list@edtextR@the\@edtext@level\
endcsname\to\sw@inthisedtext}%
1353 \else%
1354 \ifcsvoid{sw@list@edtext@the\@edtext@level}%
1355 {\global\let\sw@inthisedtext\empty}%
1356 {\expandafter\gl@p\csname sw@list@edtext@the\@edtext@level\
endcsname\to\sw@inthisedtext}%
1357 \fi%
1358 %
```

`\@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.

```
1359 \global\renewcommand{\@tag}{%
1360 \no@expands #1%
1361 }%
1362 %
```

`\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`.


```

1363 \set@line%
1364 %

```

`\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`.

```

1365 \ifledRcol \global\insert@countR \z@%
1366 \else \global\insert@count \z@ \fi%
1367 %

```

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.

```

1368 \ignorespaces #2\relax%
1369 %

```

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

```

1370 \@ifundefined{xpg@main@language}{%if not polyglossia
1371 \flag@start}%
1372 {\ifRTL\flag@end\else\flag@start\fi%
1373 }%
1374 %

```

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

```

1375 \if@lemmacommand%
1376 \ifledRcol%
1377 \write\linenum@outR{\string\@lemma}%
1378 \else%
1379 \write\linenum@out{\string\@lemma}%
1380 \fi%
1381 \fi%
1382 %

```

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.

```

1383 \endgroup%
1384 \ifdef\hypertarget{%
1385 {%
1386 \csetdef{thisedtext@the\@edtext@level}{\theedtext}%

```

```

1387 \Hy@raisedlink@left{\hypertarget{\csuse{thisedtext@the\
@edtext@level}:start}{}}%
1388 \showlemma{#1}%
1389 \Hy@raisedlink{\hypertarget{\csuse{thisedtext@the\@edtext@level}:
end}{}}%
1390 }%
1391 {%
1392 \showlemma{#1}%
1393 }%
1394 %

```

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.

```

1395 \ifx\end@lemmas\empty \else%
1396 \glp@end@lemmas\to\x@lemma%
1397 \x@lemma%
1398 \global\let\x@lemma=\relax%
1399 \fi%
1400 \ifundefined{xpg@main@language}{%if not polyglossia
1401 \flag@end}%
1402 {%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).
1403 }%
1404 %

```

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`.

```

1405 \global\@noneed@Footnotefalse%
1406 \global\advance\@edtext@level by -1%
1407 \global\@lemmacommand@false%
1408 %

```

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

```

1409 \else%
1410 \showlemma{#1} (\textbf{\textsc{Edtext outside numbered paragraph}})\
led@err@edtextoutsidestart%
1411 \fi%
1412 }%
1413
1414 \newcommand*{\flag@end}{%

```

```

1415 \ifledRcol%
1416 \write\linenum@outR{}}}%
1417 \else%
1418 \write\linenum@out{}}}%
1419 \fi}%
1420
1421 %

```

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

```

1422 \newif\ifnumberline
1423 \numberlinetrue
1424 %

```

\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.

```

1425 \newcommand*{\set@line}{%
1426 \ifledRcol
1427 \ifx\line@listR\empty
1428 \global\noteschanged@true
1429 \xdef\l@d@nums{000|000|000|000|000|000|\edfont@info}%
1430 \else
1431 \glp\line@listR\to\@tempb
1432 \xdef\l@d@nums{\@tempb|\edfont@info}%
1433 \global\let\@tempb=\undefined
1434 \fi
1435 \else
1436 \ifx\line@list\empty
1437 \global\noteschanged@true
1438 \xdef\l@d@nums{000|000|000|000|000|000|\edfont@info}%
1439 \else
1440 \glp\line@list\to\@tempb
1441 \xdef\l@d@nums{\@tempb|\edfont@info}%
1442 \global\let\@tempb=\undefined
1443 \fi
1444 \fi}
1445
1446 %

```

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

```

1447 \newcommand*{\edfont@info}{\f@encoding/\f@family/\f@series/\f@shape}
1448
1449 %

```

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. 112).

```

1450 \newcommand*{\lemma}[1]{%
1451   \global\@lemmacommand@true%
1452   \global\renewcommand{\@tag}{%
1453     \no@expands #1%
1454   }%
1455   \ignorespaces%
1456 }%
1457 %

```

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

```

1458 \newcommand{\@lemma}{%
1459   \booltrue{lemmacommand@the\@edtext@level}%
1460 }%
1461 %

```

\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.

```

1462 \newif\if@lemmacommand@%
1463 %

```

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. 87): 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.

```

1464 \newcommand*{\linenum}[1]{%
1465   \xdef\@tempa{#1} \noexpand\\l@d@nums}%
1466   \global\let\l@d@nums=\empty
1467   \expandafter\line@set\@tempa\\\ignorespaces}
1468 %

```

\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.

```

1469 \def\line@set#1|#2|#3|#4|{%
1470   \gdef\@tempb{#1}%
1471   \ifx\@tempb\empty
1472     \l@d@add{#3}%
1473   \else
1474     \l@d@add{#1}%
1475   \fi
1476   \gdef\@tempb{#4}%
1477   \ifx\@tempb\empty\else
1478     \l@d@add{|}\line@set#2|#4|}%
1479   \fi}
1480 %

```

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

```

1481 \newcommand{\l@d@add}[1]{\xdef\l@d@nums{\l@d@nums#1}}
1482
1483 %

```

VI.4 Lemma disambiguation

The mechanism which counts the occurrence of a same word in a same line is quite complex, because, when \LaTeX reads a command between a `\pstart` and a `\pend`, it does not know yet which are the line numbers.

The general mechanism is the following:

- **At the first run**, each `\sameword` command increments an `etoolbox` counter the name of which contains the argument of the `\sameword` commands.
- Then this counter, associated with the argument of `\sameword` is stored, with the `\@sw` command, in the auxiliary file of the current `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

Because there is a bug with `\detokenize` and \LaTeX when using non BMP characters²⁶, we detokenize only for not \LaTeX engines. In any case, in \LaTeX , 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.

```

1484 \newcommand{\get@sw@txt}[1]{%
1485   \ifxetex%
1486     \xdef\sw@txt{#1}%
1487   \else%
1488     \expandafter\xdef\expandafter\sw@txt\expandafter{\detokenize{#1}}%
1489   \fi%
1490 }%
1491 %

```

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

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

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

```
1492 \newcommand{\sameword}[2][1,usedefault]{%
1493     \leavevmode%
1494     \get@sw@txt{#2}%
1495 }
```

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

```
1496 \unless\ifledRcol%
1497     \csnumgdef{sw@sw@txt}{\csuse{sw@sw@txt}+1}%
1498 %
```

Then, write its value to the numbered file.

```
1499     \protected@write\linenum@out{\string\@sw{sw@txt}{\csuse{sw@sw@txt}
1500 }}{#1}}%
1501 %
```

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

```
1501 \else%
1502     \csnumgdef{sw@sw@txt}{\csuse{sw@sw@txt}+1}%
1503     \protected@write\linenum@outR{\string\@sw{sw@txt}{\csuse{sw@sw@txt}
1504 }}{#1}}%
1505 \fi%
1506 %
```

And print the word.

```
1506     #2%
1507 }%
1508 %
```

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

```
\if@addsw09 \newif\if@addsw%
1510 %
```

\@sw The command printed in the auxiliary files.

```
1511 \newcommand{\@sw}[3]{%
1512     \get@sw@txt{#1}%
1513     \unless\ifledRcol%
1514 %
```

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

```
1515     \csxdef{sw@sw@txt @\the\absline@num @\the\section@num}{#2}%
1516 %
```

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


```

1517 \numdef{\prev@line}{\the\absline@num-1}%
1518 \ifcsundef{sw@sw@txt @\prev@line @\the\section@num}{%
1519   \csnumgdef{sw@sw@txt @\prev@line @\the\section@num}{#2-1}%
1520   }{}%
1521 %

```

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

```

1522 \numdef{\the@sw}{#2-\csuse{sw@sw@txt @\prev@line @\the\section@num}}%
1523 %

```

And do the same thing for the right side.

```

1524 \else%
1525   \csxdef{sw@sw@txt @\the\absline@numR @\the\section@numR @R}{#2}%
1526   \numdef{\prev@line}{\the\absline@numR-1}%
1527   \ifcsundef{sw@sw@txt @\prev@line @\the\section@numR @R}{%
1528     \csnumgdef{sw@sw@txt @\prev@line @\the\section@numR @R}{#2-1}%
1529     }{}%
1530   \numdef{\the@sw}{#2-\csuse{sw@sw@txt @\prev@line @\the\section@numR @R}}%
1531 \fi%
1532 %

```

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

```

1533 \@tempcnta=\@edtext@level
1534 \@whilenum{\@tempcnta>0}\do{%
1535   \ifcsdef{sw@list@edtext@tmp@\the\@tempcnta}%
1536   {%
1537     \@addswfalse%
1538     \notbool{lemmacommand@\the\@tempcnta}%
1539     {\@addswtrue}%
1540     {\IfStrEq{#3}{inlemma}%
1541      {\@addswtrue}%
1542      {%
1543        \def\do##1{%
1544          \ifnumequal{##1}{\the\@tempcnta}%
1545          {\@addswtrue\listbreak}%
1546          }%
1547        }%
1548        \docsvlist{#3}%
1549      }%
1550    }%
1551   \if@addsw%
1552     \letcs{\@tmp}{sw@list@edtext@tmp@\the\@tempcnta}%
1553     \ifledRcol%
1554       \xright@appenditem{\the@sw}{\the\absline@numR}}\to\@tmp%
1555     \else%
1556       \xright@appenditem{\the@sw}{\the\absline@num}}\to\@tmp%
1557     \fi%
1558     \cslet{sw@list@edtext@tmp@\the\@tempcnta}{\@tmp}%

```

```

1559         \fi%
1560     }%
1561 }%
1562 \advance\@tempcnta by -1%
1563 }%
1564 }%
1565 %

```

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

```

1566 \newcommandx{\sameword@inedtext}[2][1,usedefault]{%
1567     \get@sw@txt{#2}%
1568     \unless\ifledRcol@%
1569 %

```

Just a precaution.

```

1570     \ifx\sw@list@inedtext\empty%
1571         \def\the@sw{999}%
1572         \def\this@absline{-99}%
1573     \else%
1574 %

```

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

```

1575         \glp\sw@list@inedtext\to\@tmp%
1576         \edef\the@sw{\expandafter\@firstoftwo\@tmp}%
1577         \edef\this@absline{\expandafter\@secondoftwo\@tmp}%
1578     \fi%
1579 %

```

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

```

1580     \ifcsdef{sw@\sw@txt @\this@absline @\the\section@num}{%
1581         \numdef{\prev@line}{\this@absline-1}%
1582         \numdef{\sw@atthisline}{\csuse{sw@\sw@txt @\this@absline @\the\
section@num}-\csuse{sw@\sw@txt @\prev@line @\the\section@num}}%
1583     }%
1584     {\numdef{\sw@atthisline}{0}}%
1585 %

```

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

```

1586         \ifnumgreater{\sw@atthisline}{1}%
1587             {\showwordrank{#2}{\the@sw}}%
1588             {#2}%
1589 %

```

And the same for right side.

```

1590     \else%
1591         \ifx\sw@list@inedtext\empty%

```

```

1592     \def\the@sw{999}%
1593     \def\this@absline{-99}%
1594     \else%
1595         \gl@p\sw@list@inedtext\to\@tmp%
1596         \edef\the@sw{\expandafter\@firstoftwo\@tmp}%
1597         \edef\this@absline{\expandafter\@secondoftwo\@tmp}%
1598     \fi%
1599     \ifcsdef{sw\sw@txt @\this@absline @\the\section@numR @R}{%
1600         \numdef{\prev@line}{\this@absline-1}%
1601         \numdef{\sw@atthisline}{\csuse{sw\sw@txt @\this@absline @\the\
section@numR @R}-\csuse{sw\sw@txt @\prev@line @\the\section@numR @R}}%
1602     }%
1603     {\numdef{\sw@atthisline}{0}}%
1604     \ifnumgreater{\sw@atthisline}{1}%
1605         {\showwordrank{#2}{\the@sw}}%
1606         {#2}%
1607     \fi%
1608 }%
1609 %

```

```

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

```

```

1611 \newcommand{\showwordrank}[2]{%
1612     #1\textsuperscript{#2}%
1613 }%
1614 %

```

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

```

```

1615 \newbox\raw@text
1616 \newif\ifnumberedpar@
1617 \newcount\num@lines
1618 \newbox\one@line
1619 \newcount\par@line
1620 %

```

`\pstart` starts the paragraph by clearing the `\inserts@list` list and other relevant variables, and then arranges for the subsequent text to go into the `\raw@text` box. `\AtEveryPstart` `\pstart` needs to appear at the start of every paragraph that is to be numbered; the `\numberstarttrue` `\pstart` command below may be used to insert these commands automatically. `\numberstartfalse` `\autopar` command below may be used to insert these commands automatically. `\labelstarttrue` 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. `\labelstartfalse` `\thepstart`

```

1621
1622 \newcommand{\AtEveryPstart}[1]{%
1623   \ifstrempy{#1}%
1624   {\xdef\at@every@pstart{}}%
1625   {\gdef\at@every@pstart{\noindent#1}}%
1626 }%
1627 \xdef\at@every@pstart{}%
1628
1629 \newcounter{pstart}
1630 \renewcommand{\thepstart}{\bfseries\@arabic\c@pstart}. }
1631 \newif\ifnumberpstart
1632 \numberpstartfalse
1633 \newif\iflabelpstart
1634 \labelpstartfalse
1635 \newcommand*{\pstart}[1][1]{%
1636   \normal@pars%
1637   \ifstrempy{#1}{\at@every@pstart}{\noindent#1}%
1638   \ifluatex%
1639     \edef\l@luatexttextdir@L{\the\textdir}%
1640   \fi%
1641   \@nobraektrue%
1642   \ifnumbering \else%
1643     \led@err@PstartNotNumbered%
1644     \beginnumbering%
1645   \fi%
1646   \ifnumberedpar@%
1647     \led@err@PstartInPstart%
1648     \pend%
1649   \fi%
1650   \list@clear{\inserts@list}%
1651   \global\let\next@insert=\empty%
1652   \begingroup%
1653   \global\advance \l@dnumpstartsL\@ne
1654   \global\setbox\raw@text=\vbox\bgroup%

```

```

1655 \ifautopar\else%
1656 \ifnumberpstart%
1657   \ifistanza\else%
1658     \ifsidepstartnum\else%
1659       \thepstart%
1660     \fi%
1661   \fi%
1662 \fi%
1663 \fi%
1664 \numberedpar@true%
1665 \iflabelpstart\protected@edef\@currentlabel%
1666   {\p@pstart\thepstart}
1667 \fi%
1668 \l@dzeropenalties%
1669 \ignorespaces%because not automatically ignored if an optional argument
is used (classical TeX behavior for space after commands)
1670 }
1671 %

```

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

```

1672 \newcommand*{\pend}[1][1]{\ifnumbering \else%
1673   \led@err@PendNotNumbered%
1674 \fi%
1675 \global\l@dskipversenumberfalse%
1676 \ifnumberedpar@ \else%
1677   \led@err@PendNoPstart%
1678 \fi%
1679 %

```

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.

```

1680 \l@dzeropenalties%
1681 \endgraf\global\num@lines=\prevgraf\egroup%
1682 \global\par@line=0%
1683 %

```

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`.

```

1684 \csnumdef{pstartline}{0}%
1685 \loop\ifvbox\raw@text%
1686   \csnumdef{pstartline}{\pstartline+1}%
1687   \do@line%

```

```

1688 \ifbypstart@%
1689 \ifnumequal{\pstartline}{1}{%
1690 \bgroup%
1691 \let\leavevmode\relax%
1692 \setline{1}%
1693 \egroup%
1694 \resetprevline@{}%
1695 \fi%
1696 \repeat%
1697 %

```

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

```

1698 \flush@notes%
1699 \endgroup%
1700 \ignorespaces%
1701 %

```

Increase `pstart` counter.

```

1702 \ifnumberpstart%
1703 \pstartnumtrue%
1704 \fi%
1705 \addtocounter{pstart}{1}%
1706 %

```

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

```

1707 \normal@pars%
1708 \ifstrempy{#1}{\at@every@pend}{\noindent#1}%
1709 %

```

Restore standard `nobreak` setting and `autopar` setting. Normally, `\if@nobreak` is equal to true only immediatly 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.

```

1710 \@nobreakfalse%
1711 \ifautopar%
1712 \autopar%
1713 \fi%
1714 }
1715
1716 %

```

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.

```

\AtEveryPend17
\at@every@pend18 \newcommand{\AtEveryPend}[1]{%
1719 \ifstrempy{#1}%
1720 {\xdef\at@every@pend{}}%

```

```

1721 {\gdef\at@every@pend{\noindent#1}}%
1722 }%
1723 \xdef\at@every@pend{}%
1724
1725 %

```

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

```

1726 \newcommand*\l@dzeropenalties{%
1727   \brokenpenalty \z@ \clubpenalty \z@
1728   \displaywidowpenalty \z@ \interlinepenalty \z@ \predisplaypenalty \z@
1729   \postdisplaypenalty \z@ \widowpenalty \z@}
1730
1731 %

```

\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.

```

1732 \newif\ifautopar
1733 \autoparfalse
1734 \newcommand*\autopar{%
1735   \ifledRcol
1736     \ifnumberingR \else
1737       \led@err@AutoparNotNumbered
1738     \beginnumberingR
1739     \fi
1740   \else

```

```

1741 \ifnumbering \else
1742 \led@err@AutoparNotNumbered
1743 \beginnumbering
1744 \fi
1745 \fi
1746 \autopartrue
1747 \everypar{\setbox0=\lastbox
1748 \endgraf \vskip-\parskip
1749 \pstart \noindent \kern\wd0 \ifnumberpstart\ifinstanza\else\thepstart\
fi\fi
1750 \let\par=\pend}%
1751 \ignorespaces}
1752 %

```

\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.

```

1753 \newcommand*{\normal@pars}{\ifautopar\everypar{}\fi\let\par\endgraf}
1754
1755 %

```

\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.

```

1756 \newif\ifautopar@pause
1757 %

```

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

```

1758 \newcommand*{\l@dunhbox@line}[1]{\unhbox #1}
1759 \newcommand*{\do@line}{%
1760 {\vbadness=10000
1761 \splittopskip=\z@
1762 \do@linehook
1763 \l@demtyd@ta
1764 \global\setbox\one@line=\vsplit\raw@text to\baselineskip}%
1765 \unvbox\one@line \global\setbox\one@line=\lastbox
1766 \getline@num
1767 \IfStrEq{\led@pb@setting}{before}{\led@check@pb\led@check@nopb}{}}
1768 \ifnum\@lock>\@ne
1769 \inserthangingsymboltrue
1770 \else
1771 \inserthangingsymbolfalse

```



```

1772 \fi
1773 \check@pb@in@verse
1774 \ifl@dhidenumber%
1775   \global\l@dhidenumberfalse%
1776   \f@x@l@cks%
1777 \else%
1778   \affixline@num%
1779 \fi%
1780 %

```

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

```

1781 \xifinlist{\the\l@dnumpstartsL}{\eled@sections@@}%
1782   {\print@eledsection}%
1783   {\print@line}%
1784 \IfStrEq{\led@pb@setting}{after}{\led@check@pb\led@check@nopb}{%
1785 }%
1786 %

```

VII.2.2 Process for “normal” line

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

```

1787 \def\print@line{
1788 %

```

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

```

1789   \affixpstart@num%
1790 %

```

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

```

1791   \hb@xt@ \linewidth{%
1792 %

```

User hook.

```

1793   \do@insidelinehook%
1794 %

```

Left line number

```

1795   \l@dld@ta%
1796 %

```

Restore marginal and footnotes.

```

1797   \add@inserts\affixside@note%
1798 %

```

Print left notes.

```

1799   \l@dlsn@te
1800 %

```

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

```
1801     {\ledllfill\hb@xt@ \wd\one@line{\new@line%
1802 %
```

If we use Lua[®]TeX then restore the direction.

```
1803     \ifluatex%
1804     \textdir\l@luatextextdir@L%
1805     \fi%
1806 %
```

Insert, if needed, the hanging symbol.

```
1807     \inserthangingsymbol %Space keep for backward compatibility
1808 %
```

And so, print the line.

```
1809     \l@dunhbox@line{\one@line}}%
1810 %
```

Right line number

```
1811     \ledrlfill\l@drd@ta%
1812 %
```

Print right notes.

```
1813     \l@drsn@te
1814     }}%
1815 %
```

And reinsert penalties (for page breaking)...

```
1816     \add@penalties%
1817 }
1818 %
```

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.

```
1819 \def\print@eledsection{%
1820     \add@inserts\affixside@note%
1821     \numdef{\temp@}{\l@dnumstartsL-1}%
1822     \xifinlist{\temp@}{\eled@sections@@}{\@nobreaktrue}{\@nobreakfalse}%
1823     \@eled@sectioningtrue%
1824     \csuse{eled@sectioning@the\l@dnumstartsL}%
1825     \@eled@sectioningfalse%
1826     \global\csundef{eled@sectioning@the\l@dnumstartsL}%
1827     \if@RTL%
```

```

1828 \hspace{-3\paperwidth}%
1829 {\hbox{\l@dunhbox@line{\one@line}} \new@line}%
1830 \else%
1831 \hspace{3\paperwidth}%
1832 {\new@line \hbox{\l@dunhbox@line{\one@line}}}%
1833 \fi%
1834 \vskip-\baselineskip%
1835 }
1836 %

```

VII.2.4 Hooks

`\do@linehook` Two hooks into `\do@line`. The first is called at the beginning of `\do@line`, the second is called in the line box. The second can, for example, have a `\markboth` command inside, the first can not.

```

1837 \newcommand*{\do@linehook}{}
1838 \newcommand*{\do@insidelinehook}{}
1839 %

```

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

```

1840 \newcommand*{\dolinehook}[1]{\gdef\do@linehook{#1}}%
1841 \newcommand*{\doinsidelinehook}[1]{\gdef\do@insidelinehook{#1}}%
1842
1843 %

```

VII.2.5 Sidenotes and marginal line number initialization

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

```

\l@dcsnotetext
1844 \newcommand*{\l@emptyd@ta}{%
\l@dcsnotetext@l
1845 \gdef\l@dld@ta{}%
\l@dcsnotetext@r
1846 \gdef\l@drd@ta{}%
1847 \gdef\l@dcsnotetext@l{}%
1848 \gdef\l@dcsnotetext@r{}%
1849 \gdef\l@dcsnotetext{}%
1850
1851 %

```

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

```

\l@drsn@te
1852 \newcommand{\l@dlsn@te}{%
1853 \hb@xt@ \z@{\hss\box\l@dldp@rbox\kern\ledlsnotesep}}
1854 \newcommand{\l@drsn@te}{%
1855 \hb@xt@ \z@{\kern\ledrsnotesep\box\l@drp@rbox\hss}}

```

```
1856
1857 %
```

`\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`.

```
1858 \newcommand*{\ledllfill}{\hfil}
1859 \newcommand*{\ledrlfill}{\hfil}
1860
1861 %
```

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.

```
1862 \newcommand*{\getline@num}{%
1863   \global\advance\absline@num \@ne%
1864   \do@actions
1865   \do@ballast
1866   \ifnumberline
1867     \ifsublines@
1868       \ifnum\sub@lock<\tw@
1869         \global\advance\subline@num \@ne
1870       \fi
1871     \else
1872       \ifnum\@lock<\tw@
1873         \global\advance\line@num \@ne
1874         \global\subline@num \z@
1875       \fi
1876     \fi
1877   \fi
1878 }
1879 %
```

`\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. 141).

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

```
1880 \newcount\ballast@count
1881 \newcounter{ballast}
1882 \setcounter{ballast}{0}
1883 %
```

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.

```

1884 \newcommand*{\do@ballast}{\global\ballast@count \z@
1885 \beginingroup
1886 \advance\absline@num \@ne
1887 \ifnum\next@actionline=\absline@num
1888 \ifnum\next@action>-1001\relax
1889 \global\advance\ballast@count by -\c@ballast
1890 \fi
1891 \fi
1892 \endgroup}
1893 %

```

`\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`.

```

1894 \newcommand*{\do@actions}{%
1895 \global\let\do@actions@next=\relax
1896 \ifnum\absline@num<\next@actionline\else
1897 %

```

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.

```

1898 \ifnum\next@action>-1001
1899 \global\page@num=\next@action
1900 \ifbypage@
1901 \global\line@num=\z@ \global\subline@num=\z@
1902 \resetprevline@
1903 \fi
1904 %

```

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`.)

```

1905 \else
1906 \ifnum\next@action<-4999
1907 \@l@tempcnta=-\next@action
1908 \advance\@l@tempcnta by -5001
1909 \ifsublines@
1910 \global\subline@num=\@l@tempcnta
1911 \else
1912 \global\line@num=\@l@tempcnta
1913 \fi
1914 %

```

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

```

1915     \else
1916         \@l@dttempcnta=-\next@action
1917         \advance\@l@dttempcnta by -1000
1918         \do@actions@fixedcode
1919     \fi
1920 \fi
1921 %

```

Now we get information about the next action off the list, and then set `\do@actions@next` so that we will call ourself recursively: the next action might also be for this line.

There is no warning if we find `\actionlines@list` empty, since that will always happen near the end of the section.

```

1922     \ifx\actionlines@list\empty
1923         \gdef\next@actionline{1000000}%
1924     \else
1925         \gl@p\actionlines@list\to\next@actionline
1926         \gl@p\actions@list\to\next@action
1927         \global\let\do@actions@next=\do@actions
1928     \fi
1929 \fi
1930 %

```

Make the recursive call, if necessary.

```

1931 \do@actions@next}
1932
1933 %

```

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

```

1934 \newcommand*{\do@actions@fixedcode}{%
1935     \ifcase\@l@dttempcnta
1936     \or% % 1001
1937         \global\sublines@true
1938     \or% % 1002
1939         \global\sublines@false
1940     \or% % 1003
1941         \global\@lock=\@ne
1942     \or% % 1004
1943         \ifnum\@lock=\tw@
1944             \global\@lock=\thr@@
1945         \else
1946             \global\@lock=\z@
1947         \fi
1948     \or% % 1005
1949         \global\sub@lock=\@ne
1950     \or% % 1006
1951         \ifnum\sub@lock=\tw@
1952             \global\sub@lock=\thr@@

```

```

1953 \else
1954 \global\sub@lock=\z@
1955 \fi
1956 \or% % 1007
1957 \l@dskipnumbertrue
1958 \or% % 1008
1959 \l@dskipversenumbertrue%
1960 \or% % 1009
1961 \l@dhidenumbertrue
1962 \else
1963 \led@warn@BadAction
1964 \fi}
1965
1966
1967 %

```

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.

```

1968 \newcommand*{\affixline@num}{%
1969 %

```

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).

```

1970 \ifledgroupnotesL@\else
1971 \ifnumberline
1972 \ifl@dskipnumber
1973 \global\l@dskipnumberfalse
1974 \else
1975 \ifsublines@
1976 \l@tempcntb=\subline@num

```

```

1977 \ifnum\subline@num>\c@firstsublinenum
1978 \@l@dttempcnta=\subline@num
1979 \advance\@l@dttempcnta by-\c@firstsublinenum
1980 \divide\@l@dttempcnta by\c@sublinenumincrement
1981 \multiply\@l@dttempcnta by\c@sublinenumincrement
1982 \advance\@l@dttempcnta by\c@firstsublinenum
1983 \else
1984 \@l@dttempcnta=\c@firstsublinenum
1985 \fi
1986 %

```

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.

```

1987 \ch@cksub@l@ck
1988 %

```

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

```

1989 \else
1990 \@l@dttempcntb=\line@num
1991 %

```

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

```

1992 \ifx\linenumberlist\empty
1993 \ifnum\line@num>\c@firstlinenum
1994 \@l@dttempcnta=\line@num
1995 \advance\@l@dttempcnta by-\c@firstlinenum
1996 \divide\@l@dttempcnta by\c@linenumincrement
1997 \multiply\@l@dttempcnta by\c@linenumincrement
1998 \advance\@l@dttempcnta by\c@firstlinenum
1999 \else
2000 \@l@dttempcnta=\c@firstlinenum
2001 \fi
2002 \else
2003 %

```

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

```

2004 \@l@dttempcnta=\line@num
2005 \edef\rem@inder{\linenumberlist,\number\line@num,}%
2006 \edef\sc@n@list{\def\noexpand\sc@n@list
2007 #####1,\number\@l@dttempcnta,####2|\def\noexpand\rem@inder
2008 {#####2}}}%
2009 \sc@n@list\expandafter\sc@n@list\rem@inder|}%
2010 \ifx\rem@inder\empty%
2011 \advance\@l@dttempcnta\@ne
2012 \fi
2013 %

```


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

```
2014 \check@l@ck
2015 \fi
2016 %
```

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

```
2017 \ifnum\@l@tempcnta=\@l@tempcntb
2018 \ifl@dskipversenumber\else
2019 %
```

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
2020 \if@twocolumn
2021 \if@firstcolumn
2022 \gdef\l@dld@ta{\llap{\leftlinenum}}}%
2023 \else
2024 \gdef\l@drd@ta{\rlap{\rightlinenum}}}%
2025 \fi
2026 \else
2027 \l@tempcntb=\line@margin
2028 \ifnum\@l@tempcntb>\@ne
2029 \advance\@l@tempcntb \page@num
2030 \fi
2031 \ifodd\@l@tempcntb
2032 \gdef\l@drd@ta{\rlap{\rightlinenum}}}%
2033 \else
2034 \gdef\l@dld@ta{\llap{\leftlinenum}}}%
2035 \fi
2036 \fi
2037 \fi
2038 \fi
2039 %
```

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

```
2040 \fix@l@cks
2041 \fi
2042 \fi
```

```

2043 \fi
2044 }
2045
2046 %

```

`\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.

```

2047 \newcommand*{\ch@cksub@l@ck}{%
2048 \ifcase\sub@lock
2049 \or
2050 \ifnum\sublock@disp=\@ne
2051 \@l@tempcntb=\z@ \@l@tempcnta=\@ne
2052 \fi
2053 \or
2054 \ifnum\sublock@disp=\tw@ \else
2055 \@l@tempcntb=\z@ \@l@tempcnta=\@ne
2056 \fi
2057 \or
2058 \ifnum\sublock@disp=\z@
2059 \@l@tempcntb=\z@ \@l@tempcnta=\@ne
2060 \fi
2061 \fi}
2062 %

```

Similarly for line numbers.

```

2063 \newcommand*{\ch@ck@l@ck}{%
2064 \ifcase@lock
2065 \or
2066 \ifnum\lock@disp=\@ne
2067 \@l@tempcntb=\z@ \@l@tempcnta=\@ne
2068 \fi
2069 \or
2070 \ifnum\lock@disp=\tw@ \else
2071 \@l@tempcntb=\z@ \@l@tempcnta=\@ne
2072 \fi
2073 \or
2074 \ifnum\lock@disp=\z@
2075 \@l@tempcntb=\z@ \@l@tempcnta=\@ne
2076 \fi
2077 \fi}
2078 %

```

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

```

2079 \newcommand*{\f@x@l@cks}{%
2080 \ifcase@lock
2081 \or

```

```

2082 \global\@lock=\tw@
2083 \or \or
2084 \global\@lock=\z@
2085 \fi
2086 \ifcase\sub@lock
2087 \or
2088 \global\sub@lock=\tw@
2089 \or \or
2090 \global\sub@lock=\z@
2091 \fi}
2092
2093 %

```

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.

```

\leftpstartnum94
\rightpstartnum95 \newif\ifsidepstartnum
\ifsidepstartnum96 \newcommand*{\affixpstart@num}{%
2097 \ifsidepstartnum
2098 \if@twocolumn
2099 \if@firstcolumn
2100 \gdef\l@dld@ta{\llap{\leftpstartnum}}}%
2101 \else
2102 \gdef\l@drd@ta{\rlap{\rightpstartnum}}}%
2103 \fi
2104 \else
2105 \l@dttempcntb=\line@margin
2106 \ifnum\l@dttempcntb>\@ne
2107 \advance\l@dttempcntb \page@num
2108 \fi
2109 \ifodd\l@dttempcntb
2110 \gdef\l@drd@ta{\rlap{\rightpstartnum}}}%
2111 \else
2112 \gdef\l@dld@ta{\llap{\leftpstartnum}}}%
2113 \fi
2114 \fi

```

```

2115 \fi
2116
2117 }
2118 %
2119
2120 \newif\ifpstartnum
2121 \pstartnumtrue
2122 \newcommand*{\leftpstartnum}{
2123   \ifpstartnum\thepstart
2124   \kern\linenumsep\fi
2125   \global\pstartnumfalse
2126 }
2127 \newcommand*{\rightpstartnum}{
2128   \ifpstartnum
2129   \kern\linenumsep
2130   \thepstart
2131   \fi
2132   \global\pstartnumfalse
2133 }
2134 %

```

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.

```

2135 \list@create{\inserts@list}
2136 %

```

`\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.

`\add@inserts@next`

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`.

```

2137 \newcommand*{\add@inserts}{%
2138   \global\let\add@inserts@next=\relax
2139   %

```

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

```

2140   \ifx\inserts@list\empty \else
2141   %

```

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.

```

2142   \ifx\next@insert\empty
2143     \ifx\insertlines@list\empty
2144       \global\noteschanged@true
2145       \gdef\next@insert{100000}%
2146     \else
2147       \gl@p\insertlines@list\to\next@insert
2148     \fi
2149   \fi
2150   %

```

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.

```

2151   \ifnum\next@insert=\absline@num
2152     \gl@p\inserts@list\to\@insert
2153     \@insert
2154     \global\let\@insert=\undefined
2155     \global\let\next@insert=\empty
2156     \global\let\add@inserts@next=\add@inserts
2157   \fi
2158 \fi
2159 %

```

Make the recursive call, if necessary.

```

2160 \add@inserts@next}
2161
2162 %

```

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@tempcnta` is used to calculate and accumulate the penalty; it is initially set to the value of `\ballast@count`, which has been worked out in `\do@ballast` above (VIII p. 132). Finally, the penalty is checked to see that it does not go below -10000 .

```

2163 \newcommand*{\add@penalties}{\@l@tempcnta=\ballast@count
2164   \ifnum\num@lines>\@ne
2165     \global\advance\par@line \@ne
2166     \ifnum\par@line=\@ne
2167       \advance\@l@tempcnta \clubpenalty
2168     \fi
2169     \@l@tempcntb=\par@line \advance\@l@tempcntb \@ne
2170     \ifnum\@l@tempcntb=\num@lines
2171       \advance\@l@tempcnta \widowpenalty
2172     \fi
2173     \ifnum\par@line<\num@lines
2174       \advance\@l@tempcnta \interlinepenalty
2175     \fi
2176   \fi
2177   \ifnum\@l@tempcnta=\z@
2178     \relax
2179   \else
2180     \ifnum\@l@tempcnta>-10000
2181       \penalty\@l@tempcnta
2182     \else
2183       \penalty -10000
2184     \fi
2185   \fi}
2186
2187 %

```

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.

```

2188 \newcommand*{\flush@notes}{%
2189   \@xloop
2190   \ifx\inserts@list\empty \else
2191     \glp\inserts@list\to\@insert
2192     \@insert
2193     \global\let\@insert=\undefined

```

```

2194 \repeat}
2195
2196 %

```

\@xloop \@xloop is a variant of the `PLAIN TEX` `\loop` macro, useful when it's hard to construct a positive test using the `TEX` `\if` commands—as in `\flush@notes` above. One types `\@xloop ... \if ... \else ... \repeat`, and the action following `\else` is repeated as long as the `\if` test fails. (This macro will work wherever the `PLAIN TEX` `\loop` is used, too, so we could just call it `\loop`; but it seems preferable not to change the definitions of any of the standard macros.)

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

```

2197 \def\@xloop#1\repeat{%
2198   \def\body{#1\expandafter\body\fi}%
2199   \body}
2200
2201 %

```

XII Critical footnotes

The footnote macros are adapted from those in `PLAIN TEX`, 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@lemmafонт `\select@lemmafонт` 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.

```

2202 \def\select@lemmafонт#1|#2|#3|#4|#5|#6|#7|{\select@@lemmafонт#7|}
2203 \def\select@@lemmafонт#1/#2/#3/#4|{%
2204   {\fontencoding{#1}\fontfamily{#2}\fontseries{#3}\fontshape{#4}%
2205    \selectfont}
2206
2207 %

```

XII.2 Individual note options

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

```

2208 \newcommand*{\footnoteoptions@}[3]{%
2209   \def\do##1{%
2210     \ifstrequal{#1}{L}{% In Leftside
2211       \xright@appenditem{\noexpand\setkeys[mac]{#3footnoteoption}{\
2212         unexpanded{##1}}}{\to\inserts@list%
2213       \global\advance\insert@count \@ne% Increment the left insert
2214       counter.
2215     }%
2216     {\%
2217       \xright@appenditem{\noexpand\setkeys[mac]{#3footnoteoption}{\
2218         unexpanded{##1}}}{\to\inserts@listR%
2219       \global\advance\insert@countR \@ne% Increment the right insert
2220       counter insert.
2221     }%
2222   }%
2223   \notblank{#2}{\docsvlist{#2}}}% Parsing all options
2224 }
2225 %

```

XII.3 Notes language

`\footnotelang@lua` `\footnotelang@lua` is called to remember the information about the direction of a lemma when Lua_{TeX} is used.

```

2222 \newcommandx*{\footnotelang@lua}[1][1=L,usedefault]{%
2223   \ifstrequal{#1}{L}{%
2224     \xright@appenditem{\csxdef{footnote@luatextextdir}{\the\textdir}}{\to\
2225     inserts@list%Know the dir of lemma
2226     \global\advance\insert@count \@ne%
2227     \xright@appenditem{\csxdef{footnote@luatexpardir}{\the\pardir}}{\to\
2228     inserts@list%Know the dir of lemma
2229     \global\advance\insert@count \@ne%
2230   }%
2231   {\%
2232     \xright@appenditem{\csxdef{footnote@luatextextdir}{\the\textdir}}{\to\
2233     inserts@listR%Know the dir of lemma
2234     \global\advance\insert@countR \@ne%
2235   }%
2236 }
2237 %

```


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

```

2237 \newcommand*{\footnotelang@poly}[1][1=L,usedefault]{%
2238   \ifstrequal{#1}{L}{%
2239     \if@RTL%
2240       \xright@appenditem{\csxdef{footnote@dir}{@RTLtrue}}{\to\
inserts@list%Know the language used in the lemma
2241       \global\advance\insert@count \@ne%
2242     }%
2243     \else
2244       \xright@appenditem{\csxdef{footnote@dir}{@RTLfalse}}{\to\
inserts@list%Know the language of lemma
2245       \global\advance\insert@count \@ne%
2246     }%
2247     \xright@appenditem{\csxdef{footnote@lang}{\expandonce\language}}{\to\
inserts@list%Know the language of lemma
2248     \global\advance\insert@count \@ne%
2249   }%
2250   \if@RTL
2251     \xright@appenditem{\csxdef{footnote@dir}{@RTLtrue}}{\to\
inserts@listR%Know the language of lemma
2252     \global\advance\insert@countR \@ne%
2253   }%
2254     \else
2255       \xright@appenditem{\csxdef{footnote@dir}{@RTLfalse}}{\to\
inserts@listR%Know the language of lemma
2256       \global\advance\insert@countR \@ne%
2257     }%
2258     \xright@appenditem{\csxdef{footnote@lang}{\expandonce\language}}{\to\
inserts@listR%Know the language of lemma
2259     \global\advance\insert@countR \@ne%
2260   }%
2261 }%

```

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`.

```

2262 \newcommand*{\footsplitskips}{%
2263   \interlinepenalty=\interfootnotelinepenalty
2264   \unless\ifl@dprintingpages%
2265     \floatingpenalty=\@MM%
2266   \fi%
2267   \splittopskip=\ht\strutbox \splitmaxdepth=\dp\strutbox
2268   \leftskip=\z@skip \rightskip=\z@skip}
2269
2270 %

```

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

```

2271 \let\normalfootnoterule=\footnoterule
2272 %

```

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.

```

2273 \newcommandx{\Xarrangement}[2][1,usedefault]{%
2274   \def\do##1{%
2275     \csname Xarrangement@#2\endcsname{##1}%
2276   }%

```

```

2277 \ifstrempy{#1}%
2278   {%
2279   \dolistloop{\@series}%
2280   }%
2281   {
2282   \docsvlist{#1}%
2283   }%
2284 }%
2285 %

```

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.

```

2286 \newcommand*{\Xarrangement@normal}[1]{%
2287   \csgdef{series@display#1}{normal}
2288   \expandafter\let\csname #1footstart\endcsname=\normalfootstart
2289   \expandafter\let\csname #1footnote\endcsname=\normalvfootnote
2290   \expandafter\let\csname #1footfmt\endcsname=\normalfootfmt
2291   \expandafter\let\csname #1footgroup\endcsname=\normalfootgroup
2292   \expandafter\let\csname #1footnoterule\endcsname=%
2293   \normalfootnoterule
2294   \count\csname #1footins\endcsname=1000
2295   \dimen\csname #1footins\endcsname=\csuse{Xmaxhnotes@#1}
2296   \skip\csname #1footins\endcsname=\csuse{Xbeforenotes@#1}%
2297   \advance\skip\csname #1footins\endcsname by\csuse{Xafterrule@#1}%
2298   %

```

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.

```

2299 \csxdef{default@#1footins}{1000}%Use this to confine the notes to one
side only
2300 %

```

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).

```

2301 \ifnoledgroup@else%
2302 \expandafter\let\csname mpv#1footnote\endcsname=\mpnormalvfootnote
2303 \expandafter\let\csname mp#1footgroup\endcsname=\mpnormalfootgroup
2304 \count\csname mp#1footins\endcsname=1000
2305 \dimen\csname mp#1footins\endcsname=\csuse{Xmaxhnotes@#1}
2306 \skip\csname mp#1footins\endcsname=\csuse{Xbeforenotes@#1}%
2307 \advance\skip\csname mp#1footins\endcsname by\csuse{Xafterrule@#1}%
2308 \fi
2309 }
2310
2311 %

```

`\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.

```

2312 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\normalvfootnote}[2]{%
2313 \insert\csname #1footins\endcsname\bgroup
2314 \hsize=\expandafter\dimexpr\csuse{Xwidth@#1}\relax%
2315 \noindent\csuse{Xbhooknote@#1}%
2316 \csuse{Xnotefontsize@#1}%
2317 \footplitskips
2318 \ifl@dpairing\ifl@dpadding\else%
2319 \setXnoteswidthliketwocolumns@{#1}%
2320 \fi\fi%
2321 \setXnotespositionliketwocolumns@{#1}%
2322 \spaceskip=\z@skip \xspaceskip=\z@skip
2323 \csname #1footfmt\endcsname #2{#1}\egroup}
2324 %

```

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

```

2325 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\mpnormalvfootnote}[2]{%
2326 \global\setbox\@nameuse{mp#1footins}\vbox{%
2327 \unvbox\@nameuse{mp#1footins}
2328 \noindent\csuse{Xbhooknote@#1}%
2329 \csuse{Xnotefontsize@#1}%
2330 \hsize\columnwidth
2331 \@parboxrestore
2332 \color@begingroup
2333 \csname #1footfmt\endcsname #2{#1}\color@endgroup}}

```

2334
2335 %

\normalfootfmt \normalfootfmt is a ‘normal’ macro to take the footnote line and page number information (see V.9 p. 87), 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.

```
2336
2337
2338 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\normalfootfmt}[4]{%
2339   \Xledsetnormalparstuff{#4}%
2340   \hangindent=\csuse{Xhangindent@#4}
2341   \everypar{\hangindent=\csuse{Xhangindent@#4}}%
2342   \strut{\printlinefootnote{#1}{#4}}%
2343   {\nottoggle{Xlemmadisablefontselection@#4}%
2344     {\select@lemmafонт#1|{\csuse{Xlemmafонт@#4}#2}}%
2345     {\csuse{Xlemmafонт@#4}#2}}%
2346   }%
2347   \iftoggle{nosep@}{\hskip\csuse{Xinplaceoflemmaseparator@#4}}{\ifcempty{
Xlemmaseparator@#4}%
2348     {\hskip\csuse{Xinplaceoflemmaseparator@#4}}%
2349     {\nobreak\hskip\csuse{Xbeforelemmaseparator@#4}\csuse{Xlemmaseparator@
#4}\hskip\csuse{Xafterlemmaseparator@#4}\relax%
2350   }}%
2351   #3\strut\par}
2352 %
```

\normalfootstart \normalfootstart is a standard footnote-starting macro, called in the output routine whenever there are footnotes of this series to be printed: it skips a bit and then draws a rule.

Any \footstart macro must put onto the page something that takes up space exactly equal to the \skip\Xfootins value for the associated series of notes. T_EX makes page computations based on that \skip value, and the output pages will suffer from spacing problems if what you add takes up a different amount of space.

But if the skip \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.

```
2353 \newcommand*{\normalfootstart}[1]{%
2354 %
```

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

```

2355 \ifdimequal{0pt}{\preXnotes@}{}%
2356   {%
2357     \iftoggle{preXnotes@}{%
2358       \togglefalse{preXnotes@}%
2359       \skip\csname #1footins\endcsname=%
2360       \dimexpr\csuse{preXnotes@}+\csuse{Xafterrule@#1}\relax%
2361     }%
2362   }%
2363 }%
2364 \vskip\skip\csname #1footins\endcsname%
2365 %

```

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. 194 for the general description of the problem.

```

2366 \leftskipOpt \rightskipOpt
2367 \ifl@dpairing\else%
2368   \hsize=\old@hsize%
2369   \fi%
2370   \setXnoteswidthliketwocolumns@{#1}%
2371   \setXnotespositionliketwocolumns@{#1}%
2372 %

```

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

```

2373 \print@Xfootnoterule{#1}%
2374 \noindent\leavevmode}
2375 %

```

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

```

2376 \newcommand*{\normalfootgroup}[1]{%
2377   {\csuse{Xnotefontsize@#1}\noindent\csuse{Xtxtbeforenotes@#1}}%
2378   \csuse{Xbhookgroup@#1}%
2379   \unvbox\csname #1footins\endcsname%
2380   \hsize=\old@hsize%
2381 }%
2382 %
2383 %

```

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

```

2384 \unless\ifnoledgroup@
2385 \newcommand*{\mpnormalfootgroup}[1]{%

```

```

2386 \vskip\skip\@nameuse{mp#1footins}
2387 \ifl@dpairing\ifparledgroup%
2388   \leavevmode\marks\parledgroup@{begin}%
2389   \marks\parledgroup@series{#1}%
2390   \marks\parledgroup@type{Xfootnote}%
2391 \fi\fi\normalcolor%
2392 \ifparledgroup%
2393   \ifl@dpairing%
2394   \else%
2395     \setXnoteswidthliketwocolumns@{#1}%
2396     \setXnotespositionliketwocolumns@{#1}%
2397     \print@Xfootnoterule{#1}%
2398   \fi%
2399 \else%
2400   \setXnoteswidthliketwocolumns@{#1}%
2401   \setXnotespositionliketwocolumns@{#1}%
2402   \print@Xfootnoterule{#1}%
2403 \fi%
2404 \setlength{\parindent}{0pt}
2405 {\csuse{Xnotefontsize@#1}\csuse{Xtxtbeforenotes@#1}}
2406 \csuse{Xhookgroup@#1}%
2407 \unvbox\csname mp#1footins\endcsname}}
2408 \fi
2409 %

```

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 \TeX of ordinary size may not be able to handle more than about 100 notes of this kind on a page.

\Xarrangement@paragraph The `\Xarrangement@paragraph` macro sets up everything for one series of the footnotes so that they will be paragraphed; it takes the series letter as argument. We include the setting of `\count\footins` to 1000 for the footnote series just in case user is switching to paragraphed footnotes after having columnar ones, since they change this value (see below).

The argument of `\Xarrangement@footparagraph` is the letter denoting the series of notes to be paragraphed.

```

2410 \newcommand*{\Xarrangement@paragraph}[1]{%
2411   \csgdef{series@display#1}{paragraph}
2412   \expandafter\newcount\csname #1prevpage@num\endcsname
2413   \expandafter\let\csname #1footstart\endcsname=\parafootstart
2414   \expandafter\let\csname v#1footnote\endcsname=\paravfootnote
2415   \expandafter\let\csname #1footfmt\endcsname=\parafootfmt
2416   \expandafter\let\csname #1footgroup\endcsname=\parafootgroup
2417   \count\csname #1footins\endcsname=1000

```

```

2418 \csxdef{default@#1footins}{1000}%Use this to confine the notes to one
side only
2419 \dimen\csname #1footins\endcsname=\csuse{Xmaxhnotes@#1}
2420 \skip\csname #1footins\endcsname=\csuse{Xbeforenotes@#1}%
2421 \advance\skip\csname #1footins\endcsname by\csuse{Xafterrule@#1}%
2422 \para@footsetup{#1}
2423 %

```

And the extra setup for minipages.

```

2424 \ifnoledgroup@else
2425 \expandafter\let\csname mpv#1footnote\endcsname=\mpparavfootnote
2426 \expandafter\let\csname mp#1footgroup\endcsname=\mpparafootgroup
2427 \count\csname mp#1footins\endcsname=1000
2428 \dimen\csname mp#1footins\endcsname=\csuse{Xmaxhnotes@#1}
2429 \skip\csname mp#1footins\endcsname=\csuse{Xbeforenotes@#1}%
2430 \advance\skip\csname mp#1footins\endcsname by\csuse{Xafterrule@#1}%
2431 \fi
2432 }
2433 %

```

\footfudgefiddle For paragraphed footnotes \TeX has to estimate the amount of space required. If it underestimates this then the notes may get too long and run off the bottom of the text block. `\footfudgefiddle` can be increased from its default 64 (say, to 70) to increase the estimate.

```

2434 \providecommand{\footfudgefiddle}{64}
2435 %

```

\para@footsetup `\footparagraph` calls the `\para@footsetup` macro to calculate a special fudge factor, which is the ratio of the `\baselineskip` to the `\hsize`. We assume that the proper value of `\baselineskip` for the footnotes (normally 9pt) has been set already. The argument of the macro is again the note series letter.

Peter Wilson thinks that `\columnwidth` should be used here for \LaTeX not `\hsize`. Peter Wilson have also included `\footfudgefiddle`.

```

2436 \newcommand*{\para@footsetup}[1]{\csuse{Xhookgroup@#1}\csuse{
Xnotefontsize@#1}
2437 \setXnoteswidthliketwocolumns@{#1}%
2438 \ifcsempy{Xwidth@#1}%
2439 {}%
2440 {\columnwidth=\expandafter\dimexpr\csuse{Xwidth@#1}\relax}%
2441 \dimen0=\baselineskip
2442 \multiply\dimen0 by 1024
2443 \divide \dimen0 by \columnwidth \multiply\dimen0 by \footfudgefiddle\
relax
2444 \csxdef{#1footfudgefactor}{%
2445 \expandafter\strip@pt\dimen0 }}
2446
2447 %

```


`\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.

```

2448 \newcommand*{\parafootstart}[1]{%
2449   \rightskip=0pt \leftskip=0pt%
2450   \nottoggle{Xparindent@#1}{\parindent=\z@}{}%
2451   \ifdimequal{0pt}{\preXnotes@}{}%
2452   {%
2453     \iftoggle{preXnotes@}{%
2454       \togglefalse{preXnotes@}%
2455       \skip\csname #1footins\endcsname=%
2456       \dimexpr\csuse{preXnotes@}+\csuse{Xafterrule@#1}\relax%
2457     }%
2458   }%
2459 }%
2460 \vskip\skip\csname #1footins\endcsname%
2461 \setXnoteswidthliketwocolumns@{#1}%
2462 \setXnotespositionliketwocolumns@{#1}%
2463 \print@Xfootnoterule{#1}%
2464 \let\bid@RTL@everypar\@empty%
2465 \noindent\leavevmode}
2466 %

```

`\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

²⁷Michael Downes, ‘Line Breaking in `\unhboxed` Text’, *TUGboat* **11** (1990), pp. 605–612.

²⁸See *The TeXbook*, p. 455 (editions after January 1990).

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 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. 149 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.

```

2467 \newcommand*{\paravfootnote}[2]{%
2468   \insert\csname #1footins\endcsname
2469   \bgroup
2470     \csuse{Xnotefontsize@#1}
2471     \footplitskips
2472     \setbox0=\vbox{\hsize=\maxdimen%
2473       \let\bidir@RTL@everypar@empty%
2474       \noindent\csuse{Xhooknote@#1}%
2475       \csname #1footfmt\endcsname #2{#1}}%
2476     \setbox0=\hbox{\Xunvxh{0}{#1}}%
2477     \dp0=0pt
2478     \ht0=\csname #1footfudgefactor\endcsname\wd0
2479   %

```

Here we produce the contents of the footnote from box 0, and add a penalty of 0 between boxes in this insert.

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.

```

2480 \if@RTL\noindent \leavevmode\fi\box0%
2481 \penalty0
2482 \egroup}
2483
2484 %

```

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 T_EX attempts to split foot paragraphs. After trying out such a split (see *The TeXbook*, p. 124), T_EX inserts a penalty of -10000 here, which nearly always forces the break at the end of the whole footnote paragraph (since individual notes can't be split) even when this leads to an overfull vbox. The change above results in a penalty of 0 instead which allows, but does not force, such breaks. This penalty of 0 is later removed, after page breaks have been decided, by the `\unpenalty` macro in `\makehboxofhboxes`. So it does not affect how the footnote paragraphs are typeset (the notes still have a penalty of -10 between them, which is added by `\parafootfmt`).

`\mpparavfootnote` This version is for minipages.

```

2485 \newcommand*{\mpparavfootnote}[2]{%
2486 \global\setbox\@nameuse{mp#1footins}\vbox{%
2487 \unvbox\@nameuse{mp#1footins}%
2488 \csuse{Xnotefontsize@#1}
2489 \footsplitskips
2490 \setbox0=\vbox{\hsize=\maxdimen%
2491 \let\bid@RTL@everypar\@empty%
2492 \noindent\color@begingroup%
2493 \csuse{Xhooknote@#1}%
2494 \csname #1footfmt\endcsname #2{#1}\color@endgroup}%
2495 \setbox0=\hbox{\Xunvxh{0}{#1}}%
2496 \dp0=\z@
2497 \ht0=\csname #1footfudgefactor\endcsname\wd0
2498 \box0
2499 \penalty0
2500 }}
2501
2502 %

```

`\Xunvxh` Here is (modified) Michael's definition of `\unvxh`, used above. Michael's macro also takes care to remove some unwanted penalties and glue that T_EX automatically attaches to the end of paragraphs. When T_EX finishes a paragraph, it throws away any remaining glue, and then tacks on the following items: a `\penalty` of 10000, a `\parfillskip` and a `\rightskip` (*The TeXbook*, pp. 99–100). `\unvxh` cancels these unwanted paragraph-final items using `\unskip` and `\unpenalty`.

```

2503 \newcommand*{\Xunvxh}[2]{%
2504 \setbox0=\vbox{\unvbox#1%
2505 \global\setbox1=\lastbox}%
2506 \unhbox1
2507 \unskip % remove \rightskip,

```

```

2508 \unskip          % remove \parfillskip,
2509 \unpenalty        % remove \penalty of 10000,
2510 \hskip\csuse{Xafternote@#2}} % but add the glue to go between the notes
2511
2512 %

```

\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).

```

2513 \newcommand*{\parafootfmt}[4]{%
2514   \Xinsertparafootsep{#4}%
2515   \ledsetnormalparstuff@common%
2516   \printlinefootnote{#1}{#4}%
2517   {\nottoggle{Xlemmadisablefontselection@#4}%
2518    {\select@lemmafонт#1|{\csuse{Xlemmafонт@#4}#2}}}%
2519    {\csuse{Xlemmafонт@#4}#2}}%
2520   }%
2521   \iftoggle{nosep@}{\hskip\csuse{Xinplaceoflemmaseparator@#4}}{\ifcsempy{
Xlemmaseparator@#4}%
2522     {\hskip\csuse{Xinplaceoflemmaseparator@#4}}}%
2523     {\nobreak\hskip\csuse{Xbeforelemmaseparator@#4}\csuse{Xlemmaseparator@
#4}\hskip\csuse{Xafterlemmaseparator@#4}%
2524   }}%
2525   #3\penalty-10 }
2526 %

```

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 \para@vfootnote.

The call to \Xnotefontsize@<s> is to ensure that the correct \baselineskip for the footnotes is used. The argument is the note series letter.

```

2527 \newcommand*{\parafootgroup}[1]{%
2528   \hsize=\expandafter\dimexpr\csuse{Xwidth@#1}\relax%
2529   \unvbox\csname #1footins\endcsname
2530   \ifcsstring{Xragged@#1}{L}{\RaggedLeft}{}%
2531   \ifcsstring{Xragged@#1}{R}{\RaggedRight}{}%
2532   \makehboxofhboxes
2533   \setbox0=\hbox{\csuse{Xnotefontsize@#1}\csuse{Xtxtbeforenotes@#1}}\
unhbox0 \removehboxes}%
2534   \csuse{Xbhookgroup@#1}%
2535   \csuse{Xnotefontsize@#1}%

```

```

2536 \unhbox0\par%
2537 \global\hsize=\old@hsize%
2538 }%
2539
2540 %

```

`\mpparafootgroup` The minipage version.

```

2541 \newcommand*{\mpparafootgroup}[1]{%
2542 \setXnoteswidthliketwocolumns@{#1}%
2543 \vskip\skip\@nameuse{mp#1footins}
2544 \ifl@dpairing\ifparledgroup%
2545 \leavevmode\marks\parledgroup@{begin}%
2546 \marks\parledgroup@series{#1}%
2547 \marks\parledgroup@type{Xfootnote}%
2548 \fi\fi\normalcolor
2549 \ifparledgroup%
2550 \ifl@dpairing%
2551 \else%
2552 \setXnoteswidthliketwocolumns@{#1}%
2553 \setXnotespositionliketwocolumns@{#1}%
2554 \print@Xfootnoterule{#1}%
2555 \fi%
2556 \else%
2557 \setXnoteswidthliketwocolumns@{#1}%
2558 \setXnotespositionliketwocolumns@{#1}%
2559 \print@Xfootnoterule{#1}%
2560 \fi%
2561 \unvbox\csname mp#1footins\endcsname
2562 \ifcsstring{Xragged@#1}{L}{\RaggedLeft}{}%
2563 \ifcsstring{Xragged@#1}{R}{\RaggedRight}{}%
2564 \makehboxofhboxes
2565 \setbox0=\hbox{\csuse{Xnotefontsize@#1}\csuse{Xtxtbeforenotes@#1}}\
unhbox0 \removehboxes}%
2566 \csuse{Xbhookgroup@#1}%
2567 \csuse{Xnotefontsize@#1}%
2568 \nottoggle{Xparindent@#1}{\parindent=\z@}{}%
2569 \unhbox0\par}}
2570
2571 %

```

And finally, the two macros which are required to transform the long horizontal box stored in the insert' box to a printable text.

```

\makehboxofhboxes% \newcommand*{\makehboxofhboxes}{\setbox0=\hbox{}}%
\removehboxes% \loop
2574 \unpenalty
2575 \setbox2=\lastbox
2576 \ifhbox2

```

```

2577 \setbox0=\hbox{\box2\unhbox0}%
2578 \repeat}
2579
2580 \newcommand*\removehboxes{\setbox0=\lastbox
2581 \ifhbox0{\removehboxes}\unhbox0 \fi}
2582
2583 %

```

Insertion of the footnotes separator The command `\Xinsertparafootsep{<series>}` must be called at the beginning of `\parafootftm`.

```

\prevpage@num\newcommand{\Xinsertparafootsep}[1]{%
\Xinsertparafootsep%
2586 \ifnumequal{\csuse{#1prevpage@num}}{\page@num}%
2587 {\ifcscdef{prevline#1}% Be sur \prevline#1 exists.
2588 {\ifnumequal{\csuse{prevline#1}}{\line@num}%
2589 {\ifcseempty{Xsymlinenum@#1}{\csuse{Xparafootsep@#1}}{}}%
2590 {\csuse{Xparafootsep@#1}}}%
2591 }%
2592 {\csuse{Xparafootsep@#1}}}%
2593 }%
2594 {\global\csname #1prevpage@num\endcsname=\page@num}%
2595 }
2596 %

```

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 `\rigidbalance` macro is taken from *The TeXbook*, p.397, with a slight change to the syntax of the arguments so that they do not depend on white space. Note also the extra unboxing in `\splitoff`, which allows the new `\vbox` to have its natural height as it goes into the alignment.

The \LaTeX `\line` macro has no relationship to the TeX `\line`. The \LaTeX equivalent is `\@@line`.

We do not call directly `\rigidbalance`, but we call `\Xrigidbalance` for critical notes and `\rigidbalanceX` for familiar notes. Both of them call `\rigidbalance`.

```

2597 \newcount\@k \newdimen\@h
2598 \newcommand*\Xrigidbalance[3]{%
2599 \hsize=\expandafter\dimexpr\csuse{Xwidth@}\currentseries\relax%
2600 \rigidbalance{#1}{#2}{#3}%
2601 }%
2602
2603 \newcommand*\rigidbalanceX[3]{%

```

```

2604 \hsize=\expandafter\dimexpr\csuse{widthX@\@currentseries}\relax%
2605 \rigidbalance{#1}{#2}{#3}%
2606 }%
2607
2608 \newcommand*{\rigidbalance}[3]{%
2609 \setbox0=\box#1 \@k=#2 \@h=#3%
2610 @@line{\splittopskip=\@h \vbadness=\@M \hfilneg
2611 \valign{##\vfil\cr\dosplits}}}%
2612
2613 \newcommand*{\dosplits}{\ifnum\@k>0 \noalign{\hfil}\splitoff
2614 \global\advance\@k-1\cr\dosplits}\fi}
2615
2616 \newcommand*{\splitoff}{\dimen0=\ht0
2617 \divide\dimen0 by\@k \advance\dimen0 by\@h
2618 \setbox2 \vsplit0 to \dimen0
2619 \unvbox2 }
2620
2621 %

```

Three columns

```

\Xarrangement@threecol 2622 \newcommand*{\Xarrangement@threecol}[1]{%
2623 \csgdef{series@display#1}{threecol}
2624 \expandafter\let\csname v#1footnote\endcsname=\threecolvfootnote
2625 \expandafter\let\csname #1footfmt\endcsname=\threecolfootfmt
2626 \expandafter\let\csname #1footgroup\endcsname=\threecolfootgroup
2627 \dimen\csname #1footins\endcsname=\csuse{Xmaxhnotes@#1}%
2628 \skip\csname #1footins\endcsname=\csuse{Xbeforenotes@#1}%
2629 \advance\skip\csname #1footins\endcsname by\csuse{Xafterrule@#1}%
2630 \threecolfootsetup{#1}
2631 %

```

The additional setup for minipages.

```

2632 \ifnoledgroup\else
2633 \expandafter\let\csname mpv#1footnote\endcsname=\mpnormalvfootnote
2634 \expandafter\let\csname mp#1footgroup\endcsname=\mpthreecolfootgroup
2635 \skip\csname mp#1footins\endcsname=\csuse{Xbeforenotes@#1}%
2636 \advance\skip\csname mp#1footins\endcsname by\csuse{Xafterrule@#1}%
2637 \mpthreecolfootsetup{#1}
2638 \fi
2639 }
2640
2641 %

```

The `\footstart` and `\footnoterule` macros for these notes assume the normal values (XII.6.2 p. 149 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.

```

2642 \newcommand*{\threecolfootsetup}[1]{%
2643   \count\csname #1footins\endcsname 333
2644   \csxdef{default@#1footins}{333}%Use this to confine the notes to one
side only
2645   \multiply\dimen\csname #1footins\endcsname \thr@@}
2646   %

```

`\mpthreecolfootsetup` The setup for minipages.

```

2647 \newcommand*{\mpthreecolfootsetup}[1]{%
2648   \count\csname mp#1footins\endcsname 333
2649   \multiply\dimen\csname mp#1footins\endcsname \thr@@}
2650   %
2651   %

```

`\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).

```

2652 \notbool{parapparatus@}{\newcommand*}{\newcommand*}{\threecolvfootnote}[2]{%
2653   \insert\csname #1footins\endcsname\bgroup%
2654   \hsize=\expandafter\dimexpr\csuse{Xwidth@#1}\relax%
2655   \noindent\csuse{Xbhooknote@#1}%
2656   \csuse{Xnotefontsize@#1}%
2657   \footplitskips%
2658   \csname #1footfmt\endcsname #2{#1}\egroup}
2659   %

```

`\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.


```

2660 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\threecolfootfmt}[4]{%
2661   \normal@pars%
2662   \hsize \csuse{Xhsizethreecol@#4}%
2663   \nottoggle{Xparindent@#4}{\parindent=\z@}{}%
2664   \tolerance=5000%
2665   \hangindent=\csuse{Xhangindent@#4}%
2666   \par%
2667   \everypar{\hangindent=\csuse{Xhangindent@#4}}%
2668   \@tempdima=\parindent%
2669   \csuse{Xcolalign@#4}%
2670   \parindent=\@tempdima%
2671   \strut{\printlinefootnote{#1}{#4}}%
2672   {\nottoggle{Xlemmadisablefontselection@#4}%
2673     {\select@lemmafont#1|{\csuse{Xlemmafont@#4}#2}}%
2674     {\csuse{Xlemmafont@#4}#2}}%
2675   }%
2676   \iftoggle{nosep@}{\hskip\csuse{Xinplaceoflemmaseparator@#4}}{\ifcsempy{
Xlemmaseparator@#4}%
2677     {\hskip\csuse{Xinplaceoflemmaseparator@#4}}%
2678     {\nobreak\hskip\csuse{Xbeforelemmaseparator@#4}\csuse{Xlemmaseparator@
#4}\hskip\csuse{Xafterlemmaseparator@#4}%
2679   }}%
2680   #3\strut\par\allowbreak
2681 }%

```

\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.

```

2682 \newcommand*{\threecolfootgroup}[1]{\csuse{Xnotefontsize@#1}%
2683   \noindent\csuse{Xtxtbeforenotes@#1}}%
2684   \csuse{Xhookgroup@#1}\par%
2685   \splittopskip=\ht\strutbox
2686   \expandafter
2687   \Xrigidbalance\csname #1footins\endcsname \thr@@ \splittopskip}
2688 %

```

\mpthreecolfootgroup The setup for minipages.

```

2689 \newcommand*{\mpthreecolfootgroup}[1]{%
2690   \vskip\skip\@nameuse{mp#1footins}
2691   \ifl@dpairing\ifparledgroup%
2692     \leavevmode\marks\parledgroup@{begin}%

```

```

2693 \marks\parledgroup@series{#1}%
2694 \marks\parledgroup@type{Xfootnote}%
2695 \fi\fi\normalcolor
2696 \ifparledgroup%
2697 \ifl@dpairing%
2698 \else%
2699 \setXnoteswidthliketwocolumns@{#1}%
2700 \setXnotespositionliketwocolumns@{#1}%
2701 \print@Xfootnoterule{#1}%
2702 \fi%
2703 \else%
2704 \setXnoteswidthliketwocolumns@{#1}%
2705 \setXnotespositionliketwocolumns@{#1}%
2706 \print@Xfootnoterule{#1}%
2707 \fi%
2708 {\cuse{Xnotefontsize@#1}\noindent\cuse{Xtxtbeforenotes@#1}}%
2709 \cuse{Xbhookgroup@#1}\par%
2710 \splittopskip=\ht\strutbox
2711 \expandafter
2712 \Xrigidbalance\cname mp#1footins\endcsname \thr@@ \splittopskip}}
2713
2714 %

```

Two columns

```

\Xarrangement@twocol15 \newcommand*{\Xarrangement@twocol}[1]{%
2716 \csgdef{series@display#1}{twocol}
2717 \expandafter\let\cname v#1footnote\endcsname=\twocolvfootnote
2718 \expandafter\let\cname #1footfmt\endcsname=\twocolfootfmt
2719 \expandafter\let\cname #1footgroup\endcsname=\twocolfootgroup
2720 \dimen\cname #1footins\endcsname=\cuse{Xmaxhnotes@#1}%
2721 \skip\cname #1footins\endcsname=\cuse{Xbeforenotes@#1}%
2722 \advance\skip\cname #1footins\endcsname by\cuse{Xafterrule@#1}%
2723 \twocolfootsetup{#1}
2724 %

```

The additional setup for minipages.

```

2725 \ifnoledgroup@else
2726 \expandafter\let\cname mpv#1footnote\endcsname=\mpnormalvfootnote
2727 \expandafter\let\cname mp#1footgroup\endcsname=\mptwocolfootgroup
2728 \skip\cname mp#1footins\endcsname=\cuse{Xbeforenotes@#1}%
2729 \advance\skip\cname mp#1footins\endcsname by\cuse{Xafterrule@#1}%
2730 \mptwocolfootsetup{#1}
2731 \fi
2732 }
2733
2734 %

```

`\twocolfootsetup` Here is a series of macros which are very similar to their three-column counterparts. In this case, each note is assumed to contribute only a half a line of text. And the notes are set in columns giving a gap between them of one tenth of the `\hsize`.

`\twocolvfootnote`

`\twocolfootfmt`

`\twocolfootgroup`

```

2735 \newcommand*{\twocolfootsetup}[1]{%
2736   \count\csname #1footins\endcsname 500
2737   \csxdef{default@#1footins}{500}%
2738   \multiply\dimen\csname #1footins\endcsname \tw@}
2739   %

2740 \notbool{parapparatus@}{\newcommand*{\newcommand}{\twocolvfootnote}[2]{%
2741   \insert\csname #1footins\endcsname\bgroup%
2742   \hsize=\expandafter\dimexpr\csuse{Xwidth@#1}\relax%
2743   \noindent\csuse{Xbhooknote@#1}%
2744   \csuse{Xnotefontsize@#1}%
2745   \footssplitsskip%
2746   \csname #1footfmt\endcsname #2{#1}\egroup}
2747   %

2748 \notbool{parapparatus@}{\newcommand*{\newcommand}{\twocolfootfmt}[4]{% 4th
   arg is optional, for backward compatibility
2749   \normal@pars%
2750   \hsize \csuse{Xhsizetwocol@#4}%
2751   \nottoggle{Xparindent@#4}{\parindent=\z@}{}%
2752   \tolerance=5000%
2753   \hangindent=\csuse{Xhangindent@#4}%
2754   \par%
2755   \everypar{\hangindent=\csuse{Xhangindent@#4}}%
2756   \@tempdima=\parindent%
2757   \csuse{Xcolalign@#4}%
2758   \parindent=\@tempdima%
2759   \strut{\printlinefootnote{#1}{#4}}%
2760   {\nottoggle{Xlemmadisablefontselection@#4}%
2761     {\select@lemmafont#1|{\csuse{Xlemmafont@#4}#2}}%
2762     {\csuse{Xlemmafont@#4}#2}}%
2763   }%
2764   \iftoggle{nosep@}{\hskip\csuse{Xinplaceoflemmaseparator@#4}}{\ifcsempy{
Xlemmaseparator@#4}%
2765     {\hskip\csuse{Xinplaceoflemmaseparator@#4}}%
2766     {\nobreak\hskip\csuse{Xbeforelemmaseparator@#4}\csuse{Xlemmaseparator@
#4}\hskip\csuse{Xafterlemmaseparator@#4}%
2767     }}%
2768   #3\strut\par\allowbreak}
2769   %

2770 \newcommand*{\twocolfootgroup}[1]{\csuse{Xnotefontsize@#1}
2771   \noindent\csuse{Xtxtbeforenotes@#1}}%
2772   \csuse{Xbhookgroup@#1}\par%
2773   \splittopskip=\ht\strutbox
2774   \expandafter

```

```

2775 \Xrigidbalance\csname #1footins\endcsname \tw@ \splittopskip}
2776
2777 %

```

`\mptwocolfootsetup` The versions for minipages.

`\mptwocolfootgroup`

```

2778 \newcommand*\mptwocolfootsetup}[1]{%
2779 \count\csname mp#1footins\endcsname 500
2780 \multiply\dimen\csname mp#1footins\endcsname \tw@}
2781 %

2782 \newcommand*\mptwocolfootgroup}[1]{%
2783 \vskip\skip\@nameuse{mp#1footins}
2784 \ifl@dpairing\ifparledgroup%
2785 \leavevmode\marks\parledgroup@{begin}%
2786 \marks\parledgroup@series{#1}%
2787 \marks\parledgroup@type{Xfootnote}%
2788 \fi\fi\normalcolor
2789 \ifparledgroup%
2790 \ifl@dpairing%
2791 \else%
2792 \setXnoteswidthliketwocolumns@{#1}%
2793 \setXnotespositionliketwocolumns@{#1}%
2794 \print@Xfootnoterule{#1}%
2795 \fi%
2796 \else%
2797 \setXnoteswidthliketwocolumns@{#1}%
2798 \setXnotespositionliketwocolumns@{#1}%
2799 \print@Xfootnoterule{#1}%
2800 \fi%
2801 {\csuse{Xnotefontsize@#1}\noindent\csuse{Xtxtbeforenotes@#1}}%
2802 \csuse{Xhookgroup@#1}\par%
2803 \splittopskip=\ht\strutbox
2804 \expandafter
2805 \Xrigidbalance\csname mp#1footins\endcsname \tw@ \splittopskip}}
2806
2807 %

```

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.

`\fullstop`

`\rbracket`

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.

```

2808 \def\endashchar{\textnormal{--}}
2809
2810 \newcommand*{\fullstop}{\textnormal{.}}
2811 \def\Xsublinesep@side{\fullstop}
2812
2813 \newcommand*{\rbracket}{\textnormal{%
2814   \csuse{text\csuse{footnote@lang}}{%
2815     \ifluatex%
2816       \ifdefstring{\footnote@luatextextdir}{TRT}{\thinspace[]{\thinspace
2817     }}%
2818     \else%
2819     \thinspace}%
2820     \fi}%
2821   }}
2822
2823 %

```

XII.7.2 Pstart number in footnote

`\printpstart` The `\printpstart` macro prints the pstart number for a note.

```

2824 \newcommand{\printpstart}[0]{%
2825   \ifboolexpr{bool{l@dpairing} or bool{l@dprintingpages} or bool{
2826     l@dprintingcolumns}}{%
2827     \ifledRcol%
2828       \thepstartR%
2829     \else%
2830       \thepstartL%
2831     \fi%
2832   }}%
2833   \thepstart%
2834 }%
2835 %

```

XII.7.3 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`.

```

2836 \newcommand{\printlinefootnote}[2]{%
2837   \l@dp@rsefootspec#1|{%
2838     \iftoggle{Xnumberonlyfirstintwolines@#2}{%
2839       \edef\lineinfo@{\l@dparsedstartline - \l@dparsedstartsub - \
2840         \l@dparsedendline - \l@dparsedendsub}%
2841       }%
2842       {%
2843         \edef\lineinfo@{\l@dparsedstartline - \l@dparsedstartsub}%
2844         }%
2845         \iftoggle{nonum@}{%Try if the line number must printed for this specific
2846         not (by default, yes)
2847         \hspace{\csuse{Xinplaceofnumber@#2}}}%
2848         }%
2849         {%
2850           \iftoggle{Xnonumber@#2}%Try if the line number must printed (by
2851           default, yes)
2852           {%
2853             \hspace{\csuse{Xinplaceofnumber@#2}}}%
2854             }%
2855             {\iftoggle{Xnumberonlyfirstinline@#2}% If for this series the
2856             line number must be printed only in the first time.
2857             {%
2858               \ifcsdef{prevline#2}%
2859               {%Be sure the \prevline exists.
2860               \ifcsequal{prevline#2}{\lineinfo@}%Try it
2861               {%
2862                 \ifcseempty{Xsymlinenum@#2}%
2863                 {%
2864                   \hspace{\csuse{Xinplaceofnumber@#2}}}%
2865                   }%
2866                   {\printsymlinefootnotearea{#2}}}%
2867                   }%
2868                   \printlinefootnotearea{#1}{#2}%
2869                   }%
2870                   }%
2871                   \printlinefootnotearea{#1}{#2}%
2872                   }%
2873                   }%
2874                   {%

```

```

2875         \printlinefootnotearea{#1}{#2}%
2876     }%
2877     \csxdef{prevline#2}{\lineinfo@}%
2878 }%
2879 }%
2880 }%
2881 }%
2882 }
2883 %

```

\printsymlinefootnotearea This macro prints the space before the line symbol, changes the font, when prints the line symbol and the space after it.

```

2884 \newcommand{\printsymlinefootnotearea}[1]{%
2885     \hspace{\csuse{Xbeforesymlinenum@#1}}%
2886     \csuse{Xnotenumfont@#1}%
2887     \ifdimequal{\csuse{Xboxsymlinenum@#1}}{\z@}%
2888         {\csuse{Xsymlinenum@#1}}%
2889         {\hbox to \csuse{Xboxsymlinenum@#1}%
2890             {\csuse{Xsymlinenum@#1}\hfill}}%
2891     }%
2892     \hspace{\csuse{Xaftersymlinenum@#1}}%
2893 }%
2894 %

```

\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.)

```

2895 \newcommand{\printlinefootnotearea}[2]{%
2896     \printXbeforenumber{#2}%
2897     \csuse{Xnotenumfont@#2}%
2898     \boxfootnotenumbers{#1}{#2}%
2899     \printXafternumber{#2}%
2900 }%
2901 %

```

\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.

```

2902 \newcommand{\boxfootnotenumbers}[2]{%
2903     \ifdimequal{\csuse{Xboxlinenum@#2}}{0pt}{%
2904         \printlinefootnotenumbers{#1}{#2}%
2905     }%
2906     {%
2907         \hbox to \csuse{Xboxlinenum@#2}%
2908         {%
2909             \IfSubStr{RC}{\csuse{Xboxlinenumalign@#2}}{\hfill}{}%

```

```

2910 \printlinefootnotenumbers{#1}{#2}%
2911 \IfSubStr{LC}{\csuse{Xboxlinenumalign@#2}}{\hfill}{}%
2912 }%
2913 }%
2914 }%
2915 %

```

\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.

```

2916 \newcommand{\printlinefootnotenumbers}[2]{%
2917 \xdef\@currentseries{#2}%
2918 \ifboolexpr{%
2919 (togl{Xpstart@#2} and bool{numberpstart})%
2920 or togl{Xpstarteverytime@#2}}%
2921 {\printpstart}{}%
2922 \iftoggle{Xstanza@#2}{%
2923 \ifnumberstanza%
2924 \printstanza%
2925 \csuse{Xstanzaseparator@#2}%
2926 \fi%
2927 }{}%
2928 \iftoggle{Xonlypstart@#2}{%}{%
2929 \csuse{Xtxtbeforenumber@#2}%
2930 \printlines#1||\ifledRcol@{\@Rlineflag\fi}%
2931 }%
2932 %

```

\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.)

```

2933 \newcommand{\printXbeforenumber}[1]{%
2934 \hspace{\csuse{Xbeforenumber@#1}}%
2935 }%
2936 %

```

\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

```

2937 \newcommand{\printXafternumber}[1]{%
2938 \iftoggle{Xnonbreakableafternumber@#1}{\nobreak}{}%
2939 \hspace{\csuse{Xafternumber@#1}}%
2940 }%
2941 %

```

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. 87: 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@pnum42 \newif\ifl@d@pnum
\ifl@d@ssub43 \newif\ifl@d@ssub
\ifl@d@elin44 \newif\ifl@d@elin
\ifl@d@esl45 \newif\ifl@d@esl
\ifl@d@dash46 \newif\ifl@d@dash
\ifl@d@Xtwolines47 \newif\ifl@d@Xtwolines%
\ifl@d@Xmorethantwolines48 \newif\ifl@d@Xmorethantwolines%
49 %

\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@dparsedendpage50 \def\l@dp@rsefootsspec#1|#2|#3|#4|#5|#6|#7|{%
\l@dparsedendline51 \gdef\l@dparsedstartpage{#1}%
\l@dparsedendsub52 \gdef\l@dparsedstartline{#2}%
53 \gdef\l@dparsedstartsub{#3}%
54 \gdef\l@dparsedendpage{#4}%
55 \gdef\l@dparsedendline{#5}%
56 \gdef\l@dparsedendsub{#6}%
57 }
58 %

```

Initialise the several number value macros.

```

59 \def\l@dparsedstartpage{0}%
60 \def\l@dparsedstartline{0}%
61 \def\l@dparsedstartsub{0}%
62 \def\l@dparsedendpage{0}%

```

```

2963 \def\l@dparsedendline{0}%
2964 \def\l@dparsedendsub{0}%
2965
2966 %

```

\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`.

```

2967 \newcommand*\setprintlines}[6]{%
2968   \l@dpnumfalse \l@d@dashfalse
2969   %

```

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

```

2970   \ifbypage@
2971     \ifnum#4=#1 \else
2972       \l@dpnumtrue
2973       \l@d@dashtrue
2974     \fi
2975   \fi
2976 %

```

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.

```

2977   \ifl@dpnum \l@d@elintrue \else \l@d@elinfalse \fi
2978   \ifnum#2=#5 \else
2979     \l@d@elintrue
2980     \l@d@dashtrue
2981   \fi
2982 %

```

We print the starting sub-line if it is nonzero.

```

2983   \l@d@ssubfalse
2984   \ifnum#3=0 \else
2985     \l@d@ssubtrue
2986   \fi
2987 %

```

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.

```

2988   \l@d@eslfalse
2989   \ifnum#6=0 \else
2990     \ifnum#6=#3
2991       \ifl@d@elin \l@d@esltrue \else \l@d@eslfalse \fi
2992     \else
2993       \l@d@esltrue
2994       \l@d@dashtrue
2995     \fi
2996   \fi%
2997 %

```

However, if the `\Xtwolines` is set for the current series, we do not print the last line number.

```

2998 \ifl@d@dash%
2999 \ifboolexpr{togl{fulllines@} or test{\ifcempty{Xtwolines@
@currentseries}}}%
3000 {}%
3001 {%
3002 \setistwofollowinglines{#1}{#2}{#4}{#5}%
3003 \ifboolexpr{%
3004 (%
3005 togl {Xtwolinesbutnotmore@ \@currentseries}%
3006 and not%
3007 (%
3008 bool {istwofollowinglines@}%
3009 )%
3010 )%
3011 or%
3012 (%
3013 (not test{\ifnumequal{#1}{#4}})%
3014 and togl{Xtwolinesonlyinsamepage@ \@currentseries}%
3015 )%
3016 }%
3017 {}%
3018 {%
3019 \l@d@dashfalse%
3020 \l@d@Xtwolinestrue%
3021 \l@d@elinfalse%
3022 \l@d@eslfalse%
3023 \ifcempty{Xmorethantwolines@ \@currentseries}%
3024 {}%
3025 {\ifistwofollowinglines@ \else%
3026 \l@d@Xmorethantwolinestrue%
3027 \fi%
3028 }%
3029 }%
3030 }%
3031 \fi%
3032 %

```

End of `\setprintlines`.

```

3033 }%
3034 %

```

`\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.

```

3035 \newif\ifistwofollowinglines@%
3036 \newcommand{\setistwofollowinglines}[4]{%
3037   \ifcsdef{lastlinenumberon@#1}%
3038     {\numdef{\tmp}{\csuse{lastlinenumberon@#1}}}%
3039     {\numdef{\tmp}{0}}%
3040   \istwofollowinglines@false%
3041   \ifnumequal{#4-#2}{1}%
3042     {\istwofollowinglines@true}%
3043     {\ifbypage@%
3044       \ifnumequal{#3-#1}{1}%
3045       {%
3046         \ifnumequal{#2}{\tmp}%
3047         {\ifnumequal{#4}{1}{\istwofollowinglines@true}{}}%
3048         {}%
3049       }%
3050     }%
3051   \fi%
3052 }%
3053 }%
3054 %

```

`\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

```

3055 \def\printlines#1|#2|#3|#4|#5|#6|#7|#8|{%
3056   \begingroup%
3057 %

```

If we use Lua_T_EX, ensure we use good text's direction.

```

3058 \ifluatex%
3059   \edef\@tmp{\the\textdir}%
3060   \ifdefstring{\@tmp}{TLT}{\textdir TLT}%Test in order to prevent
spurious space (bug #397)

```

```

3061 \fi%
3062 %

```

Decide which part of line number components we will print.

```

3063 \setprintlines{#1}{#2}{#3}{#4}{#5}{#6}%
3064 %

```

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.

```

3065 \ifdimequal{\csuse{Xboxstartlinenum@\@currentseries}}{0pt}%
3066 {\bgroup}%
3067 {\leavevmode\hbox to \csuse{Xboxstartlinenum@\@currentseries}\bgroup\
hf ill}%
3068 \ifl@d@pnum%
3069 \wrap@edcrossref{\@this@crossref@start}{#1}%
3070 \csuse{Xsublinesep@\@currentseries}%
3071 \fi%
3072 \wrap@edcrossref{\@this@crossref@start}{%
3073 \linenumrep{#2}%
3074 \iftoggle{Xlineflag@\@currentseries}{#8}{}}%
3075 }%
3076 \ifl@d@ssub%
3077 \csuse{Xsublinesep@\@currentseries}%
3078 \wrap@edcrossref{\@this@crossref@start}{\sublinenumrep{#3}}%
3079 \fi
3080 \egroup%
3081 %

```

Then print the dash + end line number, or the range symbol.

```

3082 \ifdimequal{\csuse{Xboxendlinenum@\@currentseries}}{0pt}%
3083 {\bgroup}%
3084 {\hbox to \csuse{Xboxendlinenum@\@currentseries}\bgroup}%
3085 \ifl@d@Xtwolines%
3086 \ifl@d@Xmorethantwolines%
3087 \csuse{Xmorethantwolines@\@currentseries}%
3088 \else%
3089 \csuse{Xtwolines@\@currentseries}%
3090 \fi%
3091 \else%
3092 \ifl@d@dash%
3093 \ifdefined\linerangesep%
3094 \linerangesep%
3095 \else%
3096 \csuse{Xlinerangeseparator@\@currentseries}%
3097 \fi%
3098 \fi%
3099 \ifl@d@pnum%

```

```

3100     \wrap@edcrossref{\@this@crossref@end}{#4}%
3101     \csuse{Xsublinesep@}\@currentseries}%
3102     \fi%
3103     \ifl@d@elin%
3104         \wrap@edcrossref{\@this@crossref@end}{%
3105             \linenumrep{#5}%
3106             \iftoggle{Xlineflag@}\@currentseries}{#8}{}%
3107         }%
3108     \fi%
3109     \ifl@d@esl%
3110         \ifl@d@elin%
3111             \csuse{Xsublinesep@}\@currentseries}%
3112         \fi%
3113         \wrap@edcrossref{\@this@crossref@end}{\sublinenumrep{#6}}%
3114     \fi%
3115 \fi%
3116 \ifdimequal{\csuse{Xboxendlinenum@}\@currentseries}}{0pt}%
3117 {}%
3118 {\hfill}%Prevent underfull hbox
3119 \egroup%
3120 \endgroup%
3121 }%
3122 %

```

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`

```

3123 \providecommand*\multiplefootnotemarker{3sp}
3124 \providecommand*\multfootsep{\textsuperscript{\normalfont,}}
3125
3126 %

```

`\m@mmf@prepare` A pair of self-cancelling kerns. This may have been defined in the `memoir` class.

```

3127 \providecommand*\m@mmf@prepare}{%
3128 \kern-\multiplefootnotemarker
3129 \kern\multiplefootnotemarker\relax}
3130 %

```

\m@mmf@check This may have been defined in the memoir class. If it recognises the last kern as \multiplefootnotemarker it typesets \multfootsep.

```

3131 \providecommand*\m@mmf@check}{%
3132 \ifdim\lastkern=\multiplefootnotemarker\relax
3133 \edef\x@sf{\the\spacefactor}%
3134 \unkern
3135 \multfootsep
3136 \spacefactor\x@sf\relax
3137 \fi}
3138 %
3139 %

```

We have to modify \@footnotetext and \@footnotemark. However, if memoir is used the modifications have already been made.

```

3140 \@ifclassloaded{memoir}{}{%
3141 %

```

\@footnotetext Add \m@mmf@prepare at the end of \@footnotetext.

```

3142 \apptocmd{\@footnotetext}{\m@mmf@prepare}{}{}
3143 %

```

\@footnotemark Modify \@footnotemark to cater for adjacent \footnotes.

```

3144
3145 \patchcmd{\@footnotemark}
3146 {\nobreak}
3147 {\m@mmf@check
3148 \nobreak
3149 }
3150 {}{}
3151 \patchcmd{\@footnotemark}
3152 {\@makefnmark}
3153 {\@makefnmark
3154 \m@mmf@prepare
3155 }
3156 {}{}
3157 %

```

Finished the modifications for the non-memoir case.

```

3158 }
3159
3160 %

```

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.

```

3161 \pretocmd{\@footnotetext}{%
3162   \ifnumberedpar@
3163     \edtext{}{\l@dbfnote{#1}}%
3164   \else
3165     {}{}
3166 \apptocmd{\@footnotetext}{\fi}{}{}%
3167 %

```

`\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`

```

3168 \patchcmd%
3169 {\footnote}%
3170 {\stepcounter\@mpfn}%
3171 {%
3172   \ifl@dpairing%
3173     \global\advance\footnote@reading by \@ne%
3174     \get@thisfootnote%
3175     \get@fnmark{\thisfootnote}%
3176     \ifcsdef{footnotereading\the\footnote@reading=typeset}%
3177       {\setcounter{\@mpfn}{\csuse{footnotereading\the\footnote@reading=
3178         typeset}}}%
3179       {\setcounter{\@mpfn}{\footnote@reading}}%
3180     \else%
3181       \stepcounter\@mpfn%
3182     \fi%
3183   }%
3184   {}
3185   {}
3186
3187 \newcommand{\get@thisfootnote}{%
3188   \ifl@dpairing
3189     \protected@xdef\thisfootnote{\the\footnote@reading}%
3190   \else%
3191     \protected@xdef\thisfootnote{\thefootnote}%
3192   \fi%
3193 }%
3194
3195 \newcommand{\l@dbfnote}[1]{%
3196   \get@thisfootnote%
3197   \gdef\tag{#1\relax}%

```



```

3198 \ifledRcol%
3199 \xright@appenditem{%
3200 \ifdefined\Hy@footnote@currentHref%
3201 \noexpand\def\noexpand\Hy@footnote@currentHref{\
Hy@footnote@currentHref}%
3202 \fi%
3203 \noexpand\vl@dbfnote{\expandonce\@tag}{\thisfootnote}%
3204 }%
3205 \to\inserts@listR
3206 \global\advance\insert@countR \@ne%
3207 \else%
3208 \xright@appenditem{%
3209 \ifdefined\Hy@footnote@currentHref%
3210 \noexpand\def\noexpand\Hy@footnote@currentHref{\
Hy@footnote@currentHref}%
3211 \fi%
3212 \noexpand\vl@dbfnote{\expandonce\@tag}{\thisfootnote}%
3213 }%
3214 \to\inserts@list
3215 \global\advance\insert@count \@ne%
3216 \fi
3217 \ignorespaces%
3218 }%
3219
3220 \newcommand{\get@fnmark}[1]{%
3221 \ifboolexpr{bool{l@dpairing} or bool{l@dprintingpages} or bool{
l@dprintingcolumns}}{%
3222 {%
3223 \stepcounter{footnote@typeset}%
3224 \setcounter{footnote}{\c@footnote@typeset}%
3225 \immediate\write\@mainaux{%
3226 \global\csdef{footnotereading#1=typeset}{\the\c@footnote@typeset}
%
3227 }%
3228 \def\@thefnmark{\thefootnote}%
3229 }%
3230 {%
3231 \@namedef{\@thefnmark}{#1}%
3232 }%
3233 }%
3234
3235 \newcommand{\vl@dbfnote}[2]{%
3236 \get@fnmark{#2}%
3237 \@footnotetext{#1}%
3238 }%
3239 %

```

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`

```

3240 \newcommand*{\prebodyfootmark}{%
3241   \leavevmode
3242   \ifhmode
3243     \edef\@x@sf{\the\spacefactor}%
3244     \m@mmf@check
3245     \nobreak
3246     \fi}
3247 \newcommand{\postbodyfootmark}{%
3248   \m@mmf@prepare
3249   \ifhmode\spacefactor\@x@sf\fi\relax}
3250
3251 %
```

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.

```

3252 \newcommandx{\arrangementX}[2][1,usedefault]{%
3253   \def\do##1{%
3254     \csname arrangementX@#2\endcsname{##1}%
3255   }%
3256   \ifstrempy{#1}%
3257     {%
3258       \dolistloop{\@series}%
3259     }%
3260     {
3261       \docsvlist{#1}%
3262     }%
3263   }%
3264   %
```

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.

```

3265 \newcommand*{\normal@footnotemarkX}[1]{%
3266   \prebodyfootmark
3267   \wrapped@bodyfootmarkX{#1}%
3268   \postbodyfootmark}
```

```
3269
3270 %
```

\normalbodyfootmarkX The `\normalbodyfootmarkX{⟨series⟩}` *really* typesets the in-text marker. The style is the normal superscript.

```
3271 \newcommand*{\normalbodyfootmarkX}[1]{%
3272   \hbox{\textsuperscript{\normalfont\@nameuse{@thefnmark#1}}}%
3273 }
```

\normalvfootnoteX `\normalvfootnoteX{⟨series⟩}{⟨text⟩}` does the `\insert` for the `⟨series⟩` and calls the series' `\footfmt...` to format the `⟨text⟩`.

```
3274 \notbool{parapparatus@}{\newcommand*{\normalvfootnoteX}[2]{%
3275   \insert\@nameuse{footins#1}\bgroup
3276   \hspace=\expandafter\dimexpr\csuse{widthX@#1}\relax%
3277   \noindent\csuse{bhooknoteX@#1}%
3278   \csuse{notefontsizeX@#1}%
3279   \footsplitskips
3280   \ifl@dpairing\ifl@dpaging\else%
3281     \setnoteswidthliketwocolumnsX@{#1}%
3282   \fi\fi%
3283   \setnotesXpositionliketwocolumns@{#1}%
3284   \spaceskip=\z@skip \xspaceskip=\z@skip
3285   \csuse{\csuse{footnote@dir}}\@nameuse{footfmt#1}{#1}{#2}\egroup}
3286
3287 %
```

\mpnormalvfootnoteX The minipage version.

```
3288 \newcommand*{\mpnormalvfootnoteX}[2]{%
3289   \get@thisfootnoteX{#1}%
3290   \get@fnmarkX{#1}{\thisfootnote}%
3291   \edef\this@footnoteX@reading{\the\csname footnote#1@reading\endcsname}%
3292   \global\setbox\@nameuse{mpfootins#1}\vbox{%
3293     \unvbox\@nameuse{mpfootins#1}
3294     \noindent\csuse{bhooknoteX@#1}%
3295     \csuse{notefontsizeX@#1}%
3296     \hspace\columnwidth
3297     \@parboxrestore
3298     \color@begingroup
3299     \@nameuse{footfmt#1}{#1}{#2}\color@endgroup}}
3300
3301 %
```

\normalfootfmtX `\normalfootfmtX{⟨series⟩}{⟨text⟩}` typesets the footnote text, prepended by the marker.

```
3302 \notbool{parapparatus@}{\newcommand*{\normalfootfmtX}[2]{%
3303   \ifluatex%
3304     \textdir\footnote@luatextextdir%
```

```

3305 \pardir\footnote@luatexpardir%
3306 \par%
3307 \fi%
3308 \protected@edef\@currentlabel{%
3309 \@nameuse{@thefnmark#1}%
3310 }%
3311 \ledsetnormalparstuffX{#1}%
3312 \hangindent=\csuse{hangindentX@#1}%
3313 \everypar{\hangindent=\csuse{hangindentX@#1}}%
3314 {\csuse{notenumberX@#1}\wrapped@footfootmarkX{#1}\strut%
3315 #2\strut\par}}
3316
3317 %

```

\normalfootfootmarkX `\normalfootfootmarkX{<series>}` is called by `\normalfootfmtX` to typeset the footnote marker in the footer before the footnote text.

```

3318 \newcommand*\normalfootfootmarkX[1]{%
3319 \textsuperscript{\@nameuse{@thefnmark#1}}}
3320
3321 %

```

\normalfootstartX `\normalfootstartX{<series>}` is the `<series>` footnote starting macro used in the output routine.

```

3322 \newcommand*\normalfootstartX[1]{%
3323 \ifdimequal{Opt}{\prenotesX@}{}%
3324 {%
3325 \iftoggle{prenotesX@}{%
3326 \togglefalse{prenotesX@}%
3327 \skip\csname footins#1\endcsname=%
3328 \dimexpr\csuse{prenotesX@}+\csuse{afterterruleX@#1}\relax%
3329 }%
3330 }%
3331 }%
3332 \vskip\skip\csname footins#1\endcsname%
3333 \leftskip=\z@
3334 \rightskip=\z@
3335 \ifl@dpairing\else%
3336 \hsize=\old@hsize%
3337 \fi%
3338 \setnoteswidthliketwocolumnsX@{#1}%
3339 \setnotesXpositionliketwocolumns@{#1}%
3340 \print@footnoteruleX{#1}%
3341 }%
3342
3343 %

```

\normalfootnoteruleX The rule drawn before the footnote series group.

```

3344 \let\normalfootnoteruleX=\footnoterule
3345
3346 %

```

\normalfootgroupX `\normalfootgroupX{<series>}` sends the contents of the `<series>` insert box to the output page without alteration.

```

3347 \newcommand*\normalfootgroupX[1]{%
3348   \csuse{bhookgroupX@#1}%
3349   \unvbox\@nameuse{footins#1}%
3350   \hsize=\old@hsize%
3351   }%
3352
3353 %

```

\mpnormalfootgroupX The minipage version.

```

3354 \newcommand*\mpnormalfootgroupX[1]{%
3355   \vskip\skip\@nameuse{mpfootins#1}
3356   \ifl@dpairing\ifparledgroup%
3357     \leavevmode\marks\parledgroup@{begin}%
3358     \marks\parledgroup@series{#1}%
3359     \marks\parledgroup@type{footnoteX}%
3360   \fi\fi\normalcolor
3361   \ifparledgroup%
3362     \ifl@dpairing%
3363     \else%
3364       \setnoteswidthliketwocolumnsX@{#1}%
3365       \setnotesXpositionliketwocolumns@{#1}%
3366       \print@footnoteXrule{#1}%
3367     \fi%
3368   \else%
3369     \setnoteswidthliketwocolumnsX@{#1}%
3370     \setnotesXpositionliketwocolumns@{#1}%
3371     \print@footnoteXrule{#1}%
3372   \fi%
3373   \csuse{bhookgroupX@#1}%
3374   \unvbox\@nameuse{mpfootins#1}}
3375
3376 %

```

\normalbfnoteX

```

3378 \newcommand{\normalbfnoteX}[2]{%
3379   \get@thisfootnoteX{#1}%
3380   \ifledRcol%
3381     \ifluatex
3382       \footnotelang@lua[R]%
3383     \fi
3384     \ifundefined{xpg@main@language}%if polyglossia

```

```

3385     {}%
3386     {\footnotelang@poly[R]}%
3387     \xright@appenditem{%
3388       \noexpand\led@set@index@fornote{#1}%
3389       \unexpanded{\def\this@footnoteX@reading}{\the\csname footnote#1
@reading\endcsname}%
3390       \noexpand\vbfnoteX{#1}{#2}{\expandonce\thisfootnote}%
3391       \noexpand\led@reinit@index@fornote%
3392     }%
3393         \to\inserts@listR
3394     \global\advance\insert@countR \@ne%
3395 \else%
3396     \ifluatex
3397         \footnotelang@lua%
3398     \fi
3399     \ifundefined{xpg@main@language}%if polyglossia
3400     {}%
3401     {\footnotelang@poly}%
3402     \xright@appenditem{%
3403       \noexpand\led@set@index@fornote{#1}%
3404       \unexpanded{\def\this@footnoteX@reading}{\the\csname footnote#1
@reading\endcsname}%
3405       \noexpand\vbfnoteX{#1}{#2}{\expandonce\thisfootnote}%
3406       \noexpand\led@reinit@index@fornote%
3407     }%
3408         \to\inserts@list
3409     \global\advance\insert@count \@ne%
3410 \fi
3411 \ignorespaces}
3412
3413 %

```

\get@thisfootnoteX The macro `\get@thisfootnote` command just saves the footnote number in the `\thisfootnote` macro, depending on the use of pairing environments.

```

3414 \newcommand{\get@thisfootnoteX}[1]{%
3415   \ifl@dpairing%
3416     \protected@xdef\thisfootnote{\the\csname footnote#1@reading\endcsname
}%
3417   \else%
3418     \protected@xdef\thisfootnote{\csuse{thefootnote#1}}%
3419   \fi%
3420 }%
3421 %

```

\vbfnoteX This command calls the correct footnote-inserting commands.

```

3422 \newcommand{\vbfnoteX}[3]{%
3423   \get@fnmarkX{#1}{#3}%
3424   \@nameuse{regvfootnote#1}{#1}{#2}%

```

```

3425 }%
3426
3427 %

```

\get@fnmarkX This command gets the correct footnote number when typesetting parallel texts.

```

3428 \newcommand{\get@fnmarkX}[2]{%
3429   \ifboolexpr{bool{l@dpairing} or bool{l@dprintingpages} or bool{
l@dprintingcolumns}}%
3430   {%
3431     \stepcounter{footnote#1@typeset}%
3432     \setcounter{footnote#1}{\value{footnote#1@typeset}}%
3433     \@namedef{@thefnmark#1}{\csuse{thefootnote#1}}%
3434     \immediate\write\@mainaux{%
3435       \global\csdef{footnote#1reading#2=typeset}{\the\csname c@footnote
#1@typeset\endcsname}%
3436     }%
3437   }%
3438   {%
3439     \@namedef{@thefnmark#1}{#2}%
3440   }%
3441 }
3442 %
3443 %

```

```

\newcommand{\vnumfootnoteX}[2]{%
3445   \ifnumberedpar@
3446     \edtext{}{\normalbfnoteX{#1}{#2}}%
3447   \else
3448     \def\this@footnoteX@reading{\the\csname footnote#1@reading\endcsname}%
3449     \@nameuse{regvfootnote#1}{#1}{#2}%
3450   \fi}
3451
3452 %

```

arrangementX@normal `\arrangementX@normal{<series>}` initialises the settings for the `<series>` footnotes. This should always be called for each series.

```

3453 \newcommand*{\arrangementX@normal}[1]{%
3454   \csgdef{series@display#1}{normal}
3455   \expandafter\let\csname footstart#1\endcsname=\normalfootstartX
3456   \expandafter\newcount\csname prevpage#1@num\endcsname
3457   \@namedef{@footnotemark#1}{\normal@footnotemarkX{#1}}
3458   \@namedef{@bodyfootmark#1}{\normalbodyfootmarkX{#1}}
3459   \expandafter\let\csname regvfootnote#1\endcsname=\normalvfootnoteX
3460   \expandafter\let\csname vfootnote#1\endcsname=\vnumfootnoteX
3461   \expandafter\let\csname footfmt#1\endcsname=\normalfootfmtX
3462   \@namedef{@footfootmark#1}{\normalfootfootmarkX{#1}}
3463   \expandafter\let\csname footgroup#1\endcsname=\normalfootgroupX

```

```

3464 \expandafter\let\csname footnoterule#1\endcsname=\normalfootnoteruleX
3465 \count\csname footins#1\endcsname=1000
3466 \csxdef{default@footins#1}{1000}%Use to have note only for one side
3467 \dimen\csname footins#1\endcsname=\csuse{maxhnotesX@#1}
3468 \skip\csname footins#1\endcsname=\csuse{beforenotesX@#1}%
3469 \advance\skip\csname footins#1\endcsname by\csuse{afterruleX@#1}%
3470 %

```

Additions for minipages.

```

3471 \ifnoledgroup@else%
3472 \expandafter\let\csname mpvfootnote#1\endcsname=\mpnormalvfootnoteX
3473 \expandafter\let\csname mpfootgroup#1\endcsname=\mpnormalfootgroupX
3474 \count\csname mpfootins#1\endcsname=1000
3475 \dimen\csname mpfootins#1\endcsname=\csuse{maxhnotesX@#1}
3476 \skip\csname mpfootins#1\endcsname=\csuse{beforenotesX@#1}%
3477 \advance\skip\csname mpfootins#1\endcsname by\csuse{afterruleX@#1}%
3478 \fi
3479 }
3480
3481 %

```

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 3482 \newcommand*\arrangementX@twocol}[1]{%
3483 \csgdef{series@displayX#1}{twocol}
3484 \expandafter\let\csname regvfootnote#1\endcsname=\twocolvfootnoteX
3485 \expandafter\let\csname footfmt#1\endcsname=\twocolfootfmtX
3486 \expandafter\let\csname footgroup#1\endcsname=\twocolfootgroupX
3487 \dimen\csname footins#1\endcsname=\csuse{maxhnotesX@#1}%
3488 \skip\csname footins#1\endcsname=\csuse{beforenotesX@#1}%
3489 \advance\skip\csname footins#1\endcsname by \csuse{afterruleX@#1}\relax%
3490 \twocolfootsetupX{#1}
3491 \ifnoledgroup@else%
3492 \expandafter\let\csname mpvfootnote#1\endcsname=\mpnormalvfootnoteX
3493 \expandafter\let\csname mpfootgroup#1\endcsname=\mptwocolfootgroupX
3494 \skip\csname mpfootins#1\endcsname=\csuse{beforenotesX@#1}%
3495 \advance\skip\csname mpfootins#1\endcsname by\csuse{afterruleX@#1}
3496 \mptwocolfootsetupX{#1}
3497 \fi%
3498 }
3499
3500 %

```

```

\twocolfootsetupX \twocolfootsetupX{<series>}
\mptwocolfootsetupX

```



```

3501 \newcommand*{\twocolfootsetupX}[1]{%
3502   \count\csname footins#1\endcsname 500
3503   \csxdef{default@footins#1}{500}%Use this to confine the notes to one
side only
3504   \multiply\dimen\csname footins#1\endcsname by \tw@}
3505 \newcommand*{\mptwocolfootsetupX}[1]{%
3506   \count\csname mpfootins#1\endcsname 500
3507   \multiply\dimen\csname mpfootins#1\endcsname by \tw@}
3508
3509 %

```

`\twocolvfootnoteX` `\twocolvfootnoteX{<series>}`

```

3510 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\twocolvfootnoteX}[2]{%
3511   \insert\csname footins#1\endcsname\bgroup%
3512   \hspace=\expandafter\dimexpr\csuse{widthX@#1}\relax%
3513   \noindent\csuse{bhooknoteX@#1}%
3514   \csuse{notefontsizeX@#1}%
3515   \footssplitsskip%
3516   \spaceskip=\z@skip \xspaceskip=\z@skip%
3517   \@nameuse{footfmt#1}{#1}{#2}\egroup}
3518
3519 %

```

`\twocolfootfmtX` `\twocolfootfmtX{<series>}`

```

3520 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\twocolfootfmtX}[2]{%
3521   \protected@edef\@currentlabel{%
3522     \@nameuse{@thefnmark#1}%
3523   }%
3524   \normal@pars%
3525   \hangindent=\csuse{hangindentX@#1}%
3526   \everypar{\hangindent=\csuse{hangindentX@#1}}%
3527   \hspace \csuse{hsizetwocolX@#1}%
3528   \nottoggle{parindentX@#1}{\parindent=\z@}{}%
3529   \tolerance=5000\relax%
3530   \par%
3531   \@tempdima=\parindent%
3532   \csuse{colalignX@#1}%
3533   \parindent=\@tempdima%
3534   {\csuse{notenumfontX@#1}\wrapped@footfootmarkX{#1}\strut%
3535     #2\strut\par}\allowbreak}
3536
3537 %

```

`\twocolfootgroupX` `\twocolfootgroupX{<series>}`

`\mptwocolfootgroupX`

```

3538 \newcommand*{\twocolfootgroupX}[1]{\csuse{bhookgroupX@#1}\csuse{
notefontsizeX@#1}
3539 \splittopskip=\ht\strutbox

```

```

3540 \expandafter
3541 \rigidbalanceX\csname footins#1\endcsname \tw@ \splittopskip}}
3542
3543 \newcommand*{\mptwocolfootgroupX}[1]{%
3544 \vskip\skip\@nameuse{mpfootins#1}
3545 \ifl@dpairing\ifparledgroup%
3546 \leavevmode\marks\parledgroup@{begin}%
3547 \marks\parledgroup@series{#1}%
3548 \marks\parledgroup@type{footnoteX}%
3549 \fi\fi\normalcolor
3550 \ifparledgroup%
3551 \ifl@dpairing%
3552 \else%
3553 \setnoteswidthliketwocolumnsX@{#1}%
3554 \setnotesXpositionliketwocolumns@{#1}%
3555 \print@footnoteXrule{#1}%
3556 \fi%
3557 \else%
3558 \setnoteswidthliketwocolumnsX@{#1}%
3559 \setnotesXpositionliketwocolumns@{#1}%
3560 \print@footnoteXrule{#1}%
3561 \fi%
3562 \csuse{bhookgroupX@#1}%
3563 \splittopskip=\ht\strutbox
3564 \expandafter
3565 \rigidbalanceX\csname mpfootins#1\endcsname \tw@ \splittopskip}}
3566
3567 %

```

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 3568 \newcommand*{\arrangementX@threecol}[1]{%
3569 \csgdef{series@displayX#1}{threecol}
3570 \expandafter\let\csname regvfootnote#1\endcsname=\threecolvfootnoteX
3571 \expandafter\let\csname footfmt#1\endcsname=\threecolfootfmtX
3572 \expandafter\let\csname footgroup#1\endcsname=\threecolfootgroupX
3573 \dimen\csname footins#1\endcsname=\csuse{maxhnotesX@#1}%
3574 \skip\csname footins#1\endcsname=\csuse{beforenotesX@#1}%
3575 \advance\skip\csname footins#1\endcsname by \csuse{afterruleX@#1}\relax%
3576 \threecolfootsetupX{#1}
3577 \ifnoledgroup\else%
3578 \expandafter\let\csname mpvfootnote#1\endcsname=\mpnormalvfootnoteX
3579 \expandafter\let\csname mpfootgroup#1\endcsname=\mpthreecolfootgroupX
3580 \skip\csname mpfootins#1\endcsname=\csuse{beforenotesX@#1}%
3581 \advance\skip\csname mpfootins#1\endcsname by \csuse{afterruleX@#1}
3582 \mpthreecolfootsetupX{#1}

```

```

3583 \fi%
3584 }
3585
3586 %

```

```

\threecolfootsetupX \threecolfootsetupX{<series>}
\mpthreecolfootsetupX
3587 \newcommand*{\threecolfootsetupX}[1]{%
3588   \count\csname footins#1\endcsname 333
3589   \csxdef{default@footins#1}{333}%Use this to confine the notes to one
side only
3590   \multiply\dimen\csname footins#1\endcsname by \thr@@
3591   \newcommand*{\mpthreecolfootsetupX}[1]{%
3592     \count\csname mpfootins#1\endcsname 333
3593     \multiply\dimen\csname mpfootins#1\endcsname by \thr@@
3594
3595   %

```

```

\threecolvfootnoteX \threecolvfootnoteX{<series>}{<text>}
3596 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\threecolvfootnoteX}[2]{
%
3597   \insert\csname footins#1\endcsname\bgroup%
3598     \hsize=\expandafter\dimexpr\csuse{widthX@#1}\relax%
3599     \noindent\csuse{bhooknoteX@#1}%
3600     \csuse{notefontsizeX@#1}
3601     \footsplitskips%
3602     \@nameuse{footfmt#1}{#1}{#2}\egroup
3603
3604   %

```

```

\threecolfootfmtX \threecolfootfmtX{<series>}
3605 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\threecolfootfmtX}[2]{%
3606   \protected@edef\@currentlabel{%
3607     \@nameuse{@thefnmark#1}%
3608   }%
3609   \hangindent=\csuse{hangindentX@#1}%
3610   \everypar{\hangindent=\csuse{hangindentX@#1}}%
3611   \normal@pars%
3612   \hsize \csuse{hsizethreecolX@#1}%
3613   \nottoggle{parindentX@#1}{\parindent=\z@}{}%
3614   \tolerance=5000\relax%
3615   \par%
3616   \@tempdima=\parindent%
3617   \csuse{colalignX@#1}%
3618   \parindent=\@tempdima%
3619   {\csuse{notenumfontX@#1}\wrapped@footfootmarkX{#1}\strut%
3620     #2\strut\par}\allowbreak}
3621
3622   %

```

```

\threecolfootgroupX \threecolfootgroupX{\series}
\mpthreecolfootgroupX
3623 \newcommand*\threecolfootgroupX[1]{\csuse{bhookgroupX@#1}\csuse{
notefontsizeX@#1}
3624 \splittopskip=\ht\strutbox
3625 \expandafter
3626 \rigidbalanceX\csname footins#1\endcsname \thr@@ \splittopskip}}
3627
3628 \newcommand*\mpthreecolfootgroupX[1]{\%
3629 \vskip\skip\@nameuse{mpfootins#1}
3630 \ifl@dpairing\ifparledgroup
3631 \leavevmode\marks\parledgroup@{begin}%
3632 \marks\parledgroup@series{#1}%
3633 \marks\parledgroup@type{footnoteX}%
3634 \fi\fi\normalcolor
3635 \ifparledgroup%
3636 \ifl@dpairing%
3637 \else%
3638 \setnoteswidthliketwocolumnsX@{#1}%
3639 \setnotesXpositionliketwocolumns@{#1}%
3640 \print@footnoteXrule{#1}%
3641 \fi%
3642 \else%
3643 \setnoteswidthliketwocolumnsX@{#1}%
3644 \setnotesXpositionliketwocolumns@{#1}%
3645 \print@footnoteXrule{#1}%
3646 \fi%
3647 \csuse{bhookgroupX@#1}%
3648 \splittopskip=\ht\strutbox
3649 \expandafter
3650 \rigidbalanceX\csname mpfootins#1\endcsname \thr@@ \splittopskip}}
3651
3652 \%

```

XIII.4.5 Paragraphed footnotes

The following macros set footnotes as one paragraph.

```

\arrangementX@threecol \footparagraphX{\series}
3653 \newcommand*\arrangementX@paragraph[1]{\%
3654 \csgdef{series@displayX#1}{paragraph}%
3655 \expandafter\newcount\csname #1prevpage@num\endcsname
3656 \expandafter\let\csname footstart#1\endcsname=\parafootstartX
3657 \expandafter\let\csname regvfootnote#1\endcsname=\para@vfootnoteX
3658 \expandafter\let\csname footfmt#1\endcsname=\parafootfmtX
3659 \expandafter\let\csname footgroup#1\endcsname=\para@footgroupX
3660 \expandafter\let\csname footnoterule#1\endcsname=\normalfootnoteruleX
3661 \count\csname footins#1\endcsname=1000

```

```

3662 \csxdef{default@footins#1}{1000}%Use this to confine the notes to one
side only
3663 \dimen\csname footins#1\endcsname=\csuse{maxhnotesX@#1}
3664 \skip\csname footins#1\endcsname=\csuse{beforenotesX@#1}%
3665 \advance\skip\csname footins#1\endcsname by\csuse{afterruleX@#1}%
3666 \para@footsetupX{#1}
3667 \ifnoledgroup@else
3668 \expandafter\let\csname mpvfootnote#1\endcsname=\mppara@vfootnoteX
3669 \expandafter\let\csname mpfootgroup#1\endcsname=\mppara@footgroupX
3670 \count\csname mpfootins#1\endcsname=1000
3671 \dimen\csname mpfootins#1\endcsname=\csuse{maxhnotesX@#1}
3672 \skip\csname mpfootins#1\endcsname=\csuse{beforenotesX@#1}%
3673 \advance\skip\csname mpfootins#1\endcsname by\csuse{afterruleX@#1}%
3674 \fi
3675 }
3676
3677 %

```

`\para@footsetupX` `\para@footsetupX{<series>}`

```

3678 \newcommand*{\para@footsetupX}[1]{\csuse{bhookgroupX@#1}\csuse{
notefontsizeX@#1}
3679 \setnoteswidthliketwocolumnsX@#1}%
3680 \ifcsempy{widthX@#1}%
3681 {}%
3682 {\columnwidth=\expandafter\dimexpr\csuse{widthX@#1}\relax}%
3683 \dimen0=\baselineskip
3684 \multiply\dimen0 by 1024
3685 \divide\dimen0 by \columnwidth \multiply\dimen0 by \footfudgefiddle\relax
%
3686 \expandafter
3687 \xdef\csname footfudgefactor#1\endcsname{%
3688 \expandafter\strip@pt\dimen0 }}
3689
3690 %

```

`\parafootstartX` `\parafootstartX{<series>}`

```

3691 \newcommand*{\parafootstartX}[1]{%
3692 \ifdimequal{0pt}{\prenotesX@}{}%
3693 {%
3694 \iftoggle{prenotesX@}{%
3695 \togglefalse{prenotesX@}%
3696 \skip\csname footins#1\endcsname=%
3697 \dimexpr\csuse{prenotesX@}+\csuse{afterruleX@#1}\relax%
3698 }%
3699 {}%
3700 }%
3701 \leftskip=\z@
3702 \rightskip=\z@

```

```

3703 \nottoggle{parindentX@#1}{\parindent=\z@}{}%
3704 \vskip\skip\@nameuse{footins#1}%
3705 \setnoteswidthliketwocolumnsX@{#1}%
3706 \setnotesXpositionliketwocolumns@{#1}%
3707 \print@footnoteXrule{#1}%
3708 }
3709
3710 %

```

```

\para@vfootnoteX \para@vfootnoteX{<series>}{<text>}
\mppara@vfootnoteX
3711 \newcommand*{\para@vfootnoteX}[2]{%
3712 \insert\csname footins#1\endcsname%
3713 \bgroup
3714 \csuse{notefontsizeX@#1}
3715 \footsplitskips
3716 \setbox0=\vbox{\hsize=\maxdimen%
3717 \let\bidirTL@everypar\@empty%
3718 \noindent\csuse{bhooknoteX@#1}%
3719 \@nameuse{footfmt#1}{#1}{#2}}%
3720 \setbox0=\hbox{\unvxhX{0}{#1}}%
3721 \dp0=\z@
3722 \ht0=\csname footfudgefactor#1\endcsname\wd0
3723 \box0
3724 \penalty0
3725 \egroup}
3726 \newcommand*{\mppara@vfootnoteX}[2]{%
3727 \get@thisfootnoteX{#1}%
3728 \get@fnmarkX{#1}{\thisfootnote}%
3729 \edef\this@footnoteX@reading{\the\csname footnote#1@reading\endcsname}%
3730 \global\setbox\@nameuse{mpfootins#1}\vbox{%
3731 \unvbox\@nameuse{mpfootins#1}
3732 \csuse{notefontsizeX@#1}
3733 \footsplitskips
3734 \setbox0=\vbox{\hsize=\maxdimen%
3735 \let\bidirTL@everypar\@empty%
3736 \noindent\color@begingroup%
3737 \csuse{bhooknoteX@#1}%
3738 \@nameuse{footfmt#1}{#1}{#2}\color@endgroup}%
3739 \setbox0=\hbox{\unvxhX{0}{#1}}%
3740 \dp0=\z@
3741 \ht0=\csname footfudgefactor#1\endcsname\wd0
3742 \box0
3743 \penalty0}}
3744
3745 %

```

```

\unvxhX46 \newcommand*{\unvxhX}[2]{% 2th is optional for retro-compatibility
3747 \setbox0=\vbox{\unvbox#1%

```

```

3748 \global\setbox1=\lastbox}%
3749 \unhbox1
3750 \unskip           % remove \rightskip,
3751 \unskip           % remove \parfillskip,
3752 \unpenalty        % remove \penalty of 10000,
3753 \hskip\csuse{afternoteX@#2}} % but add the glue to go between the notes
3754
3755 %

```

`\parafootfmtX` `\parafootfmtX{<series>}`

```

3756 \newcommand*{\parafootfmtX}[2]{%
3757   \protected@edef\@currentlabel{%
3758     \@nameuse{@thefnmark#1}}%
3759   }%
3760   \insertparafootsepX{#1}%
3761   \ledsetnormalparstuff@common%
3762   {\csuse{notenumfontX@#1}}%
3763   \csuse{notenumfontX@#1}}%
3764   \wrapped@footfootmarkX{#1}%
3765   \strut%
3766   #2\penalty-10}}
3767
3768 %

```

`\para@footgroupX` `\para@footgroupX{<series>}`

`\mppara@footgroupX`

```

3769 \newcommand*{\para@footgroupX}[1]{%
3770   \hsize=\expandafter\dimexpr\csuse{widthX@#1}\relax%
3771   \unvbox\cename{footins#1}\endcename
3772   \ifcsstring{raggedX@#1}{L}{\RaggedLeft}{}%
3773   \ifcsstring{raggedX@#1}{R}{\RaggedRight}{}%
3774   \makehboxofhboxes
3775   \setbox0=\hbox{\unhbox0 \removehboxes}%
3776   \csuse{bhookgroupX@#1}
3777   \csuse{notefontsizeX@#1}
3778   \unhbox0\par}
3779
3780 \newcommand*{\mppara@footgroupX}[1]{%
3781   \setnoteswidthliketwocolumnsX@{#1}%
3782   \vskip\skip\@nameuse{mpfootins#1}
3783   \ifl@dpairing\ifparledgroup
3784     \leavevmode%
3785     \leavevmode\marks\parledgroup@{begin}%
3786     \marks\parledgroup@series{#1}%
3787     \marks\parledgroup@type{footnoteX}%
3788     \fi\fi\normalcolor
3789     \ifparledgroup%
3790       \ifl@dpairing%
3791       \else%

```

```

3792 \setnoteswidthliketwocolumnsX@{#1}%
3793 \setnotesXpositionliketwocolumns@{#1}%
3794 \print@footnoteXrule{#1}%
3795 \fi%
3796 \else%
3797 \setnoteswidthliketwocolumnsX@{#1}%
3798 \setnotesXpositionliketwocolumns@{#1}%
3799 \print@footnoteXrule{#1}%
3800 \fi%
3801 \unvbox\csname mpfootins#1\endcsname
3802 \ifcsstring{raggedX@#1}{L}{\RaggedLeft}{}%
3803 \ifcsstring{raggedX@#1}{R}{\RaggedRight}{}%
3804 \makeboxofhboxes
3805 \setbox0=\hbox{\unhbox0 \removehboxes}%
3806 \csuse{bhookgroupX@#1}%
3807 \csuse{notefontsizeX@#1}%
3808 \nottoggle{parindentX@#1}{\parindent=\z@}{}%
3809 \unhbox0\par}}
3810
3811 %

```

Insertion of the footnotes separator The command `\insertparafootsepX{<series>}` must be called at the beginning of `\parafootftmX`.

```

\prevpage@num12 \newcommand{\insertparafootsepX}[1]{%
\Xinsertparafootsep13 \ifnumequal{\csuse{prevpage#1@num}}{\page@num}%
3814 {\csuse{parafootsepX@#1}}%
3815 {}%
3816 }
3817 %

```

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.

```

3818 \newcommand{\wrapped@footfootmarkX}[1]{%
3819 \ifdefined\hypertarget%
3820 \hyperlink%
3821 {\bodyfootmark#1@this@footnoteX@reading}%
3822 {\@nameuse{footfootmark#1}}%
3823 \Hy@raisedlink%
3824 \hypertarget%
3825 {\footnotemark#1@this@footnoteX@reading}%
3826 {}%
3827 }%
3828 \else%
3829 \@nameuse{footfootmark#1}%

```



```

3830 \fi%
3831 }%
3832 %

```

`\wrapped@bodyfootmarkX` `\wrapped@bodyfootmarkX` prints the footnote mark of the text body, wrapped in `hyperref` package's commands, if needed.

```

3833 \newcommand{\wrapped@bodyfootmarkX}[1]{%
3834 \ifdefined\hypertarget%
3835 \hyperlink%
3836 { \@footnotemark#1\expandafter\the\csname footnote#1@reading\
endcsname}%
3837 { \@nameuse{bodyfootmark#1}}%
3838 \Hy@raisedlink{%
3839 \hypertarget%
3840 { \@bodyfootmark#1\expandafter\the\csname footnote#1@reading\
endcsname}%
3841 {}%
3842 }%
3843 \else%
3844 \@nameuse{bodyfootmark#1}%
3845 \fi%
3846 }%
3847 %

```

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 `LuaTeX`.

That why we have defined `\ledsetnormalparstuff@common` in order to make this setting for both familiar and critical notes. This command is called by command to make specific setting to critical or familiar footnote.

```

\ledsetnormalparstuff@common 48 \newcommand*{\ledsetnormalparstuff@common}{%
\Xledsetnormalparstuff 49 \ifluatex%
\ledsetnormalparstuffX 50 \textdir\footnote@luatextextdir%
3851 \pardir\footnote@luatexpardir%
3852 \fi%
3853 \csuse{\csuse{footnote@dir}}}%
3854 \normal@pars%
3855 \parfillskip \z@ \@plus 1fil}%
3856
3857 \newcommand*{\Xledsetnormalparstuff}[1]{%

```

```

3858 \ledsetnormalparstuff@common%
3859 \nottoggle{Xparindent@#1}{\parindent=\z@}{\hspace{\parindent}}%
3860 }%
3861
3862 \newcommand*{\ledsetnormalparstuffX}[1]{%
3863 \ledsetnormalparstuff@common%
3864 \nottoggle{parindentX@#1}{\parindent=\z@}{\hspace{\parindent}}%
3865 }%
3866 %

```

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.

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.

`\old@hsize`
`\setXnoteswidthliketwocolumns@`
`\setnoteswidthliketwocolumnsX@`

```

3867
3868 \newdimen\old@hsize%
3869 \AtBeginDocument{\old@hsize=\hsize}%
3870
3871 \newcommand{\setXnoteswidthliketwocolumns@}[1]{%
3872 \global\let\hsize@fornote=\hsize%
3873 \global\old@hsize=\hsize%
3874 \let\old@columnwidth=\columnwidth%
3875 \iftoggle{Xnoteswidthliketwocolumns@#1}%
3876 {%
3877 \csuse{setwidthliketwocolumns@\columns@position}%
3878 \global\let\hsize@fornote=\hsize%
3879 }%
3880 {}%
3881 \let\hsize=\hsize@fornote%
3882 \let\columnwidth=\old@columnwidth%
3883 }%
3884
3885 \newcommand{\setnoteswidthliketwocolumnsX@}[1]{%
3886 \global\let\hsize@fornote=\hsize%
3887 \global\old@hsize=\hsize%
3888 \let\old@columnwidth=\columnwidth%
3889 \iftoggle{noteswidthliketwocolumnsX@#1}%
3890 {%
3891 \csuse{setwidthliketwocolumns@\columns@position}%
3892 \global\let\hsize@fornote=\hsize%
3893 }%
3894 {}%
3895 \let\hsize=\hsize@fornote%

```

```

3896 \let\columnwidth=\old@columnwidth%
3897 }%
3898
3899 %

```

`\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`.

```

3900 \newcommand{\setXnotespositionliketwocolumns@}[1]{%
3901 \iftoggle{Xnoteswidthliketwocolumns@#1}{%
3902 \csuse{setnotespositionliketwocolumns@\columns@position}%
3903 }{}%
3904 }%
3905
3906 \newcommand{\setnotesXpositionliketwocolumns@}[1]{%
3907 \iftoggle{noteswidthliketwocolumnsX@#1}{%
3908 \csuse{setnotespositionliketwocolumns@\columns@position}%
3909 }{}%
3910 }%
3911
3912 %

```

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.

```

\mpfnpos
\@fnpos
\@mpfnpos
3913 \def\@fnpos{familiar-critical}
3914 \def\@mpfnpos{critical-familiar}
3915 \newcommand{\fnpos}[1]{\xdef\@fnpos{#1}}
3916 \newcommand{\mpfnpos}[1]{\xdef\@mpfnpos{#1}}
3917 %

```

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 18 \newcommand{\print@Xfootnoterule}[1]{%
\print@footnotexrule 19 \vskip-\csuse{Xafterterrule@#1}%Because count in \dimen\csuse{#1footins}
3920 \nointerlineskip%
3921 \moveleft-\leftskip\vbox{\csuse{#1footnoterule}}%
3922 \nointerlineskip%
3923 \vskip\csuse{Xafterterrule@#1}%

```

```

3924 }%
3925
3926 \newcommand{\print@footnoteXrule}[1]{%
3927   \vskip-\csuse{afterruleX@#1}%Because count in \dimen\csuse{footins#1}
3928   \nointerlineskip%
3929   \moveleft-\leftskip\vbox{\csuse{footnoterule#1}}%
3930   \nointerlineskip%
3931   \vskip\csuse{afterruleX@#1}%
3932 }%
3933
3934 %

```

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.

```

3935 \newtoggle{preXnotes@}
3936 \toggletrue{preXnotes@}
3937 \newcommand{\preXnotes@}{Opt}
3938 \newcommand*{\preXnotes}[1]{\renewcommand{\preXnotes@}{#1}}
3939 %

```

The same, but for familiar footnotes.

```

\preXnotes@0 \newtoggle{prenotesX@}
\preXnotes@1 \toggletrue{prenotesX@}
3942 \newcommand{\prenotesX@}{Opt}
3943 \newcommand*{\prenotesX}[1]{\renewcommand{\prenotesX@}{#1}}
3944 %

```

XVIII.0.3 Internal commands

```

firstXseries@45 \gdef\firstXseries@{}
prepare@preXnotes46 \newcommand{\prepare@preXnotes}[1]{%
3947 \ifdimequal{Opt}{\preXnotes@}%
3948 {}%
3949 {}%
3950 \IfStrEq{\firstXseries@}{}{%
3951 \global\skip\csuse{#1footins}=\preXnotes@%
3952 \global\advance\skip\csname #1footins\endcsname by\csuse{Xafterrule@
#1}%
3953 \gdef\firstXseries@{#1}%
3954 }%
3955 {%
3956 \ifseriesbefore{#1}{\firstXseries@}%
3957 {%
3958 \global\skip\csuse{#1footins}=\csuse{Xbeforenotes@\firstXseries@}%
3959 \global\advance\skip\csname #1footins\endcsname by\csuse{Xafterrule@
#1}%
3960 \gdef\firstXseries@{#1}%
3961 }%
3962 {}%
3963 }%
3964 }%
3965 }
3966 %

```

The same thing is required for familiar notes and `\prenotesX`.

```

firstseriesX@67 \gdef\firstseriesX@{}
prepare@prenotesX68 \newcommand{\prepare@prenotesX}[1]{%
3969 \ifdimequal{0pt}{\prenotesX@}%
3970 {}%
3971 {%
3972 \IfStrEq{\firstseriesX@}{-}{%
3973 \global\skip\csuse{footins#1}=\prenotesX@%
3974 \global\advance\skip\csname footins#1\endcsname by\csuse{afterruleX@
#1}%
3975 \gdef\firstseriesX@{#1}%
3976 }%
3977 {%
3978 \ifseriesbefore{#1}{\firstseriesX@}%
3979 {%
3980 \global\skip\csuse{footins#1}=\csuse{beforenotesX@\firstseriesX@}%
3981 \global\advance\skip\csname footins#1\endcsname by\csuse{afterruleX@
#1}%
3982 \gdef\firstXseries@{#1}%
3983 }%
3984 {}%
3985 }%
3986 }%
3987 }
3988 %

```

XIX Endnotes

First, check the noend option.

```

3989 \ifbool{noend@}{-}{%Used instead of \ifnoend@ to prevent expansion problem
3990 %

```

\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.

```

3991 \newcommand{\l@dend@open}[1]{%
3992 \global\booltrue{l@dend@#1}%
3993 \expandafter\immediate%
3994 \expandafter\openout%
3995 \csname l@d@#1end\endcsname%
3996 =\jobname.#1end\relax%
3997 }%
3998 \newcommand{\l@dend@close}[1]{%
3999 \global\boolfalse{l@dend@#1}%
4000 \expandafter\immediate%

```

```

4001 \expandafter\closeout\csname l@d@#1end\endcsname%
4002 }%
4003
4004 %

```

\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.

```

4005 \newcommand{\l@dend@stuff}{%
4006 \def\do##1{%
4007 \ifbool{l@dend@##1}{}%
4008 {\l@dend@open{##1}}%
4009 \expandafter\immediate\expandafter\write\csname l@d@##1end\endcsname{\
string\l@d@section{\the\section@num}}%
4010 }%
4011 \dolistloop{\@series}%
4012 }%
4013
4014 %

```

\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.

```

4015 \global\notbool{parapparatus@}{\long}\def\endprint#1#2#3#4#5#6#7{%
4016 \csuse{Xendbhooknote@#4}%
4017 \csuse{Xendnotefontsize@#4}%
4018 \hangindent=\csuse{Xendhangindent@#4}%
4019 \ifXendinsertsep%
4020 \hskip\csuse{Xendafternote@#4}%
4021 \csuse{Xendsep@#4}%
4022 \else%
4023 \iftoggle{Xendparagraph@#4}%

```

```

4024     {\global\Xendinsertsep@true}%
4025     }%
4026 \fi%
4027 \xdef\@currentseries{#4}%
4028 \def\do##1{%
4029     \setkeys[mac]{truefootnoteoption}{##1}%
4030 }%
4031 \notblank{#5}{\docsvlist{#5}}{ }%
4032 \IfStrEq{#6}{R}{\ledRcol@true}{ }%
4033 \def\@this@crossref@start{#7:start}%
4034 \def\@this@crossref@end{#7:end}%
4035 \printlineendnote{#1}{#4}%
4036 \IfStrEq{#6}{R}{\ledRcol@false}{ }%
4037 \undef\@this@crossref@start%
4038 \undef\@this@crossref@end%
4039 \nottoggle{Xendlemmadisablefontselection@#4}%
4040     {\select@lemmafont#1|{\csuse{Xendlemmafont@#4}#2}}%
4041     {{\csuse{Xendlemmafont@#4}#2}}%
4042 \ifboolexpr{%
4043     togl {nosep@}%
4044     or test{\ifcsempy{Xendlemmaseparator@#4}}%
4045 }%
4046 {\hskip\csuse{Xendinplaceoflemmaseparator@#4}}%
4047 {\nobreak%
4048     \hskip\csuse{Xendbeforelemmaseparator@#4}%
4049     \csuse{Xendlemmaseparator@#4}%
4050     \hskip\csuse{Xendafterlemmaseparator@#4}%
4051 }%
4052 #3%
4053 \nottoggle{Xendparagraph@#4}{\par}{ }%
4054 \def\do##1{%
4055     \setkeys[mac]{falsefootnoteoption}{##1}%
4056 }%
4057 \notblank{#5}{\docsvlist{#5}}{ }%
4058 }}%
4059
4060 \let\l@d@section=\@gobble
4061
4062 %

```

\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.

```

4063 \newcommand{\printlineendnote}[2]{%
4064     \l@dp@rsefootspec#1|%
4065     \iftoggle{Xendnumberonlyfirstintwolines@#2}{%
4066         \edef\lineinfo{\l@dparsedstartpage - \l@dparsedstartline - \
4067             \l@dparsedstartsub - \l@dparsedendpage - \l@dparsedendline - \
4068             \l@dparsedendsub}%

```



```

4067 }%
4068 {%
4069 \edef\lineinfo@{\l@dparsedstartpage - \l@dparsedstartline - \
l@dparsedstartsub}%
4070 }%
4071 \ifboolexpr{%
4072 togl {nonum@}%
4073 or togl {Xendnonumber@#2}%
4074 }%
4075 {\hspace{\csuse{Xendinplaceofnumber@#2}}}%
4076 {%
4077 \iftoggle{Xendnumberonlyfirstinline@#2}%
4078 {\ifcsdef{prevendline#2}%
4079 {\ifcsequal{prevendline#2}{\lineinfo@}%
4080 {%
4081 \csuse{Xendbhookinplaceofnumber@#2}%
4082 \ifcsequal{Xendsymmlinenumber@#2}%
4083 {\hspace{\csuse{Xendinplaceofnumber@#2}}}%
4084 {\printsymlineendnotearea{#2}}%
4085 \csuse{Xendahookinplaceofnumber@#2}%
4086 }%
4087 {\printlineendnotearea{#1}{#2}}}%
4088 {\printlineendnotearea{#1}{#2}}}%
4089 }%
4090 {\printlineendnotearea{#1}{#2}}%We keep every time line
4091 \csxdef{prevendline#2}{\lineinfo@}%
4092 }%
4093 }%
4094 %

```

```

\printsymlineendnotearea#5 \newcommand{\printsymlineendnotearea}[1]{%
4096 \hspace{\csuse{Xendbeforemlinenumber@#1}}%
4097 \csuse{Xendnotenumfont@#1}%
4098 \ifdimequal{\csuse{Xendboxsymmlinenumber@#1}}{\z@}%
4099 {\csuse{Xendsymmlinenumber@#1}}%
4100 {\hbox to \csuse{Xendboxsymmlinenumber@#1}%
4101 {\csuse{Xendsymmlinenumber@#1}\hfill}%
4102 }%
4103 \hspace{\csuse{Xendaftersymmlinenumber@#1}}%
4104 }%
4105 %

```

\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.)

```

4106 \newcommand{\printlineendnotearea}[2]{%
4107 \csuse{Xendbhooklinenumber@#2}%

```

```

4108 \hspace{\csuse{Xendbeforenumber@#2}}}%
4109 \bgroup%
4110 \csuse{Xendnotenumfont@#2}%
4111 \ifdimequal{\csuse{Xendboxlinenum@#2}}{0pt}%
4112 {\printendlines#1||\ifledRcol@{\@Rlineflag\fi}%
4113 {\leavevmode%
4114 \hbox to \csuse{Xendboxlinenum@#2}%
4115 {%
4116 \IfSubStr{RC}{\csuse{Xendboxlinenumalign@#2}}{\hfill}{}%
4117 \printendlines#1||\ifledRcol@{\@Rlineflag\fi}%
4118 \IfSubStr{LC}{\csuse{Xendboxlinenumalign@#2}}{\hfill}{}%
4119 }}%
4120 \egroup%
4121 \hspace{\csuse{Xendafternumber@#2}}}%
4122 \csuse{Xendahooklinenum@#2}%
4123 }%
4124 %

```

\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.

```

4125 \newif\ifXendinsertsep@%
4126 \newcommand*{\doendnotes}[1]{%
4127 \l@dend@close{#1}%
4128 \begingroup
4129 \makeatletter
4130 \expandafter\let\csname #1end\endcsname=\endprint
4131 \input\jobname.#1end%
4132 \global\Xendinsertsep@false%
4133 \endgroup}
4134 %

```

\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.

```

4135 \newcommand*{\doendnotesbysection}[1]{%
4136 \l@dend@close{#1}%
4137 \global\expandafter\advance\csname #1end@bysection\endcsname by 1%
4138 \begingroup%
4139 \makeatletter%
4140 \def\l@d@section##1{%
4141 \ifnumequal{##1}{\csname #1end@bysection\endcsname}%
4142 {\cslet{#1end}{\endprint}}%
4143 {\cslet{#1end}{\@gobblefive}}%
4144 }%
4145 \input\jobname.#1end%

```

```

4146 \global\Xendinsertsep@false%
4147 \endgroup%
4148 }%
4149 %

```

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.

```

4150 }%
4151 %

```

\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.

```

4152 \newcommand*{\setprintendlines}[6]{%
4153 \l@dpnumfalse \l@ddashfalse
4154 \ifnum#4=#1 \else
4155 \l@dpnumtrue
4156 \l@ddashtrue
4157 \fi
4158 %

```

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.

```

4159 \ifl@dpnum \l@delintrue \else \l@delinfalse \fi
4160 \ifnum#2=#5 \else
4161 \l@delintrue
4162 \l@ddashtrue
4163 \fi
4164 %

```

We print the starting sub-line if it is nonzero.

```

4165 \l@ssubfalse
4166 \ifnum#3=0 \else
4167 \l@ssubtrue
4168 \fi
4169 %

```

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.

```

4170 \l@d@eslfalse
4171 \ifnum#6=0 \else
4172     \ifnum#6=#3
4173         \ifl@d@elin \l@d@esltrue \else \l@d@eslfalse \fi
4174     \else
4175         \l@d@esltrue
4176         \l@d@dashtrue
4177     \fi
4178 \fi%
4179 %

4180 \ifl@d@dash%
4181     \ifboolexpr{togl{fulllines@} or test{\ifcsempy{Xendtwolines@
4182 @currentseries}}}%
4183     {%
4184     {%
4185         \setistwofollowinglines{#1}{#2}{#4}{#5}%
4186         \ifboolexpr{%
4187             (%
4188                 togl {Xendtwolinesbutnotmore@ \@currentseries}%
4189                 and not%
4190                 (%
4191                     bool {istwofollowinglines@}%
4192                 )%
4193             or%
4194             (%
4195                 (not test{\ifnumequal{#1}{#4}})%
4196                 and togl{Xendtwolinesonlyinsamepage@ \@currentseries}%
4197             )%
4198             }%
4199             {%
4200             {%
4201                 \l@d@dashfalse%
4202                 \l@d@Xtwolinestrue%
4203                 \l@d@elinfalse%
4204                 \l@d@eslfalse%
4205                 \ifcsempy{Xendmoreethantwolines@ \@currentseries}%
4206                 {%
4207                     {\ifistwofollowinglines@ \else%
4208                     \l@d@Xmoreethantwolinestrue%
4209                     \fi%
4210                     }%
4211                 }%
4212             }%
4213         \fi%
4214     }%

End of \setprintendlines.

4215 }%
```

4216 %

`\printendlines` Now we are ready to print it all.

```
4217 \def\printendlines#1|#2|#3|#4|#5|#6|#7|#8|{\begingroup
4218 \setprintendlines{#1}{#2}{#3}{#4}{#5}{#6}%
4219 %
```

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.

```
4220 \ifdimequal{\csuse{Xendboxstartlinenum@\@currentseries}}{0pt}%
4221 {\bgroup}%
4222 {\leavevmode\hbox to \csuse{Xendboxstartlinenum@\@currentseries}\bgroup
\hfill}%
4223 \wrap@edcrossref{\@this@crossref@start}{\printnnum{#1}}%
4224 \ifl@d@dash%
4225 \ifl@d@pnum%
4226 \csuse{Xendlineprefixsingle@\@currentseries}%
4227 \else%
4228 \ifcempty{Xendlineprefixmore@\@currentseries}%
4229 {\csuse{Xendlineprefixsingle@\@currentseries}}
4230 {\csuse{Xendlineprefixmore@\@currentseries}}%
4231 \fi%
4232 \else%
4233 \csuse{Xendlineprefixsingle@\@currentseries}%
4234 \fi%
4235 \wrap@edcrossref{\@this@crossref@start}{\linenumrep{#2}}%
4236 \iftoggle{Xendlineflag@\@currentseries}{\ifledRcol@\@Rlineflag\fi}{}%
4237 \ifl@d@ssub%
4238 \csuse{Xendsublinesep@\@currentseries}%
4239 \wrap@edcrossref{\@this@crossref@start}{\sublinenumrep{#3}}%
4240 \fi%
4241 \egroup%
4242 %
```

And now, print the dash + the end line number, or the line number range symbol.

```
4243 \ifdimequal{\csuse{Xendboxendlinenum@\@currentseries}}{0pt}%
4244 {\bgroup}%
4245 {\hbox to \csuse{Xendboxendlinenum@\@currentseries}\bgroup}%
4246 \ifl@d@Xtwolines%
4247 \ifl@d@Xmorethantwolines%
4248 \csuse{Xendmorethantwolines@\@currentseries}%
4249 \else%
4250 \csuse{Xendtwolines@\@currentseries}%
4251 \fi%
4252 \else%
```

```

4253 \ifl@d@dash%
4254 \ifdefined\linrangesep@%
4255 \linrangesep@%
4256 \else%
4257 \csuse{Xendlinrangeseparator@ \@currentseries}%
4258 \fi%
4259 \fi%
4260 \ifl@d@pnum%
4261 \wrap@edcrossref{\@this@crossref@end}\printnpnum{#4}%
4262 \fi%
4263 \ifl@d@elin%
4264 \ifl@d@pnum\csuse{Xendlineprefixsingle@ \@currentseries}\fi%
4265 \wrap@edcrossref{\@this@crossref@end}{\linenumrep{#5}}%
4266 \iftoggle{Xendlineflag@ \@currentseries}{\ifledRcol@ \@Rlineflag\fi}{}%
4267 \fi%
4268 \ifl@d@esl%
4269 \ifl@d@elin%
4270 \csuse{Xendsublinesep@ \@currentseries}%
4271 \fi%
4272 \wrap@edcrossref{\@this@crossref@end}{\sublinenumrep{#6}}%
4273 \fi%
4274 \fi%
4275 \ifdimequal{\csuse{Xendboxendlinenum@ \@currentseries}}{0pt}%
4276 {}%
4277 {\hfill}%Prevent underfull hbox
4278 \egroup%
4279 \endgroup%
4280 }%
4281 %
4282 %

```

\printnpnum A macro to print a page number in an endnote. Should not be override anymore

```

4283 \newcommand*{\printnpnum}[1]{\csuse{Xendbeforepagenumber@ \@currentseries
4284 }#1\csuse{Xendafterpagenumber@ \@currentseries}}
4285 %

```

XX Generate series of notes

In this section, X means the name of the series (A, B etc.)

\series \series\series creates one more new series. It is a public command, which just loops on the private command \newseries@.

```

4286 \newcommand{\newseries}[1]{%
4287 \def\do##1{\newseries@{##1}}%
4288 \docsvlist{#1}
4289 }
4290 %

```

`\@series` The `\series@` macro is an etoolbox list, which contains the name of all series.

```
4291 \newcommand{\@series}{}
4292 %
```

The command `\newseries@series` creates a new series of the footnote.

```
\newseries@93 \newcommand{\newseries@}[1]{
4294 %
```

XX.1 Test if series is still existing

```
4295 \xifinlist{#1}{\@series}{\led@warn@SeriesStillExist{#1}}%
4296 {%
4297 %
```

XX.2 Init specific to reledpar

When calling `\newseries@` after having loaded `reledpar`, we need to load specific setting.

```
4298 \ifdefined\newseries@par%
4299 \newseries@par{#1}%
4300 \fi%
4301 %
```

XX.3 For critical footnotes

Critical footnotes are those which start with letters. We look for the `\nocritical` option of `reledmac`.

```
4302 \unless\ifnocritical@
4303 %
```

XX.3.1 Options

```
4304 \newtoggle{Xlineflag@#1}
4305 \newtoggle{Xparindent@#1}
4306 \newtoggle{Xlemmadisablefontselection@#1}
4307 \csgdef{Xhangindent@#1}{Opt}%
4308 \csgdef{Xragged@#1}{}%
4309 \csgdef{Xhsizetwocol@#1}{0.45 \hsize}%
4310 \csgdef{Xhsizethreecol@#1}{.3 \hsize}%
4311 \csgdef{Xcolalign@#1}{\raggedright}%
4312 \csgdef{Xnotenumfont@#1}{\normalfont}%
4313 \csgdef{Xnotefontsize@#1}{\footnotesize}%
4314 \csgdef{Xbhooknote@#1}{}%
4315 \csgdef{Xbhookgroup@#1}{}%
```

```

4316 \csgdef{Xboxlinenum@#1}{Opt}%
4317 \csgdef{Xboxlinenumalign@#1}{L}%
4318
4319
4320 \csgdef{Xboxstartlinenum@#1}{Opt}%
4321 \csgdef{Xboxendlinenum@#1}{Opt}%
4322
4323 \csgdef{Xboxsymlinenum@#1}{Opt}%
4324 \newtoggle{Xnumberonlyfirstinline@#1}%
4325 \newtoggle{Xnumberonlyfirstintwelines@#1}%
4326 \csgdef{Xtwolines@#1}{}%
4327 \csgdef{Xmorethantwolines@#1}{}%
4328 \csgdef{Xsublinesep@#1}{\fullstop}%
4329 \newtoggle{Xtwolinesbutnotmore@#1}%
4330 \newtoggle{Xtwolinesonlyinsamepage@#1}%
4331 \newtoggle{Xonlypstart@#1}%
4332 \newtoggle{Xpstarteverytime@#1}%
4333 \newtoggle{Xpstart@#1}%
4334 \newtoggle{Xstanza@#1}%
4335 \csgdef{Xstanzaseparator@#1}{}%
4336 \csgdef{Xsymlinenum@#1}{}%
4337 \newtoggle{Xnonumber@#1}%
4338 \csgdef{Xbeforenumber@#1}{Opt}%
4339 \csgdef{Xtxtbeforenumber@#1}{}%
4340 \csgdef{Xafternumber@#1}{0.5em}%
4341 \newtoggle{Xnonbreakableafternumber@#1}%
4342 \csgdef{Xbeforesymlinenum@#1}{\csuse{Xbeforenumber@#1}}%
4343 \csgdef{Xaftersymlinenum@#1}{\csuse{Xafternumber@#1}}%
4344 \csgdef{Xinplaceofnumber@#1}{1em}%
4345 \global\cslet{Xlemmaseparator@#1}{\rbracket}%
4346 \csgdef{Xbeforelemmaseparator@#1}{0em}%
4347 \csgdef{Xafterlemmaseparator@#1}{0.5em}%
4348 \csgdef{Xinplaceoflemmaseparator@#1}{1em}%
4349 \csgdef{Xbeforenotes@#1}{1.2em \@plus .6em \@minus .6em}%
4350 \csgdef{Xafterrule@#1}{Opt}%
4351 \csgdef{Xtxtbeforenotes@#1}{}%
4352 \csgdef{Xmaxhnotes@#1}{0.8\vsizel}%
4353 \newtoggle{Xnoteswidthliketwocolumns@#1}%
4354 \csgdef{Xparafootsep@#1}{}%
4355 \csgdef{Xafternote@#1}{1em plus.4em minus.4em}%
4356 \csgdef{Xlinerrangeseparator@#1}{\endashchar}%
4357
4358 \csgdef{Xlemmafont@#1}{}%
4359 \csgdef{Xwidth@#1}{\hsizel}%
4360 %

```

XX.3.2 Create inserts, needed to add notes in foot

As regards inserts, see chapter 15 of *The TeXbook* by D. Knuth.

```

4361 \expandafter\newinsert\csname #1footins\endcsname%

```



```

4362 \unless\ifnoledgroup{%
4363   \expandafter\newinsert\csname mp#1footins\endcsname%
4364   \fi%
4365 %

```

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.

```

4366 \global\newbool{parapparatus@}{\expandafter\newcommand\expandafter
*}{\expandafter\newcommand}\csname #1footnote\endcsname[2][]{%
4367   \ifnum\@edtext@level>0%
4368     \begingroup%
4369     \newcommand{\content}{##2}%
4370     \ifnumberedpar{%
4371       \ifledRcol%
4372         \ifluatex%
4373           \footnotelang@lua[R]%
4374         \fi%
4375       \ifundefined{xpg@main@language}%if polyglossia
4376         {}%
4377         {\footnotelang@poly[R]}%
4378       \footnoteoptions@{R}{##1}{true}%
4379       \xright@appenditem{%
4380         \ifbool{indtl@innote}%
4381           {\unexpanded{\let\index\nindex}}%
4382           {}%
4383         \ifbool{indtl@notenumber}%
4384           {\unexpanded{\let\index\nindex}}%There is no note
...number so
4385         {}%
4386         \noexpand\Xnote@true%
4387         \noexpand\prepare@preXnotes{#1}%
4388         \noexpand\prepare@edindex@fornote{\l@d@nums}%
4389         \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.
4390         \noexpand\setcounter{stanzaR}{\the\c@stanzaR}%Save
stanzaR counter for footnote
4391         \unexpanded{\def\@this@crossref@start}{\theedtext:
start}%
4392         \unexpanded{\def\@this@crossref@end}{\theedtext:end}%
4393         \noexpand\csuse{v#1footnote}{#1}%
4394         {\l@d@nums}\expandonce\@tag{\expandonce\content}}
%
4395         \noexpand\Xnote@false%
4396         \unexpanded{\undef\@this@crossref@start}%
4397         \unexpanded{\undef\@this@crossref@end}%

```

```

4398         \ifbool{indtl@innote}%
4399             {\unexpanded{\let\index\orig@@index}}%
4400             {}%
4401         \ifbool{indtl@notenumber}%
4402             {\unexpanded{\let\index\orig@@index}}%
4403             {}%
4404         }\to\inserts@listR
4405         \footnoteoptions@{R}{##1}{false}%
4406         \global\advance\insert@countR \@ne%
4407     \else%
4408         \ifluatex%
4409             \footnotelang@lua%
4410             \fi%
4411         \ifundefined{xpg@main@language}%if polyglossia
4412             {}%
4413             {\footnotelang@poly}%
4414         \footnoteoptions@{L}{##1}{true}%
4415         \xright@appenditem{%
4416             \ifbool{indtl@innote}%
4417                 {\unexpanded{\let\index\nindex}}%
4418                 {}%
4419             \ifbool{indtl@notenumber}%
4420                 {\unexpanded{\let\index\nindex}}%There is no note
...number so
4421             {}%
4422             \noexpand\Xnote@true%
4423             \noexpand\prepare@preXnotes{##1}%
4424             \noexpand\prepare@edindex@fornote{\l@d@nums}%
4425             \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.
4426             \ifl@dpairing%
4427                 \noexpand\setcounter{stanzaL}{\the\c@stanzaL}%Save
stanzaR counter for footnote
4428             \fi%
4429             \unexpanded{\def\@this@crossref@start}{\theedtext:
start}%
4430             \unexpanded{\def\@this@crossref@end}{\theedtext:end}%
4431             \noexpand\csuse{v#1footnote}%
4432                 {##1}%
4433                 {\l@d@nums}{\expandonce\@tag}{\expandonce\content
}}%
4434             \unexpanded{\undef\@this@crossref@start}%
4435             \unexpanded{\undef\@this@crossref@end}%
4436             \noexpand\Xnote@false%
4437             \ifbool{indtl@innote}%
4438                 {\unexpanded{\let\index\orig@@index}}%
4439                 {}%
4440             \ifbool{indtl@notenumber}%

```

```

4441         {\unexpanded{\let\index\orig@@index}}}%
4442         {}%
4443         }\to\inserts@list
4444         \global\advance\insert@count \@ne%
4445         \footnoteoptions@{L}{##1}{false}%
4446     \fi
4447     \else
4448         \csuse{v#1footnote}{#1}{0|0|0|0|0|0|0|0}{#1}%
4449     \fi%
4450 \endgroup%
4451 \else%
4452     \led@err@FootnoteWithoutEdtext%
4453 \fi%
4454 \ignorespaces%
4455 }
4456 %

```

We need to be able to modify `reledmac`'s footnote macros and restore their

```

4457     \global\csletcs{#1@@footnote}{#1footnote}
4458 %

```

XX.3.4 Set standard display

```

4459     \Xarrangement@normal{#1}%
4460 %

```

End of for critical footnotes.

```

4461     \fi
4462 %

```

XX.4 For familiar footnotes

Familiar footnotes are those which end with letters. We look for the `nofamiliar` option of `reledmac`.

```

4463     \unless\ifnofamiliar@
4464 %

```

XX.4.1 Options

```

4465     \newtoggle{parindentX@#1}
4466     \csgdef{hangindentX@#1}{Opt}%
4467     \csgdef{raggedX@#1}{}%
4468     \csgdef{hsizetwocolX@#1}{0.45 \hsize}%
4469     \csgdef{hsizethreecolX@#1}{.3 \hsize}%
4470     \csgdef{colalignX@#1}{\raggedright}%
4471     \csgdef{notenumfontX@#1}{\normalfont}%
4472     \csgdef{notefontsizeX@#1}{\footnotesize}%

```

```

4473 \csgdef{bhooknoteX@#1}{}%
4474 \csgdef{bhookgroupX@#1}{}%
4475 \csgdef{afterruleX@#1}{Opt}%
4476 \csgdef{beforenotesX@#1}{1.2em \@plus .6em \@minus .6em}%
4477 \csgdef{maxhnotesX@#1}{0.8\vsizex}%
4478 \newtoggle{noteswidthliketwocolumnsX@#1}%
4479 \csgdef{parafootsepX@#1}{}%
4480 \csgdef{afternoteX@#1}{1em plus.4em minus.4em}%
4481 \csgdef{widthX@#1}{\hsizex}%
4482 % End of for familiar footnotes.
4483 % \subsubsection{Create inserts, needed to add notes in foot}
4484 % As regards inserts, see chapter 15 of the TeXBook by D. Knuth.
4485 % \begin{macrocode}
4486 \expandafter\newinsert\csname footins#1\endcsname%
4487 \unless\ifnoledgroup%
4488 \expandafter\newinsert\csname mpfootins#1\endcsname%
4489 \fi%
4490 %

```

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.

```

4491 \global\expandafter\newcommand\csname footnote#1\endcsname[1]{%
4492 \begin{group}%
4493 \prepare@pnotesX{#1}%
4494 \newcommand{\content}{##1}%
4495 %
4496 %

```

If we are preparing parallel typesetting, we cannot just increase the footnote counter. Read `reledpar`'s handbook about that (V.2.1 p. 44).

```

4497 \global\expandafter\advance\csname footnote#1@reading\
endcsname by \@ne%
4498 \ifl@dpairing%
4499 \ifcsdef{footnote#1@reading\the\csname footnote#1@reading\
endcsname=typeset}%
4500 {\setcounter{footnote#1}{\csuse{footnote#1@reading\the\
csname footnote#1@reading\endcsname=typeset}}}%
4501 {\setcounter{footnote#1}{\the\csname footnote#1@reading\
\endcsname}}}%
4502 \else%
4503 \stepcounter{footnote#1}%
4504 \fi%
4505 %

```

And now, the feature not depending of whether we are preparing parallel typesetting

```

4506 \protected@csxdef{@thefnmark#1}{\csuse{thefootnote#1}}%
4507 \nottoggle{nomk@}%Nomk is set to true when using \
footnoteXnomk with \parpackage

```

```

4508         {\csuse{@footnotemark#1}}%
4509         {}%
4510     \ifluatex%
4511         \xdef\footnote@luatextextdir{\the\textdir}%
4512         \xdef\footnote@luatexpardir{\the\pardir}%
4513     \fi%
4514     \if@ledgroup%
4515         \led@set@index@fornote{#1}%
4516     \fi%
4517     \csuse{vfootnote#1}{#1}{\expandonce\content}\m@mmf@prepare%
4518     \ifbool{indtl@innote}%
4519         {\let\index\orig@@index}%
4520         {}%
4521     \ifbool{indtl@notenumber}%
4522         {\let\index\orig@@index}%
4523         {}%
4524     \endgroup%
4525 }
4526 %

```

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. 44).

```

4527     \newcounter{footnote#1}
4528     \global\expandafter\renewcommand\csname thefootnote#1\endcsname{\
arabic{footnote#1}}
4529     \expandafter\newcount\csname footnote#1@reading\endcsname%
4530 %

```

Do not forget to initialize series

```

4531     \arrangementX@normal{#1}%
4532     \fi
4533 %

```

XX.5 The endnotes

Endnotes are commands like `\Xendnote`, where `X` is a series letter. First, we check for the `noend` options.

```

4534     \unless\ifnoend@
4535 %

```

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`

```

4536 \expandafter\newwrite\csname l@d@#1end\endcsname%
4537 \expandafter\newif\csname ifl@dend@#1\endcsname%
4538 %

```

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.

```

4539
4540 \global\expandafter\newcommand\csname #1endnote\endcsname[2][1,
usedefault]{%
4541 \bgroup%
4542 \newlinechar='40%
4543 \global\@noneed@Footnotetrue%
4544 \newcommand{\content}{##2}%
4545 \stepcounter{labidx}%
4546 \expandafter\immediate\expandafter\write\csname l@d@#1end\
endcsname{%
4547 \expandafter\string\csname #1end\endcsname%
4548 {\ifnumberedpar@l@d@nums\fi}%
4549 {\ifnumberedpar@expandonce\@tag\fi}%
4550 {\expandonce\content}%
4551 {\#1}%
4552 {\unexpanded{##1}}%
4553 {\ifledRcol R\else L\fi}%
4554 {\theedtext}%
4555 \@percentchar%
4556 }%
4557 \egroup%
4558 \ignorespaces%
4559 }%
4560 %

```

`\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.

```

4561
4562 \global\cslet{#1end}{\@gobblefive}
4563 %

```

We need to store the number of times `\doendnotesbysection` is called for one series.

```

4564 \global\expandafter\newcount\csname #1end@bysection\endcsname%
4565 %

```

XX.5.3 The options

```

4566 \csgdef{Xendtwolines@#1}{}%
4567 \csgdef{Xendmoreethantwolines@#1}{}%
4568 \newtoggle{Xendtwolinesbutnotmore@#1}{}%
4569 \newtoggle{Xendtwolinesonlyinsamepage@#1}{}%
4570 \newtoggle{Xendlemmadisablefontselection@#1}%
4571 \csgdef{Xendnotenumfont@#1}{\normalfont}%
4572 \csgdef{Xendnotefontsize@#1}{\footnotesize}%
4573 \csgdef{Xendbhooknote@#1}{}%
4574
4575 \csgdef{Xendsublinesep@#1}{\fullstop}%
4576
4577 \csgdef{Xendbeforenumber@#1}{Opt}
4578 \csgdef{Xendafternumber@#1}{0.5em}
4579
4580 \csgdef{Xendboxlinenum@#1}{Opt}%
4581 \csgdef{Xendboxlinenumalign@#1}{L}%
4582
4583 \csgdef{Xendboxstartlinenum@#1}{Opt}%
4584 \csgdef{Xendboxendlinenum@#1}{Opt}%
4585
4586 \csgdef{Xendlemmaseparator@#1}{}%
4587 \csgdef{Xendbeforelemmaseparator@#1}{0em}%
4588 \csgdef{Xendafterlemmaseparator@#1}{0.5em}%
4589 \csgdef{Xendinplaceoflemmaseparator@#1}{0.5em}%
4590
4591 \newtoggle{Xendparagraph@#1}%
4592 \csgdef{Xendafternote@#1}{1em plus.4em minus.4em}%
4593 \csgdef{Xendsep@#1}{}%
4594
4595 \csgdef{Xendinplaceofnumber@#1}{Opt}%
4596 \newtoggle{Xendnonumber@#1}%
4597
4598 \csgdef{Xendhangindent@#1}{Opt}%
4599 \newtoggle{Xendnumberonlyfirstinline@#1}%
4600 \newtoggle{Xendnumberonlyfirstintwolines@#1}%
4601
4602 \csgdef{Xendbeforesymlinenum@#1}{\csuse{Xendbeforenumber@#1}}%
4603 \csgdef{Xendaftersymlinenum@#1}{\csuse{Xendafternumber@#1}}%
4604 \csgdef{Xendsymlinenum@#1}{}%
4605 \csgdef{Xendboxsymlinenum@#1}{Opt}%
4606
4607 \csgdef{Xendbhooklinenumber@#1}{}%
4608 \csgdef{Xendehooklinenumber@#1}{}%
4609 \csgdef{Xendbhookinplaceofnumber@#1}{}%
4610 \csgdef{Xendehookinplaceofnumber@#1}{}%
4611
4612 \csgdef{Xendlinerangeseparator@#1}{\endashchar}%
4613

```

```

4614 \csgdef{Xendbeforepagenumber@#1}{p.}%
4615 \csgdef{Xendafterpagenumber@#1}{) }%
4616 \csgdef{Xendlineprefixsingle@#1}{}%
4617 \csgdef{Xendlineprefixmore@#1}{}%
4618
4619 \newtoggle{Xendlineflag@#1}
4620
4621 \csgdef{Xendlemmafont@#1}{}%
4622 %

```

End of endnotes declaration

```

4623 \fi%
4624 %

```

Dump series in \@series

```

4625 \listxadd{\@series}{#1}
4626 }
4627 }% End of \newseries
4628 %

```

XX.6 Init standards series (A,B,C,D,E)

```

4629 \expandafter\newseries\expandafter{\default@series}
4630 %

```

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.

```

4631 \newcommand{\seriesatbegin}[1]{%
4632 \StrDel{\@series}{#1}[\@series]%
4633 \edef\@new{}%
4634 \listadd{\@new}{#1}%
4635 \listadd{\@new}{\@series}%
4636 \xdef\@series{\@new}%
4637 }
4638 %

```

\seriesatend And `\seriesatend` moves the series to the end of the list.

```

4639 \newcommand{\seriesatend}[1]{%
4640 \StrDel{\@series}{#1}[\@series]%
4641 \edef\@new{}%
4642 \listadd{\@new}{\@series}%
4643 \listadd{\@new}{#1}%
4644 \xdef\@series{\@new}%
4645 }
4646 %

```


XXI.2 Test series order

`\ifseriesbefore` `\ifseriesbefore{<seriesA>}{<seriesB>}{<true>}{<false>}` expands `<true>` if `<seriesA>` is printed before `<seriesB>`, expands `<false>` otherwise.

```

4647 \newcommand{\ifseriesbefore}[4]{%
4648   \StrPosition{\@series}{#1}[\@first]%
4649   \StrPosition{\@series}{#2}[\@second]%
4650   \ifnumgreater{\@second}{\@first}{#3}{#4}%
4651 }
4652 %

```

XXI.2.1 Get the first series

In some specific case, we need to know the first series of the list of series.

```

\@getfirstseries53 \newcommand{\@getfirstseries}{%
4654   \ifdefempty{\@series}%
4655   {\xdef\@firstseries{}}%
4656   {\StrChar{\@series}{1}[\@firstseries]}%
4657 }%
4658 %

```

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`.

```

4659 \newcommandx{\settoggle@series}[5][4,5,usedefault]{%
4660   \def\do##1{%
4661     \global\settoggle{#2@##1}{#3}%
4662     \ifstrequal{#4}{critical}{
4663       \csuse{Xarrangement@\csuse{series@display##1}}{##1}%
4664     }{}
4665     \ifstrequal{#4}{familiar}{
4666       \csuse{arrangementX@\csuse{series@displayX##1}}{##1}%
4667     }{}
4668   }%
4669   \ifstreempty{#1}{%
4670     \dolistloop{\@series}%
4671     \ifstreempty{#5}{}{%
4672       \docsvlist{#5}%
4673     }
4674   }%
4675   {%
4676     \docsvlist{#1}%
4677   }%
4678 }
4679 %

```

`\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`.

```

4680 \newcommandx{\setcommand@series}[5][4,5,usedefault]{%
4681   \def\do##1{
4682     \csgdef{#2@##1}{#3}
4683     \ifstrequal{#4}{critical}{%
4684       \csuse{Xarrangement@}\csuse{series@display##1}}{##1}%
4685     }{}
4686     \ifstrequal{#4}{familiar}{%
4687       \csuse{arrangementX@}\csuse{series@displayX##1}}{##1}%
4688     }{}%
4689   }%
4690   \ifstreempty{#1}{%
4691     \dolistloop{\@series}%
4692     \ifstreempty{#5}{}{%
4693       \docsvlist{#5}
4694     }
4695   }%
4696   {%
4697     \docsvlist{#1}%
4698   }%
4699 }%
4700 %

```

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><pseudoseris>`, like `\Xtwolinesappref`.

```

4701 \newcommandx{\newhookcommand@series}[2][2,usedefault]{%
4702   \global\expandafter\newcommand\expandafter*\csname #1\endcsname[2][]{%
4703     \setcommand@series{##1}{#1}{##2}[] [#2]%
4704   }%
4705   \ifstreempty{#2}{}{%
4706     \def\do##1{%
4707       \global\expandafter\newcommand\expandafter*\csname #1##1\endcsname
4708       [1]{%
4709         \csuse{#1}[##1]{####1}%
4709       }%
4710     }%
4711     \docsvlist{#2}%
4712   }%
4713 }%
4714 %

```

\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><pseudoserious>`, like `\Xtwolinesbutnotmoreappref`.

```

4715 \newcommandx{\newhooktoggle@series}[2][2,usedefault]{%
4716   \global\expandafter\newcommandx\expandafter*\csname #1\endcsname[2][1,2={
true},usedefault]{%
4717     \settoggle@series{##1}{#1}{##2}[][#2]%
4718   }%
4719   \ifstrempy{#2}{%
4720     \def\do##1{%
4721       \global\expandafter\newcommand\expandafter*\csname #1##1\endcsname{%
4722         \csuse{#1}[##1]%
4723       }%
4724     }%
4725     \docsvlist{#2}%
4726   }%
4727 }
4728 %

```

`\newhooktoggle@series@reload` `\newhookcommand@toggle@reload` does the same thing as `\newhooktoggle@series` but the commands created by this macro also reload the series arrangement, depending of type os notes

```

4729 \newcommand{\newhooktoggle@series@reload}[2]{%
4730   \global\expandafter\newcommandx\expandafter*\csname #1\endcsname[2][1,2={
true},usedefault]{%
4731     \settoggle@series{##1}{#1}{##2}[#2]%
4732   }%
4733 }%
4734 %

```

`\newhookcommand@series@reload` `\newhookcommand@series@reload` does the same thing as `\newhookcommand@series` but the commands created by this macro also reload the series' arrangement.

```

4735 \newcommand{\newhookcommand@series@reload}[2]{%
4736   \global\expandafter\newcommand\expandafter*\csname #1\endcsname[2][]{%
4737     \setcommand@series{##1}{#1}{##2}[#2]%
4738   }%
4739 }
4740 %

```

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.

```

4741 \unless\ifnocritical@
4742   \newhookcommand@series{Xlemmafont}%
4743   \newhooktoggle@series{Xparindent}

```

```

4744 \newhookcommand@series{Xhangindent}
4745 \newhookcommand@series{Xragged}
4746 \newhookcommand@series{Xhsizetwocol}
4747 \newhookcommand@series{Xhsizethreecol}
4748 \newhookcommand@series{Xcolalign}%
4749 \newhookcommand@series{Xnotenumfont}
4750 \newhookcommand@series{Xbhooknote}
4751 \newhookcommand@series@reload{Xbhookgroup}{critical}
4752 \newhookcommand@series{Xboxsymlinenum}%
4753 \newhookcommand@series{Xsymlinenum}
4754 \newhookcommand@series{Xbeforenumber}
4755 \newhookcommand@series{Xtxtbeforenumber}
4756 \newhookcommand@series{Xafternumber}
4757 \newhookcommand@series{Xbeforesymlinenum}
4758 \newhookcommand@series{Xaftersymlinenum}
4759 \newhookcommand@series{Xinplaceofnumber}
4760 \newhookcommand@series{Xlemmaseparator}
4761 \newhookcommand@series{Xbeforelemmaseparator}
4762 \newhookcommand@series{Xafterlemmaseparator}
4763 \newhookcommand@series{Xinplaceoflemmaseparator}
4764 \newhookcommand@series{Xtxtbeforenotes}
4765 \newhookcommand@series@reload{Xafterrule}{critical}
4766 \newhooktoggle@series{Xnumberonlyfirstinline}
4767 \newhooktoggle@series{Xnumberonlyfirstintwolines}
4768 \newhooktoggle@series{Xnonumber}
4769 \newhooktoggle@series{Xpstart}
4770 \newhooktoggle@series{Xpstarteverytime}%
4771
4772 \newhooktoggle@series{Xstanza}%
4773 \newhookcommand@series{Xstanzaseparator}%
4774
4775 \newhooktoggle@series{Xonlypstart}
4776 \newhooktoggle@series{Xnonbreakableafternumber}
4777 \newhooktoggle@series{Xlemmadisablefontselection}
4778 \newhookcommand@series@reload{Xmaxhnotes}{critical}
4779 \newhookcommand@series@reload{Xbeforenotes}{critical}
4780 \newhooktoggle@series@reload{Xnoteswidthliketwocolumns}{critical}%
4781 \newhookcommand@series@reload{Xnotefontsize}{critical}
4782
4783 \newhookcommand@series{Xboxlinenum}%
4784 \newhookcommand@series{Xboxlinenumalign}%
4785
4786 \newhookcommand@series{Xboxstartlinenum}%
4787 \newhookcommand@series{Xboxendlinenum}%
4788
4789 \newhookcommand@series{Xafternote}%
4790 \newhookcommand@series{Xparafootsep}
4791
4792 \newhookcommand@series@reload{Xwidth}{critical}%
4793

```

```

4794 \ifundef{\Xhsize}%
4795 {%
4796   \newcommand{\Xhsize}[2][1,usedefault]{%
4797     \led@warning@Xhsize@deprecated%
4798     \Xwidth[#1]{#2}%
4799   }%
4800 }%
4801 {}%
4802 \fi
4803 \newhooktoggle@series{Xlineflag}[appref,Seref]
4804 \newhookcommand@series{Xtwolines}[appref,Seref]
4805 \newhookcommand@series{Xmorethantwolines}[appref,Seref]
4806 \newhookcommand@series{Xsublinesep}[appref,Seref,side]
4807 \newhooktoggle@series{Xtwolinesbutnotmore}[appref,Seref]
4808 \newhooktoggle@series{Xtwolinesonlyinsamepage}[appref,Seref]
4809 \newhookcommand@series{Xlinerrangeseparator}[appref,Seref]
4810 %

```

XXI.3.5 Options for familiar notes

Before generating the optional commands for familiar notes, we check the `\nofamiliar` option.

```

4811 \unless\ifnofamiliar@
4812   \newhooktoggle@series{parindentX}
4813   \newhookcommand@series{hangindentX}
4814   \newhookcommand@series{raggedX}
4815   \newhookcommand@series{hsize#wocolX}
4816   \newhookcommand@series{hsize#threecolX}
4817   \newhookcommand@series{colalignX}%
4818   \newhookcommand@series{notenumfontX}
4819   \newhookcommand@series{bhooknoteX}
4820   \newhookcommand@series@reload{bhookgroupX}{familiar}
4821   \newhookcommand@series@reload{beforenotesX}{familiar}
4822   \newhookcommand@series@reload{maxhnotesX}{familiar}
4823   \newhooktoggle@series@reload{noteswidthliketwocolumnsX}{familiar}%
4824   \newhookcommand@series@reload{afterruleX}{familiar}
4825   \newhookcommand@series@reload{notefontsizeX}{familiar}
4826   \newhookcommand@series{afternoteX}
4827   \newhookcommand@series{parafootsepX}
4828   \newhookcommand@series@reload{widthX}{familiar}%
4829   \ifundef{\hsizeX}%
4830   {%
4831     \newcommand{\hsizeX}[2][1,usedefault]{%
4832       \led@warning@hsizeX@deprecated%
4833       \widthX[#1]{#2}%
4834     }%
4835   }%
4836   {}%
4837 \fi

```

4838 %

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.

```

4839 \unless\ifnoend@
4840   \newhookcommand@series{Xendnotenumfont}%
4841   \newhookcommand@series{Xendlemmafont}%
4842   \newhookcommand@series{Xendbhooknote}%
4843
4844   \newhookcommand@series{Xendboxlinenum}%
4845   \newhookcommand@series{Xendboxlinenumalign}%
4846
4847   \newhookcommand@series{Xendboxstartlinenum}%
4848   \newhookcommand@series{Xendboxendlinenum}%
4849
4850   \newhookcommand@series{Xendnotefontsize}%
4851   \newhooktoggle@series{Xendlemmadisablefontselection}%
4852   \newhookcommand@series{Xendlemmaseparator}%
4853   \newhookcommand@series{Xendbeforelemmaseparator}%
4854   \newhookcommand@series{Xendafterlemmaseparator}%
4855   \newhookcommand@series{Xendinplaceoflemmaseparator}%
4856
4857   \newhookcommand@series{Xendbeforenumber}%
4858   \newhookcommand@series{Xendafternumber}%
4859
4860   \newhooktoggle@series{Xendparagraph}%
4861   \newhookcommand@series{Xendafternote}%
4862   \newhookcommand@series{Xendsep}%
4863
4864   \newhookcommand@series{Xendinplaceofnumber}%
4865   \newhooktoggle@series{Xendnonumber}%
4866
4867   \newhooktoggle@series{Xendnumberonlyfirstinline}%
4868   \newhooktoggle@series{Xendnumberonlyfirstintwolines}%
4869
4870   \newhookcommand@series{Xendsymlinenum}%
4871   \newhookcommand@series{Xendbeforesymlinenum}%
4872   \newhookcommand@series{Xendaftersymlinenum}%
4873   \newhookcommand@series{Xendboxsymlinenum}%
4874
4875   \newhookcommand@series{Xendbhooklinenumber}%
4876   \newhookcommand@series{Xendahooklinenumber}%
4877   \newhookcommand@series{Xendbhookinplaceofnumber}%
4878   \newhookcommand@series{Xendahookinplaceofnumber}%
4879
4880   \newhookcommand@series{Xendhangindent}%
4881

```

```

4882 \fi
4883 \newhooktoggle@series{Xendlineflag}[apprefwithpage,SErefwithpage]
4884 \newhookcommand@series{Xendtwolines}[apprefwithpage,SErefwithpage]
4885 \newhookcommand@series{Xendmoreethantwolines}[apprefwithpage,SErefwithpage]
4886 \newhooktoggle@series{Xendtwolinesbutnotmore}[apprefwithpage,SErefwithpage]
4887 \newhooktoggle@series{Xendtwolinesonlyinsamepage}[apprefwithpage,
4888 SErefwithpage]
4889 \newhookcommand@series{Xendlinerangeseparator}[apprefwithpage,SErefwithpage
4890 ]
4891 \newhookcommand@series{Xendbeforepagenumber}[apprefwithpage,SErefwithpage,
4892 SErefonlypage]
4893 \newhookcommand@series{Xendafterpagenumber}[apprefwithpage,SErefwithpage]
4894 \newhookcommand@series{Xendlineprefixsingle}[apprefwithpage,SErefwithpage]
4895 \newhookcommand@series{Xendlineprefixmore}[apprefwithpage,SErefwithpage]
4896 \newhookcommand@series{Xendsublinesep}[apprefwithpage,SErefwithpage]
4897
4898 %

```

XXI.4 Hooks for a particular footnote

\newhooktoggle@specific \newhooktoggle@specific is a generic command to create boolean hook specific to a note.

```

4897 \newcommand{\newhooktoggle@specific}[1]{%
4898 \newtoggle{#1}%
4899 \define@key[mac]{truefootnoteoption}{#1}[]{\global\settoggle{#1}{true}}%
When enabling footnote option
4900 \define@key[mac]{falsefootnoteoption}{#1}[]{\global\settoggle{#1}{false}
4901 }}
4902 %

```

\newhookarg@specific \newhookarg@specific is a generic command to create argumen hook specific to a note.

```

4903 \newcommand{\newhookarg@specific}[1]{%
4904 \define@key[mac]{truefootnoteoption}{#1}{\global\def\linrangesep@{##1}}%
When enabling footnote option
4905 \define@key[mac]{falsefootnoteoption}{#1}{\global\undef\linrangesep@}%
When
4906 }
4907 %

```

And now, we define some hooks specific to a note.

```

4908 \newhooktoggle@specific{fulllines}%
4909 \newhooktoggle@specific{nonum}
4910 \newhooktoggle@specific{nosep}
4911 \newhookarg@specific{linrangesep}

```



```
4912 %
```

linangesep@ `\linangesep@` is defined by the option `linangesep` of critical notes to change temporarily the line range separator for a specific line. As we have to define it before typesetting the line and undefine it after, we use the family of `xkeyval` package's key.

```
4913 %
```

\nomk@ `\nomk@` toggle is used by `reledpar` to remove the footnote mark in the text when using `\footnotexmk`. Read `reledpar` handbook.

```
4914 \newtoggle{\nomk@}%
```

```
4915 %
```

XXI.5 Alias

\Xnolemmaseparator `\Xnolemmaseparator[⟨series⟩]` is just an alias for `\Xlemmaseparator[⟨series⟩]{}`.

```
4916 \newcommand*{\Xnolemmaseparator}[1][1]{\Xlemmaseparator[#1]{}}
```

```
4917 %
```

XXII Output routine

Now we begin the output routine and associated things.

XXII.0.1 Page number management

\pageno `\pageno` is a page number, starting at 1, and `\advancepageno` increments the number.

\advancepageno

```
4918 \countdef\pageno=0 \pageno=1
```

```
4919 \newcommand*{\advancepageno}{\ifnum\pageno<\z@ \global\advance\pageno\m@ne
4920 \else\global\advance\pageno\@ne\fi}
```

```
4921
```

```
4922 %
```

XXII.0.2 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 \TeX are first, then familiar familiar footnotes and finally the critical footnotes.

```

4923 \newcommand*{\l@ddoxtrafeet}{%
4924   \IfStrEq{familiar-critical}{\@fnpos}
4925     {\do@feetX\Xdo@feet}%
4926     {%
4927       \IfStrEq{critical-familiar}{\@fnpos}%
4928       {\Xdo@feet\do@feetX}%
4929       {\do@feetX\Xdo@feet}%
4930     }%
4931 }%
4932
4933 %

```

\Xdo@feet \Xdo@feet is the code extending \@makecol to cater for the extra critical feet.

```

4934 \newcommand*{\Xdo@feet}{%
4935   \setbox\@outputbox \vbox{%
4936     \unvbox\@outputbox
4937     \@opXfeet}}
4938 %

```

\@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.

```

4939 \newcommand\print@Xnotes[1]{%
4940   \xdef\@currentseries{#1}%
4941   \csuse{#1footstart}{#1}%
4942   \csuse{#1footgroup}{#1}%
4943 }%
4944 %

```

We print all series of notes by looping on them. We check before printing them that they are not voided.

```

4945 \newcommand*{\@opXfeet}{%
4946   \unless\ifnocritical@%
4947     \gdef\firstXseries@{}%
4948     \def\do##1{%
4949       \ifvoid\csuse{##1footins}\else%
4950         \global\skip\csuse{##1footins}=\csuse{Xbeforenotes@##1}%
4951         \global\advance\skip\csuse{##1footins} by\csuse{Xafterrule@##1}%
4952         \print@Xnotes{##1}%
4953       \fi%
4954     }%
4955     \dolistloop{\@series}%
4956   \fi%
4957 }%
4958 %

```

\l@ddodoreinextrafeet \l@ddodoreinextrafeet is the code for catering for the extra footnotes within \@reinserts. We use the same category and ordering as in \l@ddoxtrafeet.

```

4959 \newcommand*{\l@ddodoreinextrafeet}{%
4960   \IfStrEq{familiar-critical}{\@fnpos}
4961   {\@doreinfeetX\X@doreinfeet}%
4962   {%
4963     \IfStrEq{critical-familiar}{\@fnpos}%
4964     {\X@doreinfeet\@doreinfeetX}%
4965     {\@doreinfeetX\X@doreinfeet}%
4966   }%
4967 }
4968
4969 %

```

\X@doreinfeet \X@doreinfeet is the code for catering for the extra critical footnotes within \@reinserts.

```

4970 \newcommand*{\X@doreinfeet}{%
4971   \unless\ifnocritical@%
4972   \def\do##1{%
4973     \ifvoid\csuse{##1footins}\else%
4974     \insert\csuse{##1footins}{\unvbox\csuse{##1footins}}%
4975     \fi}%
4976   \dolistloop{\@series}
4977   \fi%
4978 }
4979
4980 %

```

\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

```

4981 \newcommand\print@notesX[1]{%
4982   \xdef\@currentseries{#1}%
4983   \csuse{footstart#1}{#1}%
4984   \csuse{footgroup#1}{#1}%
4985 }%
4986 %

```

We print all the series of notes by looping on them. We check before printing them that they are not voided.

```

4987 \newcommand*{\do@feetX}{%
4988   \unless\ifnofamiliar@%
4989   \gdef\firstseriesX@{%
4990     \setbox\@outputbox \vbox{%
4991       \unvbox\@outputbox%
4992       \def\do##1{%
4993         \ifvoid\csuse{footins##1}\else%
4994         \global\skip\csuse{footins##1}=\csuse{beforenotesX@##1}%
4995         \global\advance\skip\csuse{footins##1} by\csuse{afterterruleX@##1}%
4996         \print@notesX{##1}%
4997       \fi%
4998     }%

```

```

4999   \dolistloop{\@series}}%
5000   \fi%
5001 }%
5002
5003 \newcommand{\@doreinfeetX}{%
5004   \unless\ifnofamiliar@%
5005     \def\do##1{%
5006       \ifvoid\csuse{footins##1}\else
5007         \insert%
5008           \csuse{footins##1}
5009         {\unvbox\csuse{footins##1}}%
5010       \fi%
5011     }%
5012     \dolistloop{\@series}%
5013   \fi%
5014 }%
5015
5016 %

```

XXII.0.3 Standard output's commands patching

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.)

```

5017 \@ifclassloaded{memoir}{%
5018   %
5019
5020   memoir is loaded so we use memoir's built in hooks.
5021
5022   \g@addto@macro{\m@mdoextrafeet}{\l@ddoxtrafeet}%
5023   \g@addto@macro{\m@mdodoreinextrafeet}{\l@ddodoreinextrafeet}%
5024 }{%
5025   %

```

memoir has not been loaded, so patch `\@makecol` and `\@reinserts`.

```

5023 \@ifpackageloaded{fancyhdr}{%
5024   \patchcmd%
5025     {\latex@makecol}%
5026     {\xdef\@freelist{\@freelist\@midlist}}%
5027     {\xdef\@freelist{\@freelist\@midlist}\l@ddoxtrafeet}%
5028     {}%
5029     {\led@error@fail@patch@makecol}%
5030   }{%
5031     \patchcmd%
5032       {\@makecol}%
5033       {\xdef\@freelist{\@freelist\@midlist}}%
5034       {\xdef\@freelist{\@freelist\@midlist}\l@ddoxtrafeet}%
5035       {}%

```

```

5036     {\led@error@fail@patch@@makecol}%
5037   }%
5038
5039   \patchcmd%
5040     {\@reinserts}%
5041     {\ifvbox}%
5042     {\l@ddodoreinextrafeet\ifvbox}%
5043     {}%
5044     {\led@error@fail@patch@@reinserts}%
5045   }
5046
5047   %

```

It turns out that `\@doclearpage` also needs modifying.

`\if@led@nofoot` We have to check if there are any leftover feet.

```

5048   \newif\if@led@nofoot
5049
5050   %

```

```

5051   \@ifclassloaded{memoir}{%
5052   %

```

If the memoir class is loaded we hook into its modified `\@doclearpage`.

```

\@mem@extranofeet 5053 \g@addto@macro{\@mem@extranofeet}{%%
5054   \def\do#1{%
5055     \unless\ifnocritical@%
5056       \ifvoid\csuse{#1footins}\else\@mem@nofootfalse\fi%
5057     \fi%
5058     \unless\ifnofamiliar@%
5059       \ifvoid\csuse{footins#1}\else\@mem@nofootfalse\fi%
5060     \fi%
5061   }
5062   \dolistloop{\@series}%
5063   }%
5064 }{%
5065   %

```

As memoir is not loaded we have patch `\@doclearpage`.

```

\@led@testifnofoot 5066 \newcommand*{\@led@testifnofoot}{%
\@doclearpage      5067   \@led@nofoottrue%
5068   \ifvoid\footins\else%
5069     \@led@nofootfalse%
5070   \fi%
5071   \def\do##1{%
5072     \unless\ifnocritical@%
5073       \ifvoid\csuse{##1footins}\else%

```

```

5074     \@led@nofootfalse%
5075     \fi%
5076   \fi%
5077   \unless\ifnofamiliar@%
5078     \ifvoid\csuse{footins##1}\else%
5079     \@led@nofootfalse%
5080     \fi%
5081   \fi%
5082 }%
5083 \dolistloop{\@series}%
5084 }%
5085
5086 \pretocmd%
5087 {\@docclearpage}%
5088 {\@led@testifnofoot}%
5089 {}%
5090 {\led@error@fail@patch@@docclearpage}%
5091
5092 \patchcmd%
5093 {\@docclearpage}%
5094 {\ifvoid\footins}%
5095 {\if@led@nofoot}%
5096 {}%
5097 {\led@error@fail@patch@@docclearpage}%
5098
5099 }
5100
5101 %

```

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.

```

5102 \list@create{\labelref@list}
5103 %

```

`\zz@@@` A convenience macro to zero two labeling counters in one go.

```

5104 \newcommand*{\zz@@@}{000|000} % set two counters to zero in one go
5105
5106 %

```

`\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.

```

5107 \newcommand*{\edlabel}[1]{%
5108   \ifl@dpairing\ifautopar%
5109     \strut%
5110     \fi\fi%
5111     \@bsphack%
5112     \ifboolexpr{bool{ledRcol} or bool{ledRcol@}}{%
5113       \ifxnote@%
5114         \protected@write\@auxout{%
5115           {\string\l@dmake@labelsR\space\thepage|\l@dparsedstartline|\l@dparsedstartsub|\the\c@pstartR|{#1}}}%
5116           \ifdef{\hypertarget}%
5117             {\Hy@raisedlink{\hypertarget{#1}{}}}%
5118             {}%
5119         \else%
5120           \write\linenum@out{\string\@lab}%
5121           \ifx\labelref@listR\empty%
5122             \xdef\label@refs{\zz@@@}%
5123           \else%
5124             \gl@p\labelref@listR\to\label@refs%
5125           \fi%
5126           \ifvmode%
5127             \advancelabel@refs%
5128           \fi%
5129   %

```

Use code from the kernel `\label` command to write the correct page number. Also define an `hypertarget` if `hyperref` package is loaded.

```

5130   \protected@write\@auxout{%
5131     {\string\l@dmake@labelsR\space\thepage|\label@refs|\the\c@pstartR|{#1}}}%
5132     \ifdef{\hypertarget}%
5133       {\Hy@raisedlink{\hypertarget{#1}{}}}%

```

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.

```

5134     {}%
5135     \fi%
5136   }{%
5137     \ifXnote%
5138       \protected@write\@auxout{%
5139         {\string\l@dmake@labels\space\thepage|\l@dparsedstartline|\
l@dparsedstartsub|\the\c@pstart|{#1}}%
5140         \ifdef{\hypertarget}%
5141           {\Hy@raisedlink{\hypertarget{#1}}}%
5142         {}%
5143       \else%
5144         \write\linenum@out{\string\@lab}%
5145         \ifx\labelref@list\empty%
5146           \xdef\label@refs{\zz@@@}%
5147         \else%
5148           \gl@p\labelref@list\to\label@refs%
5149         \fi%
5150         \ifvmode%
5151           \advancelabel@refs%
5152         \fi%
5153         \protected@write\@auxout{%
5154           {\string\l@dmake@labels\space\thepage|\label@refs|\the\c@pstart
|{#1}}%
5155           \ifdef{\hypertarget}%
5156             {\Hy@raisedlink{\hypertarget{#1}}}%
5157           {}%
5158         \fi%
5159       }%
5160     \@esphack}%
5161
5162 %

```

`\advancelabel@refs` In cases where `\edlabel` is the first element in a paragraph, we have a problem with line counts, because line counts change only at the first horizontal box of the paragraph. Hence, we need to test `\edlabel` if it occurs at the start of a paragraph. To do so, we use `\ifvmode`. If the test is true, we must advance by one unit the amount of text we write into the .aux file. We do so using `\advancelabel@refs` command.

`\labelrefsparseline`

`\labelrefsparsesubline`

```

5163 \newcounter{line}%
5164 \newcounter{subline}%
5165 \newcommand{\advancelabel@refs}{%
5166   \setcounter{line}{\expandafter\labelrefsparseline\label@refs}%
5167   \stepcounter{line}%
5168   \ifsublines%
5169     \setcounter{subline}{\expandafter\labelrefsparsesubline\label@refs}
%
5170     \stepcounter{subline}{1}%
5171     \def\label@refs{\theline|\thesubline}%
5172   \else%
5173     \def\label@refs{\theline|0}%

```



```

5174 \fi%
5175 }
5176 \def\labelrefsparseline#1|#2{#1}
5177 \def\labelrefsparsesubline#1|#2{#2}
5178 %

```

\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.

```

5179 \newcommand*{\l@dmake@labels}{%
5180 \def\l@dmake@labels#1|#2|#3|#4|#5{%
5181 \expandafter\ifx\csname the@label\csuse{XR@prefix}#5\endcsname \relax\
else
5182 \led@warn@DuplicateLabel{\csuse{XR@prefix}#5}%
5183 \fi
5184 \expandafter\gdef\csname the@label\csuse{XR@prefix}#5\endcsname
{#1|#2|#3|#4|\relax}%
5185 \ignorespaces}
5186 %
5187 %

```

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.

```

5188 \AtBeginDocument{%
5189 \def\l@dmake@labels#1|#2|#3|#4|#5{%
5190 }
5191 %
5192 %

```

\@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.

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`.

```

5193 \newcommand*{\@lab}{%
5194 \ifledRcol
5195 \xright@appenditem{\linenumr@p{\line@numR}}{|%
5196 \ifsublines@ \sublinenumr@p{\subline@numR}\else 0\fi}%
5197

```

```

5198     \to\labelref@listR
5199   \else
5200     \xright@appenditem{\linenumr@p{\line@num}|%
5201       \ifsublines@ \sublinenumr@p{\subline@num}\else 0\fi}%
5202     \to\labelref@list
5203   \fi}
5204 %

```

\apptlabel \apptlabel, if called in \edtext will insert automatically both a start and an end label for the current edtext lines.

```

5205 \newcommand*{\apptlabel}[1]{%
5206   \ifnum\@edtext@level>0%
5207 %

```

Label should not be already defined.

```

5208   \ifcsundef{the@label#1}{%
5209     \csdef{the@label#1}{apptlabel}%
5210   }%
5211   {%
5212     \led@warn@DuplicateLabel{#1 (apptlabel)}%
5213   }%
5214 %

```

Parse the \edtext line numbers.

```

5215   \expandafter\l@dp@rsefootspec\l@d@nums|%
5216 %

```

Use the L^AT_EX standard hack for label.

```

5217   \@bsphack%
5218 %

```

And now, write the data in the auxiliary file.

```

5219   \ifledRcol%
5220     \protected@write\@auxout{%
5221       {\string\l@dmake@labelsR\space\l@dparsedstartpage|\
5222       \l@dparsedstartline|\l@dparsedstartsub|\the\c@pstartR|{#1:start}}%
5223       \ifdef{\hypertarget}%
5224         {\Hy@raisedlink{\hypertarget{#1:start}}}%
5225       }%
5226     \protected@write\@auxout{%
5227       {\string\l@dmake@labelsR\space\l@dparsedendpage|\l@dparsedendline
5228       |\l@dparsedendsub|\the\c@pstartR|{#1:end}}%
5229     }%
5230   \else%
5231     \protected@write\@auxout{%
5232       {\string\l@dmake@labelsR\space\l@dparsedstartpage|\
5233       \l@dparsedstartline|\l@dparsedstartsub|\the\c@pstartR|{#1:start}}%
5234       \ifdef{\hypertarget}%
5235         {\Hy@raisedlink{\hypertarget{#1:start}}}%
5236       }%
5237   }%

```

```

5232     {}%
5233     \protected@write\@auxout{}%
5234     {\string\l@dmake@labels\space\l@dparsedendpage|\l@dparsedendline
|\l@dparsedendsub|\the\c@pstart|{#1:end}}}%
5235     \fi%
5236 %

```

Use the \TeX standard hack for label.

```

5237     \@esphack%
5238 %

```

Warning if `\applabel` is called outside of `\edtext`.

```

5239 \else%
5240 \led@warn@AppLabelOutEdtext{#1}%
5241 \fi%
5242 %

```

End of `\applabel`

```

5243 }%
5244 %

```

`\edlabels` `\edlabels` and `\edlabelE` are just used to mark the beginning and the end of a passage.

```

\edlabelE
\edlabelSE
5245 \newcommand{\edlabels}[1]{%
5246 \edlabel{#1:start}%
5247 }
5248 \newcommand{\edlabelE}[1]{%
5249 \edlabel{#1:end}%
5250 }
5251 \newcommand{\edlabelSE}[1]{%
5252 \edlabels{#1}%
5253 \edlabelE{#1}%
5254 }
5255 %

```

`\wrap@edcrossref` `\wrap@edcrossref` is called around all `reledmac` crossref commands, except those which start with `x`. It adds the hyperlink.

```

5256 \newrobustcmd{\wrap@edcrossref}[2]{%
5257 \ifdef\hyperlink{%
5258 {\hyperlink{#1}{#2}}%
5259 {#2}%
5260 }
5261 %

```

`\edpageref` If the specified label exists, `\edpageref` gives its page number.

`\xpageref` For this reference command, as for the other two, a special version with prefix `x` is provided for use in places where the command is to be scanned as a number, as in

`\linenum`. These special versions have two limitations: they do not print error messages if the reference is unknown, and they can't appear as the first label or reference command in the file; you must ensure that a `\edlabel` or a normal reference command appears first, or these x-commands will always return zeros.

L^AT_EX already defines a `\pageref`, so changing the name to `\edpageref`.

```
5262 \newcommand*\edpageref}[1]{\l@dref@undefined{#1}\wrap@edcrossref{#1}{\l@dgetref@num{1}{#1}}}
5263 \newcommand*\xpageref}[1]{\l@dgetref@num{1}{#1}}
5264
5265 %
```

`\edlineref` If the specified label exists, `\lineref` gives its line number.

```
\xlineref
5266 \newcommand*\edlineref}[1]{\l@dref@undefined{#1}\wrap@edcrossref{#1}{\l@dgetref@num{2}{#1}\xflagref{#1}}}%
5267 \newcommand*\xlineref}[1]{\l@dgetref@num{2}{#1}}%
5268
5269 %
```

`\sublineref` If the specified label exists, `\sublineref` gives its sub-line number.

```
\xsublineref
5270 \newcommand*\sublineref}[1]{\l@dref@undefined{#1}\wrap@edcrossref{#1}{\l@dgetref@num{3}{#1}}}
5271 \newcommand*\xsublineref}[1]{\l@dgetref@num{3}{#1}}
5272
5273 %
```

`\pstartref` If the specified label exists, `\pstartref` gives its pstart number.

```
\xpstartref
5274 \newcommand*\pstartref}[1]{\l@dref@undefined{#1}\wrap@edcrossref{#1}{\l@dgetref@num{4}{#1}}}
5275 \newcommand*\xpstartref}[1]{\l@dgetref@num{4}{#1}}
5276
5277 %
```

`\xflagref` `\xflagref` finds the side flag of any ref defined with `\edlabel`.

```
5278 \newcommand*\xflagref}[1]{\l@dgetref@num{5}{#1}}
5279 %
```

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.

```

5280 \newcommand*{\l@dref@undefined}[1]{%
5281   \expandafter\ifx\csname the@label#1\endcsname\relax
5282     \led@warn@RefUndefined{#1}%
5283   \fi}
5284
5285 %

```

\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`.

```

5286 \newcommand*{\l@dgetref@num}[2]{%
5287   \expandafter
5288   \ifx\csname the@label#2\endcsname \relax
5289     000%
5290   \else
5291     \expandafter\expandafter\expandafter
5292     \l@dlabel@parse\csname the@label#2\endcsname|#1%
5293   \fi}
5294
5295 %

```

\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`.)

```

5296 \newcommand*{\l@dlabel@parse}{%
5297   \def\l@dlabel@parse#1|#2|#3|#4|#5|#6{%
5298     \ifcase #6%
5299       \or #1%
5300       \or #2%
5301       \or #3%
5302       \or #4%
5303       \or #5%
5304     \fi}
5305 %

```

\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.

```

5306 \newcommand*\xxref}[2]{%
5307   {%
5308     \expandafter\ifx\csname the@label#1\endcsname \relax%
5309       \expandafter\let\csname the@@label#1\endcsname\zz@@@%
5310     \else%
5311       \expandafter\def\csname the@@label#1\endcsname{\l@dgetref@num
5312         {1}{#1}|\l@dgetref@num{2}{#1}|\l@dgetref@num{3}{#1}}%
5313       \fi%
5314       \expandafter\ifx\csname the@label#2\endcsname \relax%
5315         \expandafter\let\csname the@@label#2\endcsname\zz@@@%
5316       \else%
5317         \expandafter\def\csname the@@label#2\endcsname{\l@dgetref@num
5318           {1}{#2}|\l@dgetref@num{2}{#2}|\l@dgetref@num{3}{#2}}%
5319       \fi%
5320       \letcs{\@tempa}{the@@label#1}%
5321       \letcs{\@tempb}{the@@label#2}%
5322       \linenum{\@tempa|
5323         \@tempb}}}%

```

`\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.

So, first declare the default value of the hooks for the pseudo-series. Also declare the internal toggle which are switch by `reledmac`.

```

5324 \def\Xtwolines@appref{}%
5325 \def\Xtwolines@SEref{}%
5326
5327 \def\Xmorethantwolines@appref{}%
5328 \def\Xmorethantwolines@SEref{}%
5329
5330 \def\Xlinerangeseparator@appref{\endashchar}%
5331 \def\Xlinerangeseparator@SEref{\endashchar}%
5332
5333 \def\Xsublinesep@appref{\fullstop}%
5334 \def\Xsublinesep@SEref{\fullstop}%
5335
5336 \newtoggle{Xtwolinesbutnotmore@appref}%
5337 \newtoggle{Xtwolinesbutnotmore@SEref}%
5338
5339 \newtoggle{Xtwolinesonlyinsamepage@appref}%
5340

```

```

5341 \newtoggle{Xtwolinesonlyinsamepage@SEref}%
5342
5343 \newtoggle{Xlineflag@appref}%
5344 \toggletrue{Xlineflag@appref}%Here exception
5345 \newtoggle{Xlineflag@SEref}%
5346 \toggletrue{Xlineflag@SEref}%Here exception
5347
5348 \def\Xendtwolines@apprefwithpage{}%
5349 \def\Xendtwolines@SErefwithpage{}%
5350
5351 \def\Xendmoreethantwolines@apprefwithpage{}%
5352 \def\Xendmoreethantwolines@SErefwithpage{}%
5353
5354 \def\Xendlinerangeseparator@apprefwithpage{\endashchar}
5355 \def\Xendlinerangeseparator@SErefwithpage{\endashchar}
5356 \def\Xendlinerangeseparator@SErefonlypage{\endashchar}
5357
5358 \def\Xendbeforepagenumber@apprefwithpage{p.}%
5359 \def\Xendbeforepagenumber@SErefwithpage{p.}%
5360 \def\Xendbeforepagenumber@SEonlypage{p.}%
5361
5362 \def\Xendafterpagenumber@apprefwithpage{}%
5363 \def\Xendafterpagenumber@SErefwithpage{}%
5364
5365
5366 \def\Xendlineprefixsingle@apprefwithpage{}%
5367 \def\Xendlineprefixsingle@SErefwithpage{}%
5368
5369 \def\Xendlineprefixmore@apprefwithpage{}%
5370 \def\Xendlineprefixmore@SErefwithpage{}%
5371
5372 \newtoggle{Xendtwolinesbutnotmore@apprefwithpage}%
5373 \newtoggle{Xendtwolinesbutnotmore@SErefwithpage}%
5374
5375 \def\Xendsublinesep@apprefwithpage{\fullstop}%
5376 \def\Xendsublinesep@SErefwithpage{\fullstop}%
5377
5378 \newtoggle{Xendtwolinesonlyinsamepage@apprefwithpage}%
5379 \newtoggle{Xendtwolinesonlyinsamepage@SErefwithpage}%
5380
5381 \newtoggle{Xendlineflag@apprefwithpage}
5382 \toggletrue{Xendlineflag@apprefwithpage}%Here, exception
5383 \newtoggle{Xendlineflag@SErefwithpage}
5384 \toggletrue{Xendlineflag@SErefwithpage}%Here, exception
5385
5386 %

```

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.

```

5387 \xdef\Xboxstartlinenum@appref{Opt}
5388 \xdef\Xboxstartlinenum@SEref{Opt}
5390
5391 \xdef\Xboxendlinenum@appref{Opt}
5392 \xdef\Xboxendlinenum@SEref{Opt}
5393
5394 \xdef\Xendboxstartlinenum@apprefwithpage{Opt}
5395 \xdef\Xendboxstartlinenum@SErefwithpage{Opt}
5396
5397 \xdef\Xendboxendlinenum@apprefwithpage{Opt}
5398 \xdef\Xendboxendlinenum@SErefwithpage{Opt}
5399
5400 %

```

Now, declare the default values of \@apprefprefixsingle and \@apprefprefixmore, \@SErefprefix, \@SErefprefixmore and the commands which defines them.

```

5401 \newcommand\@apprefprefixsingle{}%
5402 \newcommand\@SErefprefixsingle{}%
5403
5404 \newcommand\@apprefprefixmore{}%
5405 \newcommand\@SErefprefixmore{}%
5406
5407 \newcommand{\setapprefprefixsingle}[1]{%
5408   \gdef\@apprefprefixsingle{#1}%
5409 }
5410 \newcommand{\setSErefprefixsingle}[1]{%
5411   \gdef\@SErefprefixsingle{#1}%
5412 }
5413
5414 \newcommand{\setapprefprefixmore}[1]{%
5415   \gdef\@apprefprefixmore{#1}%
5416 }
5417 \newcommand{\setSErefprefixmore}[1]{%
5418   \gdef\@SErefprefixmore{#1}%
5419 }
5420
5421 %

```

And not \setSErefonlypageprefixsingle and \setSErefonlypageprefixmore.

```

5422 \let\setSErefonlypageprefixsingle\XendbeforepagenumberSErefonlypage%
5423 \newcommand{\setSErefonlypageprefixmore}[1]{%
5424   \gdef\SErefonlypage@prefixmore{#1}%
5425 }%
5426 %

```

And now, the main commands: \appref, \apprefwithpage, \SEref and \SErefwithpage. These commands call \refformated@ and \refformatedwithpage, which calls \printlines and \printendlines. That is why we have previously declared all hooks values tested inside these last commands.


```

5427 \newcommandx{\appref}[2][1,usedefault]{\reformatted@{#1}{#2}{appref}}
5428 \newcommandx{\Seref}[2][1,usedefault]{\reformatted@{#1}{#2}{Seref}}
5429
5430
5431 \newcommandx{\apprefwithpage}[2][1,usedefault]{\reformattedwithpage@
5432 {#1}{#2}{appref}}
5433 \newcommandx{\Serefwithpage}[2][1,usedefault]{\reformattedwithpage@
5434 {#1}{#2}{Seref}}
5435 \newcommandx{\Serefonlypage}[2][1,usedefault]{\reformattedonlypage@
5436 {#1}{#2}{Seref}}
5437
5438 \newcommand{\reformatted@}[3]{%
5439   \def\do##1{%
5440     \setkeys[mac]{truefootnoteoption}{##1}%
5441     }%
5442     \notblank{#1}{\docsvlist{#1}}{ }%
5443     \xdef\@currentseries{#3}%
5444     \ifcempty{@#3prefixmore}%
5445       {\@apprefprefixsingle}%
5446       {%
5447         \IfEq{\xlineref{#2:start}}{\xlineref{#2:end}}%
5448         {\csuse{@#3prefixsingle}}%
5449         {\csuse{@#3prefixmore}}%
5450       }%
5451     \ifboolexpr{%
5452       test{\ifcsundef{the@label#2:start}}%
5453       or test{\ifcsundef{the@label#2:end}}%
5454     }%
5455     {\led@warn@pairRefUndefined{#2}\nfss@text{\reset@font\bfseries ??}}%
5456     {%
5457       \def\@this@crossref@start{#2:start}%
5458       \def\@this@crossref@end{#2:end}%
5459       \printlines\xpageref{#2:start}|\xlineref{#2:start}|\xsublineref{#2:
5460 start}|\xpageref{#2:end}|\xlineref{#2:end}|\xsublineref{#2:end}|\relax|\
5461 xflagref{#2:start}|}%
5462       \undef\@this@crossref@end%
5463       \undef\@this@crossref@start%
5464     }%
5465     \def\do##1{%
5466       \setkeys[mac]{falsefootnoteoption}{##1}%
5467       }%
5468       \notblank{#1}{\docsvlist{#1}}{ }%
5469     }%
5470   }%
5471 \newcommand{\reformattedwithpage@}[3]{%
5472   \def\do##1{%
5473     \setkeys[mac]{truefootnoteoption}{##1}%
5474     }%
5475     \notblank{#1}{\docsvlist{#1}}{ }%

```

```

5472 \xdef\@currentseries{#3withpage}%
5473 \ifboolexpr{%
5474   test{\ifcsundef{the@label#2:start}}%
5475   or test{\ifcsundef{the@label#2:end}}%
5476 }%
5477 {\led@warn@pairRefUndefined{#2}\nfss@text{\reset@font\bfseries ??}}%
5478 {%
5479 \def\@this@crossref@start{#2:start}%
5480 \def\@this@crossref@end{#2:end}%
5481 \printendlines\xpageref{#2:start}|\xlineref{#2:start}|\xsublineref{#2:
start}|\xpageref{#2:end}|\xlineref{#2:end}|\xsublineref{#2:end}|\relax|\
xflagref{#2:start}}%
5482 \undef\@this@crossref@end%
5483 \undef\@this@crossref@start%
5484 }%
5485 \def\do##1{%
5486 \setkeys[mac]{falsefootnoteoption}{##1}%
5487 }%
5488 \notblank{#1}{\docsvlist{#1}}}%
5489 }%
5490
5491 \newcommand{\reformattedonlypage@}[3]{%
5492 \def\do##1{%
5493 \setkeys[mac]{truefootnoteoption}{##1}%
5494 }%
5495 \notblank{#1}{\docsvlist{#1}}}%
5496 \xdef\@currentseries{#3onlypage}%
5497 \ifboolexpr{%
5498   test{\ifcsundef{the@label#2:start}}%
5499   or test{\ifcsundef{the@label#2:end}}%
5500 }%
5501 {\led@warn@pairRefUndefined{#2}\nfss@text{\reset@font\bfseries ??}}%
5502 {\ifnumequal{\xpageref{#2:end}}{\xpageref{#2:start}}%
5503   {%
5504     \printnpnum{%
5505       \wrap@edcrossref{#2:start}{\xpageref{#2:start}}%
5506     }%
5507   }%
5508 }%
5509 \ifcsvoid{#3onlypage@prefixmore}%
5510 {}%
5511 {\csletcs{Xendbeforepagenumber@#3onlypage}{#3onlypage@prefixmore}}%
5512 \ifdefined\linerrangesep%
5513 \printnpnum{%
5514   \wrap@edcrossref{#2:start}{\xpageref{#2:start}}%
5515   \linerrangesep%
5516   \wrap@edcrossref{#2:end}{\xpageref{#2:end}}%
5517 }%
5518 \else%
5519 \printnpnum{%

```

```

5520 \wrap@edcrossref{#2:start}{\xpageref{#2:start}}}%
5521 \csuse{Xendlinrangeseparator@{\@currentseries}}%
5522 \wrap@edcrossref{#2:end}{\xpageref{#2:end}}}%
5523 }%
5524 \fi%
5525 }%
5526 }%
5527 \def\do##1{%
5528 \setkeys[mac]{falsefootnoteoption}{##1}%
5529 }%
5530 \notblank{#1}{\docsvlist{#1}}{ }%
5531 }%
5532 %

```

\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`.

```

5533 \newcommand*{\edmakelabel}[2]{\expandafter\xdef\csname the@label#1\
endcsname{#2}}
5534
5535 %

```

(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. 116 and V.9 p. 87), 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`.

```

5536 \pretocmd{\XR@test}%
5537 {\XR@test@mac+++#1#2#3#4+++}%
5538 {}%
5539 {}%
5540 %

```

\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`.

```

5541 \long\def\XR@test@mac+++#1+++{\XR@test@mac@test#1}
5542 %

```

`\XR@test@mac@test` And finally, `\XR@test@mac@test` does the job. This code is based on the `\XR@test` macro of the `xr` package. However, not that the `\XR@prefix` is not called here, but it is integrated directly in `\l@dmake@labels` and `\l@dmake@labelsR`.

```

5543 \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
5544 \ifx#1\l@dmake@labels%
5545 \l@dmake@labels#2%
5546 \else
5547 \ifx#1\l@dmake@labelsR%
5548 \l@dmake@labelsR #2%
5549 \fi%
5550 \fi%
5551 }%
5552 %

```

XXIV Side notes

Regular `\marginpars` do not work inside numbered text — they do not produce any note but do put an extra unnumbered blank line into the text.

`\@xympar` Changing `\@xympar` a little at least ensures that `\marginpars` in numbered text do not disturb the flow.

```

5553 \pretocmd{\@xympar}%
5554 {\ifnumberedpar@
5555 \led@warn@NoMarginpars
5556 \@esphack
5557 \else}%
5558 {}%
5559 {}%
5560
5561 \apptocmd{\@xympar}%
5562 {\fi}%
5563 {}
5564 {}
5565
5566 %

```

We provide side notes as replacement for `\marginpar` in numbered text.

`\sidenote@margin` These are the sidenote equivalents to `\line@margin` and `\linenummargin` for specifying which margin. The default is the right margin (opposite to the default for line numbers). `\l@dgetsidenote@margin` returns the number associated to side note margin:

left : 0

right : 1

outer : 2

inner : 3

```

5567 \newcount\sidenote@margin
5568 \newcommand*{\sidenotemargin}[1]{\{
5569   \l@dgetsidenote@margin{#1}%
5570   \ifnum\@l@tempcntb>\m@ne
5571     \ifledRcol
5572       \global\sidenote@marginR=\@l@tempcntb
5573     \else
5574       \global\sidenote@margin=\@l@tempcntb
5575     \fi
5576   \fi}}
5577 \newcommand*{\l@dgetsidenote@margin}[1]{%
5578   \def\@tempa{#1}\def\@tempb{left}%
5579   \ifx\@tempa\@tempb
5580     \@l@tempcntb \z@
5581   \else
5582     \def\@tempb{right}%
5583     \ifx\@tempa\@tempb
5584       \@l@tempcntb \@ne
5585     \else
5586       \def\@tempb{outer}%
5587       \ifx\@tempa\@tempb
5588         \@l@tempcntb \tw@
5589       \else
5590         \def\@tempb{inner}%
5591         \ifx\@tempa\@tempb
5592           \@l@tempcntb \thr@@
5593         \else
5594           \led@warn@BadSidenotemargin
5595           \@l@tempcntb \m@ne
5596         \fi
5597       \fi
5598     \fi
5599   \fi}
5600 \sidenotemargin{right}
5601
5602 %

```

\l@dlp@rbox We need two boxes to store sidenote texts.

\l@drp@rbox

```

5603 \newbox\l@dlp@rbox
5604 \newbox\l@drp@rbox
5605
5606 %

```

`\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`
`\ledlsnotefontsetup`
`\ledrsnotefontsetup`

```

5607 \newdimen\ledlsnotewidth \ledlsnotewidth=\marginparwidth
5608 \newdimen\ledrsnotewidth \ledrsnotewidth=\marginparwidth
5609 \newdimen\ledlsnotesep \ledlsnotesep=\linenumsep
5610 \newdimen\ledrsnotesep \ledrsnotesep=\linenumsep
5611 \newcommand*\ledlsnotefontsetup{\raggedleft\footnotesize}
5612 \newcommand*\ledrsnotefontsetup{\raggedright\footnotesize}
5613
5614 %

```

`\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`

```

5615 \newcommand*\ledleftnote[1]{\edtext{}\l@dlsnote{#1}}
5616 \newcommand*\ledrightnote[1]{\edtext{}\l@drsnote{#1}}
5617
5618 \newcommand*\ledinnernote[1]{%
5619   \ifodd\c@page% Do not use \page@num, because it is not yet calculated
when command is called
5620   \ledleftnote{#1}%
5621   \else%
5622   \ledrightnote{#1}%
5623   \fi%
5624 }
5625
5626 \newcommand*\ledouternote[1]{%
5627   \ifodd\c@page% Do not use \page@num, because it is not yet calculated
when command is called
5628   \ledrightnote{#1}%
5629   \else%
5630   \ledleftnote{#1}%
5631   \fi%
5632 }
5633
5634 \newcommand*\ledsidenote[1]{\edtext{}\l@dcsnote{#1}}
5635 %

```

`\l@dlsnote` . The ‘footnotes’ for left, right, and moveable sidenotes. The whole scheme is reminiscent
`\l@drsnote` of the critical footnotes code.
`\l@dcsnote`

```

5636 \newif\ifrighnoteup
5637 \righnoteuptrue
5638
5639 \newcommand*\l@dlsnote[1]{%
5640   \begingroup%
5641   \newcommand{\content}{#1}%

```

```

5642 \ifnumberedpar@
5643 \ifledRcol%
5644 \xright@appenditem{\noexpand\vl@dlsnote{\expandonce\content}}}%
5645 \to\inserts@listR
5646 \global\advance\insert@countR \@ne%
5647 \else%
5648 \xright@appenditem{\noexpand\vl@dlsnote{\expandonce\content}}}%
5649 \to\inserts@list
5650 \global\advance\insert@count \@ne%
5651 \fi
5652 \fi\ignorespaces\endgroup}
5653
5654 \newcommand*{\l@drsnote}[1]{%
5655 \begingroup%
5656 \newcommand{\content}{#1}%
5657 \ifnumberedpar@
5658 \ifledRcol%
5659 \xright@appenditem{\noexpand\vl@drsnote{\expandonce\content}}}%
5660 \to\inserts@listR
5661 \global\advance\insert@countR \@ne%
5662 \else%
5663 \xright@appenditem{\noexpand\vl@drsnote{\expandonce\content}}}%
5664 \to\inserts@list
5665 \global\advance\insert@count \@ne%
5666 \fi
5667 \fi\ignorespaces\endgroup}
5668
5669 \newcommand*{\l@dcsnote}[1]{%
5670 \begingroup%
5671 \newcommand{\content}{#1}%
5672 \ifnumberedpar@
5673 \ifledRcol%
5674 \xright@appenditem{\noexpand\vl@dcsnote{\expandonce\content}}}%
5675 \to\inserts@listR
5676 \global\advance\insert@countR \@ne%
5677 \else%
5678 \xright@appenditem{\noexpand\vl@dcsnote{\expandonce\content}}}%
5679 \to\inserts@list
5680 \global\advance\insert@count \@ne%
5681 \fi
5682 \fi\ignorespaces\endgroup}
5683
5684 %

```

`\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@dcscnotetext in any cases.
- Store the content of \rightsidenote to:
 - \l@dcscnotetext if \ledsidenote is to be put on right.
 - \l@dcscnotetext@r if \ledsidenote is to be put on left.
- Store the content of \leftsidenote to:
 - \l@dcscnotetext if \ledsidenote is to be put on left.
 - \l@dcscnotetext@l if \ledsidenote is to be put on right.

```

5685 \newcommand*{\vl@dlsnote}[1]{%
5686   \ifledRcol@%
5687     \@l@dttempcntb=\sidenote@marginR%
5688     \ifnum\@l@dttempcntb>\@ne%
5689       \advance\@l@dttempcntb by\page@numR%
5690     \fi%
5691   \else%
5692     \@l@dttempcntb=\sidenote@margin%
5693     \ifnum\@l@dttempcntb>\@ne%
5694       \advance\@l@dttempcntb by\page@num%
5695     \fi%
5696   \fi%
5697   \ifodd\@l@dttempcntb%
5698     \listgadd{\l@dcscnotetext@l}{#1}%
5699   \else%
5700     \listgadd{\l@dcscnotetext}{#1}%
5701   \fi
5702 }
5703 \newcommand*{\vl@drsnote}[1]{%
5704   \ifledRcol@%
5705     \@l@dttempcntb=\sidenote@marginR%
5706     \ifnum\@l@dttempcntb>\@ne%
5707       \advance\@l@dttempcntb by\page@numR%
5708     \fi%
5709   \else%
5710     \@l@dttempcntb=\sidenote@margin%
5711     \ifnum\@l@dttempcntb>\@ne%
5712       \advance\@l@dttempcntb by\page@num%
5713     \fi%
5714   \fi%
5715   \ifodd\@l@dttempcntb%
5716     \listgadd{\l@dcscnotetext}{#1}%
5717   \else%
5718     \listgadd{\l@dcscnotetext@r}{#1}%
5719   \fi%
5720 }
5721 \newcommand*{\vl@dcscnote}[1]{\listgadd{\l@dcscnotetext}{#1}}
5722

```


5723 %

`\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.

```
5724 \newcommand*{\setl@dlp@rbox}[1]{%
5725   {\parindent\z@\hsize=\ledlsnotewidth\ledlsnotefontsetup
5726     \global\setbox\l@dlp@rbox
5727     \ifleftnoteup
5728       =\vbox to\z@{\vss #1}%
5729     \else
5730       =\vbox to 0.70\baselineskip{\strut#1\vss}%
5731     \fi}}
5732 \newcommand*{\setl@drp@rbox}[1]{%
5733   {\parindent\z@\hsize=\ledrsnotewidth\ledrsnotefontsetup
5734     \global\setbox\l@drp@rbox
5735     \ifrightnoteup
5736       =\vbox to\z@{\vss#1}%
5737     \else
5738       =\vbox to0.7\baselineskip{\strut#1\vss}%
5739     \fi}}
5740 \newif\ifleftnoteup
5741 \leftnoteuptrue
5742 %
```

`\@sidenotesep` This macro is used to separate sidenotes of the same line.

```
5743 \newcommand{\setsidenotesep}[1]{\gdef\@sidenotesep{#1}}
5744 \newcommand{\@sidenotesep}{, }
5745 %
```

`\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.

```
5746 \newcommand*{\affixside@note}{%
5747   \def\sidenotecontent@{}%
5748   \numgdef{\itemcount@}{0}%
5749   \def\do##1{%
5750     \ifnumequal{\itemcount@}{0}%
5751       {%
5752         \appto\sidenotecontent@{##1}}% Not print not separator before
5753         the 1st note
5754       {\appto\sidenotecontent@{\@sidenotesep ##1}%
5755       }%
5756   \numgdef{\itemcount@}{\itemcount@+1}%
5757 }
```

```

5756 }%
5757 \dolistloop{\l@dcnotetext}%
5758 \ifnumgreater{\itemcount@}{1}{\led@err@ManySidenotes}{}%
5759 %

```

And we do the same for left and right notes (not movable).

```

5760 \gdef\@templ@d{%
5761 \gdef\@templ@n{\l@dcnotetext\l@dcnotetext@l\l@dcnotetext@r}%
5762 \ifx\@templ@d\@templ@n \else%
5763 \if@twocolumn%
5764 \if@firstcolumn%
5765 \setl@dlp@rbox{##1}{\sidenotecontent@}%
5766 \else%
5767 \setl@drp@rbox{\sidenotecontent@}%
5768 \fi%
5769 \else%
5770 \l@dtempcntb=\sidenote@margin%
5771 \ifnum\l@dtempcntb>\@ne%
5772 \advance\l@dtempcntb by\page@num%
5773 \fi%
5774 \ifodd\l@dtempcntb%
5775 \setl@drp@rbox{\sidenotecontent@}%
5776 \gdef\sidenotecontent@{%
5777 \numgdef{\itemcount@}{0}%
5778 \dolistloop{\l@dcnotetext@l}%
5779 \ifnumgreater{\itemcount@}{1}{\led@err@ManyLeftnotes}{}%
5780 \setl@dlp@rbox{\sidenotecontent@}%
5781 \else%
5782 \setl@dlp@rbox{\sidenotecontent@}%
5783 \gdef\sidenotecontent@{%
5784 \numgdef{\itemcount@}{0}%
5785 \dolistloop{\l@dcnotetext@r}%
5786 \ifnumgreater{\itemcount@}{1}{\led@err@ManyRightnotes}{}%
5787 \setl@drp@rbox{\sidenotecontent@}%
5788 \fi%
5789 \fi%
5790 \fi%
5791 }
5792 %

```

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@dfeetbeginmini
\l@dfeetendmini

```

These will be the hooks in `\@iiminipage` and `\endminipage`.

They can be extended to handle other things if necessary.

```

5793 \ifnoledgroup@else%
5794 \newcommand*{\l@dfetbeginmini}{\@ledgrouptrue\l@dedbeginmini\
l@dfambeginmini}
5795 \newcommand*{\l@dfetendmini}{%
5796   \IfStrEq{critical-familiar}{\@mpfnpos}%
5797   {\l@dedendmini\l@dfamendmini}%
5798   {%
5799     \IfStrEq{familiar-critical}{\@mpfnpos}%
5800     {\l@dfamendmini\l@dedendmini}%
5801     {\l@dedendmini\l@dfamendmini}%
5802   }%
5803 }%
5804 %

```

`\l@dedbeginmini` These handle the initiation and closure of critical footnotes in a minipage environment.

```

\l@dedendmini
5805 \newcommand*{\l@dedbeginmini}{%
5806   \unless\ifnocritical@%
5807   \def\do##1{\csletcs{v##1footnote}{mpv##1footnote}}%
5808   \dolistloop{\@series}%
5809   \fi%
5810 }
5811 \newcommand*{\l@dedendmini}{%
5812   \unless\ifnocritical@%
5813   \ifl@dpairing%
5814     \ifledRcol%
5815       \flush@notesR%
5816     \else%
5817       \flush@notes%
5818     \fi%
5819   \fi
5820   \def\do##1{%
5821     \ifvoid\csuse{mp##1footins}\else%
5822     \ifl@dpairing\ifparledgroup%
5823       \ifledRcol%
5824         \dingdef{\parledgroup@beforenotesR}{\parledgroup@beforenotesR+
skip\@nameuse{mp##1footins}}%
5825       \else%
5826         \dingdef{\parledgroup@beforenotesL}{\parledgroup@beforenotesL
+\skip\@nameuse{mp##1footins}}%
5827       \fi%
5828       \fi\fi%
5829       \csuse{mp##1footgroup}{##1}%
5830     \fi}%
5831   \dolistloop{\@series}%
5832   \fi%
5833 }%
5834 %
5835 %

```

`\l@dfambeginmini` These handle the initiation and closure of familiar footnotes in a minipage environment.

`\l@dfamendmini`

```

5836 \newcommand*\l@dfambeginmini}{%
5837   \unless\ifnofamiliar{%
5838     \def\do##1{\csletcs{vfootnote##1}{mpvfootnote##1}}%
5839     \dolistloop{\@series}%
5840     \fi%
5841   }%
5842
5843 \newcommand*\l@dfamendmini}{%
5844   \unless\ifnofamiliar{%
5845     \def\do##1{%
5846       \ifvoid\csuse{mpfootins##1}\else%
5847       \csuse{mpfootgroup##1}{##1}%
5848       \fi}%
5849     \dolistloop{\@series}%
5850     \fi%
5851   }%
5852   %

```

`\@iiiminipage` This is our extended form of the kernel `\@iiiminipage` defined in `ltboxes.dtx`.

```

5853 \patchcmd%
5854   {\@iiiminipage}%
5855   {\let\@footnotetext\@mpfootnotetext}%
5856   {\let\@footnotetext\@mpfootnotetext\l@dfeetbeginmini}%
5857   {}%
5858   {\led@error@fail@patch@\@iiiminipage}%
5859   %

```

`\endminipage` This is our extended form of the kernel `\endminipage` defined in `ltboxes.dtx`.

```

5860 \patchcmd%
5861   {\endminipage}%
5862   {\footnoterule}%
5863   {\footnoterule\l@advance@parledgroup@beforenormalnotes}%
5864   {}%
5865   {\led@error@fail@patch@endminipage}%
5866
5867 \patchcmd%
5868   {\endminipage}%
5869   {\@minipagefalse}%
5870   {\l@dfeetendmini\@minipagefalse}%
5871   {}%
5872   {\led@error@fail@patch@endminipage}%
5873
5874   %

```

`\l@dunboxmpfoot`

`\l@dunboxmpfoot` insert normal footnotes for `ledgroup`.

`\l@advance@parledgroup@beforenormalnotes`

```

5875 \newcommand*{\l@dunboxmpfoot}{%
5876     \vskip\skip\@mpfootins
5877     \normalcolor
5878     \footnoterule
5879     \l@advance@parledgroup@beforenormalnotes
5880     \unvbox\@mpfootins%
5881 }
5882 %

```

When using parallel ledgroup, we need to store the vertical space added before footnote, in order to compensate them between left and right pages.

```

5883 \newcommand{\l@advance@parledgroup@beforenormalnotes}{%
5884     \ifparledgroup
5885         \ifl@dpairing
5886             \ifledRcol
5887                 \dimdef{\parledgroup@beforenotesR}{\parledgroup@beforenotesR+
skip\@mpfootins}
5888             \else
5889                 \dimdef{\parledgroup@beforenotesL}{\parledgroup@beforenotesL+
skip\@mpfootins}
5890             \fi
5891         \fi
5892     \fi
5893 }
5894 %

```

`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.

```

5895 \newenvironment{ledgroup}{%
5896     \resetprevpage@num%
5897     \def\@mpfn{mpfootnote}\def\thempfn{\thempfootnote}\c@mpfootnote\z@%
5898     \let\@footnotetext\@mpfootnotetext
5899     \l@dfeetbeginmini%
5900 }{%
5901     \par
5902     \unskip
5903     \ifvoid\@mpfootins\else
5904         \l@dunboxmpfoot
5905     \fi
5906     \l@dfeetendmini%
5907     \@ledgroupfalse%
5908 }
5909
5910
5911
5912 %

```

`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.

```
5913 \newenvironment{ledgroupsize}[2][1]{%
5914 %
```

Set the various text measures.

```
5915 \hspace #2\relax
5916 %
```

Initialize fills for centering.

```
5917 \let\ledllfill\hfil
5918 \let\ledrlfill\hfil
5919 \def\@tempa{#1}\def\@tempb{1}%
5920 %
```

Left adjusted numbered lines

```
5921 \ifx\@tempa\@tempb
5922 \let\ledllfill\relax
5923 \else
5924 \def\@tempb{r}%
5925 \ifx\@tempa\@tempb
5926 %
```

Right adjusted numbered lines

```
5927 \let\ledrlfill\relax
5928 \fi
5929 \fi
5930 %
```

Set up the footnoting.

```
5931 \def\@mpfn{mpfootnote}\def\thempfn{\thempfootnote}\c@mpfootnote\z@
5932 \let\@footnotetext\@mpfootnotetext
5933 \l@dfetbeginmini%
5934 }{%
5935 \par
5936 \unskip
5937 \ifvoid\@mpfootins\else
5938 \l@dunboxmpfoot
5939 \fi
5940 \l@dfetendmini%
5941 }
5942
5943 %
```

Close the `\ifnoledgroup@` else.

```
5944 \fi%
5945 %
```

`\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`.

```
5946 \newif\ifledgroupnotesL@
5947 \newif\ifledgroupnotesR@
5948 %
```

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`.

```
5949 \AtBeginDocument{%
5950   \unless\ifl@imakeidx%
5951     \ifpackageloaded{imakeidx}{\led@error@ImakeidxAfterEledmac}{}%
5952   \fi%
5953   \unless\ifl@indextools%
5954     \ifpackageloaded{indextools}{\led@error@indextoolsAfterEledmac}{}%
5955   \fi%
5956 }
5957 %
```

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
5958 \newcommand{\pagelinesep}{-}
5959 \newcommand{\edindexlab}{${&}}
5960 \newcounter{labidx}
5961 \setcounter{labidx}{0}
5962
5963 %
```

`\doedindexlabel` This macro sets an `\edlabel`.

```
5964 \newcommand{\doedindexlabel}{%
5965   \stepcounter{labidx}%
5966   \edlabel{\edindexlab\thelabidx}%
5967 }
5968
5969 %
```

`\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`.

```

5970 \newcounter{pageline}%
5971 \renewcommand{\thepageline}{%
5972   \thepage%
5973   \pagelinesep%
5974   \xlineref{\edindexlab\thelabidx}%
5975 }
5976 %

```

\thestartpageline These macros make up the page/line start/end number when the `\edindex` command is called in critical notes.

\theendpageline

```

5977 \newcommand{\thestartpageline}{%
5978   \l@dparsedstartpage%
5979   \pagelinesep%
5980   \l@dparsedstartline%
5981 }
5982 \newcommand{\theendpageline}{%
5983   \l@dparsedendpage%
5984   \pagelinesep%
5985   \l@dparsedendline%
5986 }
5987 %

```

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.

```

5988 \newif\if@edindex@fornote@
5989 %

```

\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 `|`.

```

5990 \newcommand{\prepare@edindex@fornote}[1]{%
5991   \l@dp@rsefootspec#1|}%
5992   \@edindex@fornote@true%
5993 }
5994 %

```

\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.

```

5995 \newcommand{\get@edindex@ledinnote@command}{%
5996   \ifxindy%

```



```

5997 \gdef\@ledinnote@command{%
5998     ledinnote\thelabidx%
5999 }%
6000 \ifxindyhyperref{%
6001     \immediate\write\eledmac@xindy@out{%
6002         (define-attributes ("ledinnote\thelabidx"))^^J
6003         \space\space(markup-locref^^J
6004         \eledmacmarkuplocrefdepth^^J
6005         :open "\string\ledinnote[\edindexlab\thelabidx]{\@index@command
}{"^^J
6006         :close "}"^^J
6007         :attr "ledinnote\thelabidx"^^J
6008         )
6009     }%
6010 \else%
6011     \immediate\write\eledmac@xindy@out{%
6012         (define-attributes ("ledinnote\thelabidx"))^^J
6013         \space\space(markup-locref^^J
6014         \eledmacmarkuplocrefdepth^^J
6015         :open "\string\ledinnote{\@index@command}{"}^^J
6016         :close "}"^^J
6017         :attr "ledinnote\thelabidx"^^J
6018         )
6019     }%
6020 \fi%
6021 %

```

If we do not use xindy option, \@ledinnote@command will produce something like ledinnote{formattingcommand}.

```

6022 \else%
6023 \gdef\@ledinnote@command{%
6024     ledinnote[\edindexlab\thelabidx]{\@index@command}%
6025 }%
6026 \fi%
6027 }
6028 %

```

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 |.

```

6029 \def\get@index@command#1|#2+{%
6030 \gdef\@index@txt{#1}%
6031 \gdef\@index@command{#2}%
6032 \xdef\@index@parenthesis{}%
6033 \IfBeginWith{\@index@command}{({}%
6034 \StrGobbleLeft{\@index@command}{1}{\@index@command}%
6035 \global\let\@index@command\@index@command%
6036 \xdef\@index@parenthesis{({}%

```

```

6037 }{}%
6038 \IfBeginWith{\@index@command}{})}{%
6039   \StrGobbleLeft{\@index@command}{1}[\@index@command]%
6040   \global\let\@index@command\@index@command%
6041   \xdef\@index@parenthesis{)}%
6042 }{}%
6043 }
6044 %

```

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.

`\ledinnotehyperpage`

`\ledinnotemark`

```

6045 \newcommand{\ledinnote}[3][1,usedefault]{%
6046   \ifboolexpr{%
6047     test{\ifdefequal{\iftrue}{\ifHy@hyperindex}}%
6048     or%
6049     bool {xindyhyperref@}%
6050   }%
6051   {%
6052     \csuse{#2}{\hyperlink{#1}{\ledinnotemark{#3}}}%
6053   }%
6054   {%
6055     \csuse{#2}{\ledinnotemark{#3}}%
6056   }%
6057 }%
6058 \newcommand{\ledinnotehyperpage}[2]{\csuse{#1}{\ledinnotemark{\hyperpage
6059   {#2}}}}%
6059 \newcommand{\ledinnotemark}[1]{#1\emph{n}}%
6060 %

```

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`

```

6061 \newcommand{\@wredindex}[2][1=\expandonce\jobname,usedefault]{%#1 = the
6062   index name, #2 = the text
6063   \ifl@imakeidx%

```

```

6063 \if@edindex@fornote@%
6064 \IfSubStr[1]{#2}{|}{\get@index@command#2+}{\get@index@command#2|+}%
6065 \get@edindex@ledinnote@command%
6066 \expandafter\imki@wrindexentry{#1}{\@index@txt|(\@ledinnote@command
}{\thestartpageline}%
6067 \expandafter\imki@wrindexentry{#1}{\@index@txt|)\@ledinnote@command
}{\theendpageline}%
6068 \else%
6069 \get@edindex@hyperref{#2}%
6070 \imki@wrindexentry{#1}{\@index@txt\@edindex@hyperref}{\thepageline}%
6071 \fi%
6072 \else%
6073 \if@edindex@fornote@%
6074 \IfSubStr[1]{#2}{|}{\get@index@command#2+}{\get@index@command#2|+}%
6075 \get@edindex@ledinnote@command%
6076 \expandafter\protected@write@indexfile{%
6077 {\string\indexentry{\@index@txt|(\@ledinnote@command}{\thestartpageline}
6078 }%
6079 \expandafter\protected@write@indexfile{%
6080 {\string\indexentry{\@index@txt|)\@ledinnote@command}{\theendpageline}
6081 }%
6082 \else%
6083 \protected@write@indexfile{%
6084 {\string\indexentry{#2}{\thepageline}
6085 }%
6086 \fi%
6087 \fi%
6088 \endgroup
6089 \@esphack%
6090 }
6091 %

```

Need to add the definition of `\edindex` to `\makeindex`, and initialise `\edindex` to do nothing.

```

6092 \pretocmd{\makeindex}{%
6093 \def\edindex{\@bsphack
6094 \doedindexlabel
6095 \begingroup
6096 \@sanitize
6097 \@wredindex}}{}{}
6098 \newcommand{\edindex}[1]{\@bsphack\@esphack}
6099 %

```

XXVI.7 Hyperlink

`\hyperlinkformat` `\hyperlinkformat` command is to be used to have both a internal hyperlink and a format, when indexing.

```

6100 \newcommand{\hyperlinkformat}[3]{%

```

```

6101 \ifstrempy{#1}%
6102   {\hyperlink{#2}{#3}}%
6103   {\csuse{#1}{\hyperlink{#2}{#3}}}%
6104 }%
6105 %

```

\hyperlinkR \hyperlinkR command is to be used to create a internal hyperlink and \ledRflag, when indexing.

```

6106 \newcommand{\hyperlinkR}[2]{%
6107   \hyperlink{#1}{#2\@Rlineflag}%
6108 }%
6109 %
6110 %

```

\hyperlinkformatR \hyperlinkformatR command is to be used to create a internal hyperlink, a format and a \@Rlineflag, when indexing.

```

6111 \newcommand{\hyperlinkformatR}[3]{%
6112   \hyperlinkformat{#1}{#2}{#3\@Rlineflag}%
6113 }%
6114 %
6115 %

```

\get@edindex@hyperref \get@edindex@hyperref is to be used to define the \@edindex@hyperref macro, which, in index, links to the point where the index was called (with hyperref).

```

6116 \newcommand{\get@edindex@hyperref}[1]{%
6117 %

```

We have to disable temporary spaces to work through a xstring bug (or feature?)

```

6118 \edef\temp@{%
6119 \catcode`\ =9 %space need for catcode
6120 \detokenize{#1}%For active character in unicode
6121 \catcode`\ =10 % space need for catcode
6122 }%
6123 %

```

Now, we define \@edindex@hyperref if the hyperindex of hyperref is enabled.

```

6124 \ifdefequal{\iftrue}{\ifHy@hyperindex}{%
6125   \IfSubStr{\temp@}{|}%
6126   {\get@index@command#1+%
6127   \ifledRcol%
6128     \gdef\@edindex@hyperref{|\@index@parenthesis %space kept
6129     hyperlinkformatR{\@index@command}%
6130     {\edindexlab\thelabidx}}%
6131   \else%
6132     \gdef\@edindex@hyperref{|\@index@parenthesis %space kept
6133     hyperlinkformat{\@index@command}%

```

```

6134     {\edindexlab\thelabidx}}%
6135     \fi%
6136   }%
6137   {\get@index@command#1|+%
6138     \ifledRcol%
6139     \gdef\@edindex@hyperref{|hyperlinkR{\edindexlab\thelabidx}}%
6140     \else%
6141     \gdef\@edindex@hyperref{|hyperlink{\edindexlab\thelabidx}}%
6142     \fi%
6143   }%
6144 }%
6145 %

6146 % If we use both xindy and hyperref, first get the \protect\cs{
index@command} command.
6147 % Then define \protect\cs{@edindex@hyperref} in the form \verb+eledmacXXX+
6148 % \begin{macrocode}
6149 {\ifxindyhyperref%
6150   \IfSubStr{\temp@}{|}%
6151   {\get@index@command#1+}%
6152   {\get@index@command#1|+}%
6153   \gdef\@edindex@hyperref{|eledmac\thelabidx}%
6154 %

```

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`.

```

6155   \IfStrEq{\@index@parenthesis}{(}%
6156   {%
6157     \csxdef{xindyparenthesis@\@index@txt}{\thelabidx}%
6158     \gdef\@edindex@hyperref{| (eledmac\thelabidx}%
6159   }%
6160   {}%
6161 %

```

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`.

```

6162   \IfStrEq{\@index@parenthesis}{)}%
6163   {%
6164     \xdef\@edindex@hyperref{|)eledmac\csuse{xindyparenthesis@\@index@txt}}%
6165     \global\csundef{xindyparenthesis@\@index@txt}%
6166   }%
6167 %

```

Write in the `.xdy` file the attributes of the location.

```

6168   {%
6169     \immediate\write\eledmac@xindy@out{%

```

```

6170      (define-attributes ("eledmac\thelabidx"))^^J
6171      \space\space(markup-locref^^J
6172      \eledmacmarkuplocorefdepth^^J
6173      :open "\string\hyperlink%
6174      \ifledRcol R\fi%
6175      {\edindexlab\thelabidx}%
6176      {\ifdefempty{\@index@command}%
6177      {}%
6178      {\@backslashchar\@index@command}%
6179      {"^^J
6180      :close "}}^^J
6181      :attr "eledmac\thelabidx"^^J
6182      )
6183      }%
6184      }%
6185      %

```

And now, in any other case.

```

6186      \else%
6187      \gdef\@index@txt{#1}%
6188      \gdef\@edindex@hyperref{}%
6189      \fi%
6190      }%
6191      }
6192      %

```

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.

```

6193 \newcommand{\led@set@index@fornote}[1]{%
6194   \ifbool{indtl@innote}%
6195   {\let\index\nindex}%
6196   {}%
6197   \ifbool{indtl@notenumber}%
6198   {%
6199     \renewcommand{\index}[2][\indtl@jobname]{%
6200       \orig@index[##1]{%
6201         ##2|innotenumber{\csuse{thefootnote#1}}}%
6202       }%
6203     }%
6204   }%
6205   {}%
6206 }%
6207 %

```

`\led@reinit@index@fornote` The `\led@reinit@index@fornote` just reset the default value of `\index`.

```

6208 \newcommand{\led@reinit@index@fornote}{%
6209   \ifbool{indtl@innote}%
6210     {\let\index\orig@index}%
6211     {}%
6212   \ifbool{indtl@notenumber}%
6213     {\let\index\orig@index}%
6214     {}%
6215 }%
6216 %

```

XXVII Glossaries

Here, we define the `\gls`-like commands prefixed by `ed`, only if the package `glossaries` is loaded.

```

6217 \AtBeginDocument{%
6218   \@ifpackageloaded{glossaries}{%
6219     %

```

First those which arguments are `[<options>]{<label>}[<insert>]`.

```

6220   \renewcommand{\do}[1]{%
6221     \expandafter\DeclareRobustCommand\csname ed#1\endcsname[3][1,3,
usedefault]{%
6222       \doedindexlabel%
6223       \csname#1\endcsname[counter=pageline,##1]{##2}[##3]%
6224     }%
6225     \expandafter\WithSuffix\expandafter\DeclareRobustCommand\csname ed
#1\endcsname*[3][1,3,usedefault]{%
6226       \doedindexlabel%
6227       \csname#1\endcsname*[counter=pageline,##1]{##2}[##3]%
6228     }%
6229   }%
6230   \docsvlist{%
6231     gls,%
6232     Gls,%
6233     GLS,%
6234     glspl,%
6235     Glspl,%
6236     GLSpl,%
6237     glstext,%
6238     Glstext,%
6239     GLStext,%
6240     Glsfirst,%
6241     GLSfirst,%
6242     glsplural,%
6243     Glsplural,%
6244     GLSplural,%

```

```

6245     glsfirstplural,%
6246     Glsfirstplural,%
6247     GLSfirstplural,%
6248     glsname,%
6249     Glsname,%
6250     GLSname,%
6251     glssymbol,%
6252     Glsymbol,%
6253     GLSsymbol,%
6254     glsdesc,%
6255     Glsdesc,%
6256     GLSdesc,%
6257     glsuseri,%
6258     Glsuseri,%
6259     GLSuseri,%
6260     glsuserii,%
6261     Glsuserii,%
6262     GLSuserii,%
6263     glsuseriii,%
6264     Glsuseriii,%
6265     GLSuseriii,%
6266     glsuseriv,%
6267     Glsuseriv,%
6268     GLSuseriv,%
6269     glsuserv,%
6270     Glsuserv,%
6271     GLSuserv,%
6272     glsuservi,%
6273     Glsuservi,%
6274     GLSuservi%
6275     }%
6276 %

```

First those which arguments are [*options*]{*label*}{*link text*}.

```

6277     \renewcommand{\do}[1]{%
6278     \expandafter\DeclareRobustCommand\csname ed#1\endcsname[3][1,
usedefault]{%
6279     \doedindexlabel%
6280     \csname#1\endcsname[counter=pageline,##1]{##2}{##3}%
6281     }%
6282     \expandafter\WithSuffix\expandafter\DeclareRobustCommand\csname ed
#1\endcsname*[3][1,usedefault]{%
6283     \doedindexlabel%
6284     \csname#1\endcsname*[counter=pageline,##1]{##2}{##3}%
6285     }%
6286     }%
6287     \docsvlist{glsdisp,glslink}%
6288     }{}%
6289     }%
6290 %

```


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.

```
6291 \def\@hangingsymbol{}
6292 \newcommand*\sethangingsymbol[1]{%
6293   \gdef\@hangingsymbol{#1}%
6294 }%
6295 \newif\ifinstanza
6296 %
```

`\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.

```
6297 \newif\ifinserthangingsymbol
6298 \newcommand{\inserthangingsymbol}{%
6299   \ifinserthangingsymbol%
6300     \ifinstanza%
6301       \@hangingsymbol%
6302     \fi%
6303   \fi%
6304 }
6305 %
```

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.

```
6306 \newcommand*\ampersand{\char`\&}
6307
6308 %
```

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.

```
6309 \chardef\body=\catcode`\@
6310 \catcode`\@=11
6311 \chardef\next=\catcode`\&
6312 \catcode`\&=\active
6313
6314 %
```

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.

```
6315 \newcount\stanza@count
6316 \newlength{\stanzaindentbase}
6317 \setlength{\stanzaindentbase}{20pt}
6318
6319 %
```

`\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.

```
6320 \def\strip@szacnt#1,#2|{\def\@tempb{#1}\def\@tempa{#2|}}
6321 \newcommand*\setstanzavalues[2]{\def\@tempa{#2,,|}%
6322 \stanza@count\z@
6323 \def\next{\expandafter\strip@szacnt\@tempa
6324 \ifx\@tempb\empty\let\next\relax\else
6325 \expandafter\mathchardef\csname #1@\number\stanza@count
6326 \@endcsname\@tempb\relax
6327 \advance\stanza@count\@ne\fi\next}%
6328 \next}
6329
6330 %
```

`\setstanzaindents` In the original edmac, `\setstanzavalues{sza}{\langle...\rangle}` had to be called to set the in-
`\setstanzapenalties` dents, and similarly `\setstanzavalues{szp}{\langle...\rangle}` 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).

```
6331 \newcommand*\setstanzaindents[1]{\setstanzavalues{sza}{#1}}
6332 \newcommand*\setstanzapenalties[1]{\setstanzavalues{szp}{#1}}
```

```
6333 %
6334 %
```

\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.

```
6335 \newcounter{stanzaindentsrepetition}
6336 \newcount\stanza@modulo
6337
6338 \newcommand*{\managestanza@modulo}[0]{%
6339   \advance\stanza@modulo\@ne%
6340   \ifnum\stanza@modulo>\value{stanzaindentsrepetition}%
6341     \stanza@modulo\@ne%
6342   \fi%
6343 }
6344 %
```

\stanzaindent The macro `\stanzaindent`, when called at the beginning of a verse, changes the indentation normally defined for this verse by `\setstanzaindent`. The starred version **\stanzaindent*** skips the current verse for the repetition of stanza indent.

```
6345 \newcommand{\stanzaindent}[1]{%
6346   \hspace{\dimexpr#1\stanzaindentbase-\parindent\relax}%
6347   \ignorespaces%
6348 }%
6349 \WithSuffix\newcommand\stanzaindent*[1]{%
6350   \stanzaindent{#1}%
6351   \global\advance\stanza@modulo-\@ne%
6352   \ifnum\stanza@modulo=0%
6353     \global\stanza@modulo=\value{stanzaindentsrepetition}%
6354   \fi%
6355   \ignorespaces%
6356 }%
6357 %
```

XXVIII.5 Numbering stanza

Here, macro for numbering stanza. First, the stanza counter.

```
\thestanza58 \newcounter{stanza}
6359 \renewcommand{\thestanza}{%
6360   \textbf{\arabic{stanza}}%
6361 }
6362 %
```

`\ifnumberstanza` Then, macro to activate automatically numbering of stanza.

```
6363 \newif\ifnumberstanza%
6364 %
```

`\@insertstanzanumber` Now, macro called at the first line of of verse of a stanza.

```
6365 \newcommand{\@insertstanzanumber}[0]{%
6366   \ifnumberstanza%
6367     \ifl@dpairing%
6368       \ifledRcol%
6369         \stanzanumwrapper{\thestanzaR}%
6370       \else%
6371         \stanzanumwrapper{\thestanzaL}%
6372       \fi%
6373     \else%
6374       \stanzanumwrapper{\thestanza}%
6375     \fi%
6376     \setline{1}%
6377   \fi%
6378 }%
6379 %
```

`\@advancestanzanumber` Also a command to advance the counter of stanza.

```
6380 \newcommand{\@advancestanzanumber}[0]{%
6381   \ifnumberstanza%
6382     \ifl@dpairing%
6383       \ifledRcol%
6384         \addtocounter{stanzaR}{1}%
6385       \else%
6386         \addtocounter{stanzaL}{1}%
6387       \fi%
6388     \else%
6389       \addtocounter{stanza}{1}%
6390     \fi%
6391   \fi%
6392 }%
6393 %
```

`\stanzanumwrapper` And finally, the wrapper for stanza number

```
6394 \newcommand{\stanzanumwrapper}[1]{%
6395   \flagstanza{#1}%
6396 }%
6397 %
```

XXVIII.6 Stanza number in note

Here, the command called when printing stanza number in notes.

```

6398 \newcommand{\printstanza}[0]{%
6399   \ifbool{bool{!@dpairing}} or bool{!@dprintingpages} or bool{
1@dprintingcolumns}}{%
6400     \ifledRcol{%
6401       \thestanzaR%
6402     }else%
6403       \thestanzaL%
6404     \fi%
6405   }{%
6406     \thestanza%
6407   }%
6408 }
6409 %

```

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.

```

6410 \newcommandx{\stanza@line}[1][1]{
6411   \ifnum\value{stanzaindentrepetition}=0
6412     \parindent=\csname sza@number\stanza@count
6413       @\endcsname\stanzaindentbase
6414   }else
6415     \parindent=\csname sza@number\stanza@modulo
6416       @\endcsname\stanzaindentbase
6417     \managestanza@modulo
6418   \fi
6419   \pstart[#1]\stanza@hang\ignorespaces}
6420 \xdef\stanza@hang{\noexpand\leavevmode\noexpand\startlock
6421   \hangindent\expandafter
6422   \noexpand\csname sza@0@\endcsname\stanzaindentbase
6423   \hangafter\@ne}
6424 \def\sza@penalty{\count@\csname szp@number\stanza@count @\endcsname
6425   \ifnum\count@>\@M\advance\count@-\@M\penalty-\else
6426   \penalty\fi\count@}
6427 %

```

`\@startstanza` Now we have the components of the `\stanza` macro, which appears at the start of a group of lines. This macro initializes the count and checks to see if hanging indentation and penalties are to be included. Hanging indentation suspends the line count, so that the enumeration is by verse line rather than by print line. If the print line count is

desired, invoke `\let\startlock\relax` and do the same for `\endlock`. Here and above we have used `\xdef` to make the stored macros take up a bit less space, but it also makes them more obscure to the reader. Lines of the stanza are delimited by ampersands `&`. The last line of the stanza must end with `\&`.

```

6428 \xdef\@startstanza[#1]{%
6429   \noexpand\instanzatrue\expandafter
6430   \begingroup%
6431   \catcode`\noexpand\&\active%
6432   \global\stanza@count\@ne\stanza@modulo\@ne
6433   \noexpand\ifnum\expandafter\noexpand
6434   \csname sza@0@\endcsname=\z@\let\noexpand\stanza@hang\relax
6435   \let\noexpand\endlock\relax\noexpand\else\interlinepenalty
6436   \@M\rightskip\z@ plus 1fil\relax\noexpand\fi\noexpand\ifnum
6437   \expandafter\noexpand\csname szp@0@\endcsname=\z@
6438   \let\noexpand\sza@penalty\relax\noexpand\fi%
6439   \def\noexpand&{%
6440     \noexpand\newverse[] []}%
6441   \def\noexpand\&{\noexpand\@stopstanza}%
6442   \noexpand\@advancestanzanumber%
6443   \noexpand\stanza@line[#1]\noexpand\@insertstanzanumber%
6444   \let\par\relax\ignorespaces%No paragraph in verses
6445 }
6446
6447 \newcommandx{\stanza}[1][1,usedefault]{\@startstanza[#1]}
6448
6449 \newcommandx{\@stopstanza}[1][1,usedefault]{%
6450   \unskip%
6451   \endlock%
6452   \pend[#1]%
6453   \endgroup%
6454   \instanzafalse%
6455 }
6456
6457 \newcommandx*\{newverse}[2][1,2,usedefault]{%
6458   \unskip%
6459   \endlock\pend[#1]\sza@penalty\global%
6460   \advance\stanza@count\@ne\stanza@line[#2]%
6461   }
6462
6463 %

```

\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`.

```

6464 \newcommand*\{flagstanza}[2][\stanzaindentbase]{%
6465   \hskip -#1\llap{#2}\hskip #1\ignorespaces}
6466
6467 %

```

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.

```

6468 \catcode`\&=\next
6469 \catcode`\@=\body
6470 \setstanzavalues{szp}{0}
6471
6472 %

```

XXIX Arrays and tables

XXIX.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.

```

6473 \newtoks\@emptytoks
6474
6475 %

```

The rest is from `amsmath`.

`\l@denbody` A token register to contain the body.

```

6476 \newtoks\l@denbody
6477
6478 %

```

`\addtol@denbody` `\addtol@denbody{arg}` adds `arg` to the token register `\l@denbody`.

```

6479 \newcommand{\addtol@denbody}[1]{%
6480 \global\l@denbody\expandafter{\the\l@denbody#1}}
6481
6482 %

```

`\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`.

```

6483 \newcommand{\l@dcollect@body}[1]{%
6484   \l@denbody{\expandafter#1\expandafter{\the\l@denbody}}%
6485   \edef\processl@denbody{\the\l@denbody\noexpand\end{\@currenvir}}%
6486   \l@denbody\@emptytoks \def\l@dbegin@stack{b}%
6487   \begingroup
6488     \expandafter\let\csname\@currenvir\endcsname\l@dcollect@@body
6489     \edef\processl@denbody{\expandafter\noexpand\csname\@currenvir\
endcsname}%
6490     \processl@denbody%
6491   }%
6492
6493 %

```

`\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.

```

6494 \def\l@dpush@begins#1\begin#2{%
6495   \ifx\end#2\else b\expandafter\l@dpush@begins\fi}
6496
6497 %

```

`\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.

```

6498 \def\l@dcollect@@body#1\end#2{%
6499   \edef\l@dbegin@stack{\l@dpush@begins#1\begin\end
6500     \expandafter\@gobble\l@dbegin@stack}%
6501   \ifx\@empty\l@dbegin@stack
6502     \endgroup
6503     \@checkend{#2}%
6504     \addtol@denbody{#1}%
6505   \else
6506     \addtol@denbody{#1\end{#2}}%
6507   \fi
6508   \processl@denbody % A little tricky! Note the grouping
6509 }
6510
6511 %

```


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>

```

XXIX.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`.

XXIX.2.1 Disabling and restoring commands

`\l@dtabnoexpands` More no expansion for critical and familiar footnotes in tabular environment.

```

6512 \newcommand*{\l@dtabnoexpands}{%
6513   \let\rtab=0%
6514   \let\ctab=0%
6515   \let\ltab=0%
6516   \let\rtabtext=0%
6517   \let\ltabtext=0%
6518   \let\ctabtext=0%
6519   \let\edbeforetab=0%
6520   \let\edaftertab=0%
6521   \let\edatleft=0%
6522   \let\edatright=0%
6523   \let\edvertline=0%
6524   \let\edvertdots=0%
6525   \let\edrowfill=0%
6526 }
6527
6528 %

```

`\disable@familiarnotes` Macros to disable and restore familiar notes, to prevent them from printing multiple times in `edtabularx` and `edarrayx` environments.

```

6529 \newcommand{\disable@familiarnotes}{%
6530   \unless\ifnofamiliar%
6531     \def\do##1{%
6532       \csletcs{footnote@@##1}{footnote##1}%
6533       \expandafter\renewcommand \csname footnote##1\endcsname[1]{%
6534         \protected@csxdef{@thefnmark##1}{\csuse{thefootnote##1}}%
6535         \csuse{@footnotemark##1}%
6536       }%
6537     }%
6538     \dolistloop{\@series}%
6539   \fi%
6540 }%
6541 \newcommand{\restore@familiarnotes}{%
6542   \unless\ifnofamiliar%
6543     \def\do##1{%
6544       \csletcs{footnote##1}{footnote@@##1}%
6545     }%
6546     \dolistloop{\@series}%
6547   \fi%

```

```

6548 }%
6549
6550 %

```

\disable@sidenotes The same, for side notes.

\restore@sidenotes

```

6551 \newcommand{\disable@sidenotes}{%
6552   \let\@@ledrightnote\ledrightnote%
6553   \let\@@ledleftnote\ledleftnote%
6554   \let\@@ledsidenote\ledsidenote%
6555   \let\ledrightnote@gobble%
6556   \let\ledleftnote@gobble%
6557   \let\ledsidenote@gobble%
6558 }%
6559 \newcommand{\restore@sidenotes}{%
6560   \let\ledrightnote\@@ledrightnote%
6561   \let\ledleftnote\@@ledleftnote%
6562   \let\ledsidenote\@@ledsidenote%
6563 }%
6564 %

```

\disable@notes Disable/restore side and familiar notes.

\restore@notes

```

6565 \newcommand{\disable@notes}{%
6566   \disable@sidenotes%
6567   \disable@familiarnotes%
6568 }%
6569 \newcommand{\restore@notes}{%
6570   \restore@sidenotes%
6571   \restore@familiarnotes%
6572 }%
6573 %

```

\EDTEXT We need to be able to modify the `\edtext` macros and also restore their original definitions.

\xedtext

```

6574 \let\EDTEXT=\edtext
6575 \newcommand{\xedtext}[2]{\EDTEXT{#1}{#2}}
6576 %

```

\EDLABEL We need to be able to modify and restore the `\edlabel` macro.

\xedlabel

```

6577 \let\EDLABEL=\edlabel
6578 \newcommand*{\xedlabel}[1]{\EDLABEL{#1}}
6579 %

```

\EDINDEX Macros supporting modification and restoration of `\edindex`.

\xedindex

\nulledindex

```

6580 \let\EDINDEX=\edindex
6581 \newcommand{\xedindex}{\@bsphack%
6582   \@ifnextchar [{\l@d@index}{\l@d@index[\jobname]}}
6583 \newcommand{\nulledindex}[2][\jobname]{\@bsphack\@esphack}
6584
6585 %

```

\@line@num Macro supporting restoration of \linenum.

```

6586 \let\@line@num=\linenum
6587 %

```

\l@gobblearg **\l@gobbleoptarg**[<arg>]{<arg>} replaces these two arguments (first is optional) by \relax.

```

6588 \newcommand*{\l@gobbleoptarg}[2][\relax]%
6589
6590 %

```

\Relax₉₁ \let\Relax=\relax

\NEXT₉₂ \let\NEXT=\next

```

6593
6594 %

```

\l@dmodforedtext Modify and restore various macros for when \edtext is used.

\l@drestoreforedtext

```

6595 \newcommand{\l@dmodforedtext}{%
6596   \let\edtext\relax
6597   \def\do##1{\global\csletcs{##1footnote}{\l@gobbleoptarg}}%
6598   \dolistloop{\@series}%
6599   \let\edindex\nulledindex
6600   \let\linenum\@gobble}
6601 \newcommand{\l@drestoreforedtext}{%
6602   \def\do##1{\global\csletcs{##1footnote}{##1@footnote}}
6603   \dolistloop{\@series}%
6604   \let\edindex\xedindex}
6605 %

```

\l@dnullfills Nullify and restore some column fillers, etc.

\l@drestorefills

```

6606 \newcommand{\l@dnullfills}{%
6607   \def\edlabel##1{}%
6608   \def\edrowfill##1##2##3{}%
6609 }
6610 \newcommand{\l@drestorefills}{%
6611   \def\edrowfill##1##2##3{\@EDROWFILL@{##1}{##2}{##3}}%
6612 }
6613
6614 %

```

`\letsforverteilen` Gathers some lets and other code that is common to the `*verteilen*` macros.

```

6615 \newcommand{\letsforverteilen}{%
6616   \let\edtext\xeddtext
6617   \let\edindex\xeddindex
6618   \def\do##1{\global\csletcs{##1footnote}{##1@footnote}}
6619   \dolistloop{\@series}%
6620   \let\linenum\@line@num
6621   \hilfe skip=\l@dcolwidth%
6622   \advance\hilfe skip by -\wd\hilfe box
6623   \def\edlabel##1{\xedlabel{##1}}
6624
6625 %

```

`\disablel@dtabfeet` Declarations for using or using `\edtext` inside tabulars. The default at this point is for `\enablel@dtabfeet` `\edtext`.

```

6626 \newcommand\disablel@dtabfeet{\l@dmodforedtext}%
6627 \newcommand\enablel@dtabfeet{\l@drestoreforedtext}%
6628 %

```

XXIX.2.2 Counters, boxes and lengths

`\l@dampcount` `\l@dampcount` is a counter for the & column dividers and `\l@dcolcount` is a counter for the columns.

```

6629 \newcount\l@dampcount
6630 \l@dampcount=1\relax
6631 \newcount\l@dcolcount
6632 \l@dcolcount=0\relax
6633
6634 %

```

`\hilfe box` Some (temporary) helper items.

```

\hilfe skip
6635 \newbox\hilfe box
\hilfe box
6636 \newskip\hilfe skip
\hilfe count
6637 \newbox\hilfe box
6638 \newcount\hilfe count
6639
6640 %

```

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).

```

6641 \newdimen\dcoli
6642 \newdimen\dcolii
6643 \newdimen\dcoliii
6644 \newdimen\dcoliv
6645 \newdimen\dcolv

```

```

6646 \newdimen\dc colvi
6647 \newdimen\dc colvii
6648 \newdimen\dc colviii
6649 \newdimen\dc colix
6650 \newdimen\dc colx
6651 \newdimen\dc colxi
6652 \newdimen\dc colxii
6653 \newdimen\dc colxiii
6654 \newdimen\dc colxiv
6655 \newdimen\dc colxv
6656 \newdimen\dc colxvi
6657 \newdimen\dc colxvii
6658 \newdimen\dc colxviii
6659 \newdimen\dc colxix
6660 \newdimen\dc colxx
6661 \newdimen\dc colxxi
6662 \newdimen\dc colxxii
6663 \newdimen\dc colxxiii
6664 \newdimen\dc colxxiv
6665 \newdimen\dc colxxv
6666 \newdimen\dc colxxvi
6667 \newdimen\dc colxxvii
6668 \newdimen\dc colxxviii
6669 \newdimen\dc colxxix
6670 \newdimen\dc colxxx
6671 \newdimen\dc colerr % added for error handling
6672
6673 %

```

\l@dc colwidth This is a cunning way of storing the columnwidths indexed by the column number \l@dc colcount, like an array. (was \Dimenzuordnung)

```

6674 \newcommand{\l@dc colwidth}{\ifcase \the\l@dc colcount \dc oli %???
6675 \or \dc oli \or \dc olii \or \dc oliii
6676 \or \dc oliv \or \dc olv \or \dc olvi
6677 \or \dc olvii \or \dc olviii \or \dc olix \or \dc olx
6678 \or \dc olxi \or \dc olxii \or \dc olxiii
6679 \or \dc olxiv \or \dc olxv \or \dc olxvi
6680 \or \dc olxvii \or \dc olxviii \or \dc olxix \or \dc olxx
6681 \or \dc olxxi \or \dc olxxii \or \dc olxxiii
6682 \or \dc olxxiv \or \dc olxxv \or \dc olxxvi
6683 \or \dc olxxvii \or \dc olxxviii \or \dc olxxix \or \dc olxxx
6684 \else \dc olerr \fi}
6685
6686 %

```

\step1@dc colcount This increments the column counter, and issues an error message if it is too large.

```

6687 \newcommand*{\step1@dc colcount}{\advance\l@dc colcount\@ne
6688 \ifnum\l@dc colcount>30\relax

```

```

6689 \led@err@TooManyColumns
6690 \fi}
6691
6692 %

```

\l@setmaxcolwidth Sets the column width to the maximum value seen so far.

```

6693 \newcommand{\l@setmaxcolwidth}{%
6694   \ifdim\l@dcwidth < \wd\hilfsbox
6695     \l@dcwidth = \wd\hilfsbox
6696   \else \relax \fi}
6697
6698 %

```

\measuremcell Measure (recursively) the width required for a math cell.

```

6699 \def\measuremcell #1{%
6700   \ifx #1\ \ifnum\l@dcwidth=0\let\NEXT\relax%
6701     \else\l@dcheckcols%
6702       \l@dcwidth=0%
6703       \let\NEXT\measuremcell%
6704     \fi%
6705   \else\setbox\hilfsbox=\hbox{$\displaystyle{#1}$}%
6706     \step\l@dcwidth%
6707     \l@setmaxcolwidth%
6708     \let\NEXT\measuremcell%
6709   \fi\NEXT}
6710
6711 %

```

\measuretcell Measure (recursively) the width required for a text cell.

```

6712 \def\measuretcell #1{%
6713   \ifx #1\ \ifnum\l@dcwidth=0\let\NEXT\relax%
6714     \else\l@dcheckcols%
6715       \l@dcwidth=0%
6716       \let\NEXT\measuretcell%
6717     \fi%
6718   \else\setbox\hilfsbox=\hbox{#1}%
6719     \step\l@dcwidth%
6720     \l@setmaxcolwidth%
6721     \let\NEXT\measuretcell%
6722   \fi\NEXT}
6723
6724 %

```

\measuremrow Measure (recursively) the width required for a math row.

```

6725 \def\measuremrow #1{%
6726   \ifx #1\ \let\NEXT\relax%

```



```

6727 \else\measuremcell #1&\\&\\&%
6728 \let\NEXT\measuremrow%
6729 \fi\NEXT}
6730 %

```

\measuretrow Measure (recursively) the width required for a text row.

```

6731 \def\measuretrow #1\{/%
6732 \ifx #1&\let\NEXT\relax%
6733 \else\measuretcell #1&\\&\\&%
6734 \let\NEXT\measuretrow%
6735 \fi\NEXT}
6736
6737 %

```

\edtabcolsep The length \edtabcolsep controls the distance between columns.

```

6738 \newskip\edtabcolsep
6739 \global\edtabcolsep=10pt
6740
6741 %

```

\variab₄₂ \newcommand{\variab}{\relax}

```

6743
6744 %

```

\l@dcheckcols Check that the number of columns is consistent.

```

6745 \newcommand*{\l@dcheckcols}{/%
6746 \ifnum\l@dcolcount=1\relax
6747 \else
6748 \ifnum\l@dampcount=1\relax
6749 \else
6750 \ifnum\l@dcolcount=\l@dampcount\relax
6751 \else
6752 \l@d@err@UnequalColumns
6753 \fi
6754 \fi
6755 \l@dampcount=\l@dcolcount
6756 \fi}
6757
6758 %

```

\edfilldimen A length.

```

6759 \newdimen\edfilldimen
6760 \edfilldimen=0pt
6761
6762 %

```

`\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`

```
6763 \newcounter{addcolcount}
6764 \renewcommand{\theadcolcount}{\romannumeral \c@addcolcount}
6765 %
```

XXIX.2.3 Tabular typesetting

`\setmcellright` Typeset (recursively) cells of display math right justified.

```
6766 \def\setmcellright #1{\def\edlabel##1{}%
6767   \let\edindex\nulledindex
6768   \ifx #1\ \ifnum\l@dcolcount=0%\removelastskip
6769     \let\Next\relax%
6770   \else\l@dcolcount=0%
6771     \let\Next=\setmcellright%
6772   \fi%
6773 \else%
6774   \disablel@dtabfeet%
6775   \step1@dcolcount%
6776   \disable@notes%
6777   \setbox\hilfsbox=\hbox{$\displaystyle{#1}$}%
6778   \restore@notes%
6779   \letsforverteilen%
6780   \hskip\hilfsskip$\displaystyle{#1}$%
6781   \hskip\edtabcolsep%
6782   \let\Next=\setmcellright%
6783 \fi\Next}
6784
6785 %
```

`\settcclright` Typeset (recursively) cells of text right justified.

```
6786 \def\settcclright #1{\def\edlabel##1{}%
6787   \let\edindex\nulledindex
6788   \ifx #1\ \ifnum\l@dcolcount=0%\removelastskip
6789     \let\Next\relax%
6790   \else\l@dcolcount=0%
6791     \let\Next=\settcclright%
6792   \fi%
6793 \else%
6794   \disablel@dtabfeet%
6795   \step1@dcolcount%
6796   \disable@notes%
6797   \setbox\hilfsbox=\hbox{#1}%
6798   \restore@notes%
6799   \letsforverteilen%
6800   \hskip\hilfsskip#1%
```

```

6801         \hskip\edtabcolsep%
6802         \let\Next=\settcellright%
6803     \fi\Next}
6804 %

```

\setmcellleft Typeset (recursively) cells of display math left justified.

```

6805 \def\setmcellleft #1&{\def\edlabel##1{}}%
6806     \let\edindex\nulledindex
6807     \ifx #1\\ \ifnum\l@dc@colcount=0 \let\Next\relax%
6808         \else\l@dc@colcount=0%
6809         \let\Next=\setmcellleft%
6810         \fi%
6811     \else \disablel@dtabfeet%
6812         \stepl@dc@colcount%
6813         \disable@notes%
6814         \setbox\hilfsbox=\hbox{$\displaystyle{#1}$}%
6815         \restore@notes%
6816         \letsforverteilen%
6817         $\displaystyle{#1}$\hskip\hilfsskip\hskip\edtabcolsep%
6818         \let\Next=\setmcellleft%
6819     \fi\Next}
6820 %
6821 %

```

\settcellleft Typeset (recursively) cells of text left justified.

```

6822 \def\settcellleft #1&{\def\edlabel##1{}}%
6823     \let\edindex\nulledindex
6824     \ifx #1\\ \ifnum\l@dc@colcount=0 \let\Next\relax%
6825         \else\l@dc@colcount=0%
6826         \let\Next=\settcellleft%
6827         \fi%
6828     \else \disablel@dtabfeet%
6829         \stepl@dc@colcount%
6830         \disable@notes%
6831         \setbox\hilfsbox=\hbox{#1}%
6832         \restore@notes%
6833         \letsforverteilen%
6834         #1\hskip\hilfsskip\hskip\edtabcolsep%
6835         \let\Next=\settcellleft%
6836     \fi\Next}
6837 %

```

\setmcellcenter Typeset (recursively) cells of display math centered.

```

6838 \def\setmcellcenter #1&{\def\edlabel##1{}}%
6839     \let\edindex\nulledindex
6840     \ifx #1\\ \ifnum\l@dc@colcount=0\let\Next\relax%
6841         \else\l@dc@colcount=0%

```

```

6842         \let\Next=\setmcellcenter%
6843     \fi%
6844 \else    \disablel@dtabfeet%
6845         \stepl@dcolcount%
6846         \disable@notes%
6847         \setbox\hilfsbox=\hbox{$\displaystyle{#1}$}%
6848         \restore@notes%
6849         \letsforverteilen%
6850         \hskip 0.5\hilfsskip$\displaystyle{#1}$\hskip0.5\hilfsskip%
6851         \hskip\edtabcolsep%
6852         \let\Next=\setmcellcenter%
6853 \fi\Next}
6854
6855 %

```

\settcclcenter Typeset (recursively) cells of text centered.

```

6856 \def\settcclcenter #1&{\def\edlabel##1{}%
6857     \let\edindex\nulledindex
6858     \ifx #1\\ \ifnum\l@dcolcount=0 \let\Next\relax%
6859         \else\l@dcolcount=0%
6860             \let\Next=\settcclcenter%
6861         \fi%
6862     \else    \disablel@dtabfeet%
6863             \stepl@dcolcount%
6864             \disable@notes%
6865             \setbox\hilfsbox=\hbox{#1}%
6866             \restore@notes%
6867             \letsforverteilen%
6868             \hskip 0.5\hilfsskip #1\hskip 0.5\hilfsskip%
6869             \hskip\edtabcolsep%
6870             \let\Next=\settcclcenter%
6871     \fi\Next}
6872
6873 %

```

\NEXT \let\NEXT=\relax

```

6875
6876 %

```

\setmrowright Typeset (recursively) rows of right justified math.

```

6877 \def\setmrowright #1\\{%
6878     \ifx #1& \let\NEXT\relax
6879     \else \centerline{\setmcellright #1&\\&\\&}
6880         \let\NEXT=\setmrowright
6881     \fi\NEXT}
6882 %

```

\settroright Typeset (recursively) rows of right justified text.

```

6883 \def\settroright #1\{\%
6884   \ifx #1& \let\NEXT\relax
6885   \else \centerline{\settcellright #1&\&\&\&}
6886         \let\NEXT=\settroright
6887   \fi\NEXT}
6888
6889 %

```

\setmrowleft Typeset (recursively) rows of left justified math.

```

6890 \def\setmrowleft #1\{\%
6891   \ifx #1& \let\NEXT\relax
6892   \else \centerline{\setmcellleft #1&\&\&\&}
6893         \let\NEXT=\setmrowleft
6894   \fi\NEXT}
6895 %

```

\settrorleft Typeset (recursively) rows of left justified text.

```

6896 \def\settrorleft #1\{\%
6897   \ifx #1& \let\NEXT\relax
6898   \else \centerline{\settcellleft #1&\&\&\&}
6899         \let\NEXT=\settrorleft
6900   \fi\NEXT}
6901
6902 %

```

\setmrowcenter Typeset (recursively) rows of centered math.

```

6903 \def\setmrowcenter #1\{\%
6904   \ifx #1& \let\NEXT\relax%
6905   \else \centerline{\setmcellcenter #1&\&\&\&}
6906         \let\NEXT=\setmrowcenter
6907   \fi\NEXT}
6908 %

```

\settrorcenter Typeset (recursively) rows of centered text.

```

6909 \def\settrorcenter #1\{\%
6910   \ifx #1& \let\NEXT\relax
6911   \else \centerline{\settcellcenter #1&\&\&\&}
6912         \let\NEXT=\settrorcenter
6913   \fi\NEXT}
6914
6915 %

```

\nullsetzen `\newcommand{\nullsetzen}{%`

```

6917   \step1@dcolcount%
6918   \l@dcolwidth=0pt%
6919   \ifnum\l@dcolcount=30\let\NEXT\relax%
6920       \l@dcolcount=0\relax
6921   \else\let\NEXT\nullsetzen%
6922   \fi\NEXT}
6923
6924 %
```

\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.`

```

6925 \newcommand{\edatleft}[3][\@empty]{%
6926   \ifx#1\@empty
6927     \vbox to 10pt{\vss\hbox{$\left#2\vrule width0pt height #3
6928         depth 0pt \right. $\hss}\vfil}
6929   \else
6930     \vbox to 4pt{\vss\hbox{$#1\left#2\vrule width0pt height #3
6931         depth 0pt \right. $\}\vfil}
6932   \fi}
6933 %
```

\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.`

```

6934 \newcommand{\edatright}[3][\@empty]{%
6935   \ifx#1\@empty
6936     \vbox to 10pt{\vss\hbox{$\left.\vrule width0pt height #3
6937         depth 0pt \right#2 $\hss}\vfil}
6938   \else
6939     \vbox to 4pt{\vss\hbox{$\left.\vrule width0pt height #3
6940         depth 0pt \right#2 #1 $\}\vfil}
6941   \fi}
6942
6943 %
```

\edvertline `\edvertline{ $\langle len \rangle$ }` vertical line $\langle len \rangle$ high.

```

6944 \newcommand{\edvertline}[1]{\vbox to 8pt{\vss\hbox{\vrule height #1}\vfil}}
6945
6946 %
```

\edvertdots `\edvertdots{ $\langle len \rangle$ }` vertical dotted line $\langle len \rangle$ high.

```

6947 \newcommand{\edvertdots}[1]{\vbox to 1pt{\vss\vbox to #1%
6948   {\cleaders\hbox{$\math\hbox{.}\vbox to 0.5em{ }$\}\vfil}}}
6949
6950 %
```

`\l@dtabaddcols` `\l@dtabaddcols{<startcol>}{<endcol>}` adds the widths of the columns `<startcol>` through `<endcol>` to `\edfilldimen`. It is a \TeX style reimplementation of the original `\@add@`.

```

6951 \newcommand{\l@dtabaddcols}[2]{%
6952   \l@dcheckstartend{#1}{#2}%
6953   \ifl@dstartendok
6954   \setcounter{addcolcount}{#1}%
6955   \@whilenum \value{addcolcount}<#2\relax \do
6956   {\advance\edfilldimen by \the\csname dcol\theaddcolcount\endcsname
6957    \advance\edfilldimen by \edtabcolsep
6958    \stepcounter{addcolcount}}%
6959   \advance\edfilldimen by \the\csname dcol\theaddcolcount\endcsname
6960   \fi
6961 }
6962 %
6963 %

```

`\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.

```

6964 \newif\ifl@dstartendok
6965 \newcommand{\l@dcheckstartend}[2]{%
6966   \l@dstartendoktrue
6967   \ifnum #1<\@ne
6968     \l@dstartendokfalse
6969     \led@err@LowStartColumn
6970   \fi
6971   \ifnum #2>30\relax
6972     \l@dstartendokfalse
6973     \led@err@HighEndColumn
6974   \fi
6975   \ifnum #1>#2\relax
6976     \l@dstartendokfalse
6977     \led@err@ReverseColumns
6978   \fi
6979 }
6980 %
6981 %

```

`\edrowfill` `\edrowfill{<startcol>}{<endcol>}` fill fills columns `<startcol>` to `<endcol>` inclusive with `<fill>` (e.g. `\hrulefill`, `\upbracefill`). This is a \TeX style reimplementation and generalization of the original `\waklam`, `\Waklam`, `\waklamec`, `\wastricht` and `\wapunktetel` macros.

```

6982 \newcommand*\edrowfill}[3]{%
6983   \l@dtabaddcols{#1}{#2}%
6984   \hb@xt@ \the\l@dcolwidth{\hb@xt@ \the\edfilldimen{#3}\hss}}
6985 \let\@edrowfill@=\edrowfill
6986 \def\@EDROWFILL@#1#2#3{\@edrowfill@{#1}{#2}{#3}}

```

```
6987
6988 %
```

\edbeforetab The macro `\edbeforetab{⟨text⟩}{⟨math⟩}` puts `⟨text⟩` at the left margin before array cell entry `⟨math⟩`. Conversely, the macro `\edaftertab{⟨math⟩}{⟨text⟩}` puts `⟨text⟩` at the right margin after array cell entry `⟨math⟩`. `\edbeforetab` should be in the first column and `\edaftertab` in the last column. The following macros support these.

\leftltab `\leftltab{⟨text⟩}` for `\edbeforetab` in `\ltab`.

```
6989 \newcommand{\leftltab}[1]{%
6990   \hb@xt@z@{\vbox{\edtabindent%
6991     \moveleft\Hilfsskip\hbox{\ #1}}\hss}}
6992
6993 %
```

\leftrtab `\leftrtab{⟨text⟩}{⟨math⟩}` for `\edbeforetab` in `\rtab`.

```
6994 \newcommand{\leftrtab}[2]{%
6995   #2\hb@xt@z@{\vbox{\edtabindent%
6996     \advance\Hilfsskip by\dcoli%
6997     \moveleft\Hilfsskip\hbox{\ #1}}\hss}}
6998
6999 %
```

\leftctab `\leftctab{⟨text⟩}{⟨math⟩}` for `\edbeforetab` in `\ctab`.

```
7000 \newcommand{\leftctab}[2]{%
7001   \hb@xt@z@{\vbox{\edtabindent\l@dcolcount=\l@dampcount%
7002     \advance\Hilfsskip by 0.5\dcoli%
7003     \setbox\hilfsbox=\hbox{\def\edlabel##1{}%
7004       \disablel@dtabfeet$\displaystyle{#2}$}%
7005     \advance\Hilfsskip by -0.5\wd\hilfsbox%
7006     \moveleft\Hilfsskip\hbox{\ #1}}\hss}%
7007   #2}
7008
7009 %
```

\rightctab `\rightctab{⟨math⟩}{⟨text⟩}` for `\edaftertab` in `\ctab`.

```
7010 \newcommand{\rightctab}[2]{%
7011   \setbox\hilfsbox=\hbox{\def\edlabel##1{}%
7012     \disablel@dtabfeet#2}\l@dampcount=\l@dcolcount%
7013   #1\hb@xt@z@{\vbox{\edtabindent\l@dcolcount=\l@dampcount%
7014     \advance\Hilfsskip by 0.5\l@dcolwidth%
7015     \advance\Hilfsskip by -\wd\hilfsbox%
7016     \setbox\hilfsbox=\hbox{\def\edlabel##1{}%
7017       \disablel@dtabfeet$\displaystyle{#1}$}%
7018     \advance\Hilfsskip by -0.5\wd\hilfsbox%
```



```

7019 \advance\Hilfsskip by \edtabcolsep%
7020 \moveright\Hilfsskip\hbox{ #2}}\hss}%
7021 }
7022
7023 %

```

\rightltab `\rightltab{<math><math>}{<text>}` for `\edaftertab` in `\ltab`.

```

7024 \newcommand{\rightltab}[2]{%
7025     \setbox\hilfsbox=\hbox{\def\edlabel##1{%
7026         \disablel@dtabfeet#2}\l@dampcount=\l@dcolcount%
7027         #1\hb@xt@{\z@\vbox{\edtabindent\l@dcolcount=\l@dampcount%
7028         \advance\Hilfsskip by\l@dcolwidth%
7029         \advance\Hilfsskip by-\wd\hilfsbox%
7030         \setbox\hilfsbox=\hbox{\def\edlabel##1{%
7031             \disablel@dtabfeet$\displaystyle{#1}$}%
7032             \advance\Hilfsskip by-\wd\hilfsbox%
7033             \advance\Hilfsskip by\edtabcolsep%
7034             \moveright\Hilfsskip\hbox{ #2}}\hss}%
7035         }
7036     }
7037 %

```

\rightrtab `\rightrtab{<math><math>}{<text>}` for `\edaftertab` in `\rtab`.

```

7038 \newcommand{\rightrtab}[2]{%
7039     \setbox\hilfsbox=\hbox{\def\edlabel##1{%
7040         \disablel@dtabfeet#2}%
7041         #1\hb@xt@{\z@\vbox{\edtabindent%
7042         \advance\Hilfsskip by-\wd\hilfsbox%
7043         \advance\Hilfsskip by\edtabcolsep%
7044         \moveright\Hilfsskip\hbox{ #2}}\hss}%
7045     }
7046
7047 %

```

\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.

```

7048 \newcommand{\rtab}[1]{%
7049     \l@dnnullfills
7050     \def\edbeforetab##1##2{\lefttab{##1}{##2}}%
7051     \def\edaftertab##1##2{\righttab{##1}{##2}}%
7052     \measurebody{#1}%
7053     \l@drestorefills
7054     \variab
7055     \setmrowright #1\&\&%
7056     \enablel@dtabfeet}
7057
7058 %

```

`\measurebody` `\measurebody{⟨body⟩}` measures the array `⟨body⟩`.

```
7059 \newcommand{\measurebody}[1]{%
7060   \disablel@dtabfeet%
7061   \l@dcolcount=0%
7062   \nullsetzen%
7063   \l@dcolcount=0
7064   \measuremrow #1\\&\\%
7065   \global\l@dampcount=1}
7066
7067 %
```

`\rtabtext` `\rtabtext{⟨body⟩}` typesets `⟨body⟩` as a tabular with the entries right justified.

```
7068 \newcommand{\rtabtext}[1]{%
7069   \l@dnullfills
7070   \measuretbody{#1}%
7071   \l@drestorefills
7072   \variab
7073   \setthrowright #1\\&\\%
7074   \enablel@dtabfeet}
7075
7076 %
```

`\measuretbody` `\measuretbody{⟨body⟩}` measures the tabular `⟨body⟩`.

```
7077 \newcommand{\measuretbody}[1]{%
7078   \disable@notes%
7079   \disablel@dtabfeet%
7080   \l@dcolcount=0%
7081   \nullsetzen%
7082   \l@dcolcount=0
7083   \measuretrow #1\\&\\%
7084   \restore@notes%
7085   \global\l@dampcount=1}
7086
7087 %
```

`\ltab` Array with entries left justified.

```
\edbeforetab
\edaftertab
7088 \newcommand{\ltab}[1]{%
7089   \l@dnullfills
7090   \def\edbeforetab##1##2{\leftltab{##1}{##2}}%
7091   \def\edaftertab##1##2{\rightltab{##1}{##2}}%
7092   \measuretbody{#1}%
7093   \l@drestorefills
7094   \variab
7095   \setmrowleft #1\\&\\%
7096   \enablel@dtabfeet}
7097
7098 %
```

`\ltabtext` Tabular with entries left justified.

```

7099 \newcommand{\ltabtext}[1]{%
7100   \l@dnnullfills
7101   \measuretbody{#1}%
7102   \l@drestorefills
7103   \variab
7104   \settrorleft #1\&\%
7105   \enablel@dtabfeet}
7106
7107 %

```

`\ctab` Array with centered entries.

```

\edbeforetab
\edaftertab
7108 \newcommand{\ctab}[1]{%
7109   \l@dnnullfills
7110   \def\edbeforetab##1##2{\leftctab{##1}{##2}}%
7111   \def\edaftertab##1##2{\rightctab{##1}{##2}}%
7112   \measuretbody{#1}%
7113   \l@drestorefills
7114   \variab
7115   \setmrowcenter #1\&\%
7116   \enablel@dtabfeet}
7117
7118 %

```

`\ctabtext` Tabular with entries centered.

```

7119 \newcommand{\ctabtext}[1]{%
7120   \l@dnnullfills
7121   \measuretbody{#1}%
7122   \l@drestorefills
7123   \variab
7124   \settrorcenter #1\&\%
7125   \enablel@dtabfeet}
7126
7127 %

```

```

\spreadtext28 \newcommand{\spreadtext}[1]{%\l@dcolcount=\l@dampcount%
7129   \hb@xt@ \the\l@dcolwidth{\hbox{#1}\hss}}
7130 %

```

```

\spreadmath31 \newcommand{\spreadmath}[1]{%
7132   \hb@xt@ \the\l@dcolwidth{\hbox{$\displaystyle{#1}$}\hss}}
7133
7134 %

```

`\HILFSskip` More helpers.

`\Hilfsskip`

```

7135 \newskip\HILFSskip
7136 \newskip\Hilfsskip
7137
7138 %

```

```

\EDTABINDENT \newcommand{\EDTABINDENT}{%
7139
7140   \ifnum\l@dcolcount=30\let\NEXT\relax\l@dcolcount=0%
7141   \else\step\l@dcolcount%
7142     \advance\Hilfsskip by\l@dcolwidth%
7143     \ifdim\l@dcolwidth=0pt\advance\hilfscount\@ne
7144     \else\advance\Hilfsskip by \the\hilfscount\edtabcolsep%
7145     \hilfscount=1\fi%
7146     \let\NEXT=\EDTABINDENT%
7147   \fi\NEXT}%
7148 %

```

\edtabindent (was \tabindent)

```

7149 \newcommand{\edtabindent}{%
7150   \l@dcolcount=0\relax
7151   \Hilfsskip=0pt%
7152   \hilfscount=1\relax
7153   \EDTABINDENT%
7154   \hilfsskip=\hsize%
7155   \advance\hilfsskip -\Hilfsskip%
7156   \Hilfsskip=0.5\hilfsskip%
7157   }%
7158
7159 %

```

\EDTAB (was \TAB)

```

7160 \def\EDTAB #1|#2|{%
7161   \setbox\tabhilfbox=\hbox{$\displaystyle{#1}$}%
7162   \setbox\tabHilfbox=\hbox{$\displaystyle{#2}$}%
7163   \advance\tabelskip -\wd\tabhilfbox%
7164   \advance\tabelskip -\wd\tabHilfbox%
7165   \unhbox\tabhilfbox\hskip\tabelskip%
7166   \unhbox\tabHilfbox}%
7167
7168 %

```

\EDTABtext (was \TABtext)

```

7169 \def\EDTABtext #1|#2|{%
7170   \setbox\tabhilfbox=\hbox{#1}%
7171   \setbox\tabHilfbox=\hbox{#2}%
7172   \advance\tabelskip -\wd\tabhilfbox%
7173   \advance\tabelskip -\wd\tabHilfbox%

```

```

7174 \unhbox\tabhilfbox\hskip\tabelskip%
7175 \unhbox\tabHilfbox}%
7176 %

```

`\tabhilfbox` Further helpers.

```

\tabHilfbox
7177 \newbox\tabhilfbox
7178 \newbox\tabHilfbox
7179
7180 %

```

XXIX.2.4 Environments

`edarrayl edarrayc edarrayr` The ‘environment’ forms for `\ltab`, `\ctab` and `\rtab`.

```

7181 \newenvironment{edarrayl}{\l@collect@body\ltab}{}
7182 \newenvironment{edarrayc}{\l@collect@body\ctab}{}
7183 \newenvironment{edarrayr}{\l@collect@body\rtab}{}
7184
7185 %

```

`edtabularl edtabularc edtabularr` The ‘environment’ forms for `\ltabtext`, `\ctabtext` and `\rtabtext`.

```

7186 \newenvironment{edtabularl}{\l@collect@body\ltabtext}{}
7187 \newenvironment{edtabularc}{\l@collect@body\ctabtext}{}
7188 \newenvironment{edtabularr}{\l@collect@body\rtabtext}{}
7189
7190 %

```

XXX Quotation’s commands

`\initnumbering@quote` This macro, called at the beginning of any numbered section, locally redefines the quotation and quote environments, in order to allow their use inside of numbered sections.

`\quotation` `\initnumbering@quote` defines quotation environment.

```

\endquotation
\quote
\endquote
7191 \newcommand{\initnumbering@quote}{
7192   \ifnoquotation@else
7193   \renewcommand{\quotation}{\par\leavevmode%
7194     \parindent=1.5em%
7195     \skipnumbering%
7196     \ifautopar%
7197       \vskip-\parskip%
7198     \else%
7199       \vskip\topsep%
7200     \fi%
7201     \global\leftskip=\leftmargin%

```

```

7202             \global\rightskip=\leftmargin%
7203         }
7204     \renewcommand{\endquotation}{\par%
7205         \global\leftskip=0pt%
7206         \global\rightskip=0pt%
7207         \leavevmode%
7208         \skipnumbering%
7209         \ifautopar%
7210             \vskip-\parskip%
7211         \else%
7212             \vskip\topsep%
7213         \fi%
7214     }
7215     \renewcommand{\quote}{\par\leavevmode%
7216         \parindent=0pt%
7217         \skipnumbering%
7218         \ifautopar%
7219             \vskip-\parskip%
7220         \else%
7221             \vskip\topsep%
7222         \fi%
7223         \global\leftskip=\leftmargin%
7224         \global\rightskip=\leftmargin%
7225     }
7226     \renewcommand{\endquote}{\par%
7227         \global\leftskip=0pt%
7228         \global\rightskip=0pt%
7229         \leavevmode%
7230         \skipnumbering%
7231         \ifautopar%
7232             \vskip-\parskip%
7233         \else%
7234             \vskip\topsep%
7235         \fi%
7236     }
7237     \fi
7238 }
7239 %

```

XXXI Section's title commands

XXXI.1 Commands to disable some feature

\ledsectnotoc The `\ledsectnotoc` only disables the `\addcontentsline` macro.

```

7240 \newcommand{\ledsectnotoc}{\let\addcontentsline\@gobblethree}
7241 %

```

`\ledsectnomark` The `\ledsectnomark` only disables the `\chaptermark`, `\sectionmark` and `\subsectionmark` macros.

```

7242 \newcommand{\ledsectnomark}{%
7243   \let\chaptermark@gobble%
7244   \let\sectionmark@gobble%
7245   \let\subsectionmark@gobble%
7246 }
7247 %

```

XXXI.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 `ℒTeX` 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 `ℒTeX` sectioning command.
4. This last command is called when the `pstart` is effectively printed.

XXXI.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).

```

7248 \catcode`\#=12
7249 \notbool{@noeled@sec}{%
7250 %

```

`\beforeeledchapter` For technical reasons, not yet solved, page-breaking before chapters can't be made automatically by `eledmac`. Users have to use `\beforeeledchapter`.

```

7251 \ifl@dmemoir
7252   \newcommand\beforeeledchapter{%
7253     \clearforchapter%
7254   }
7255 \else
7256   \newcommand\beforeeledchapter{%
7257     \if@openright%
7258       \cleardoublepage%
7259     \else%
7260       \clearpage%
7261     \fi%
7262   }
7263 \fi
7264 %

```

XXXI.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.

```

7265 \newif\if@eled@sectioning
7266 %

```

`\print@leftmargin@eledsection` `\print@rightmargin@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.

```

7267 \def\print@rightmargin@eledsection{%
7268   \if@eled@sectioning%
7269     \begingroup%
7270     \if@RTL%
7271       \let\llap\rlap%
7272       \let\leftlinenum\rightlinenum%
7273       \let\leftlinenumR\rightlinenumR%
7274       \let\l@drd@ta\l@dld@ta%
7275       \let\l@drsn@te\l@dlsn@te%
7276     \fi%
7277     \hfill\l@drd@ta \csuse{LR}{\l@drsn@te}%
7278     \endgroup%
7279   \fi%
7280 }%
7281
7282 \def\print@leftmargin@eledsection{%
7283   \if@eled@sectioning%
7284     \leavevmode%
7285     \begingroup%

```



```

7286 \if@RTL%
7287 \let\rlap\llap%
7288 \let\rightlinenum\leftlinenum%
7289 \let\rightlinenumR\leftlinenumR%
7290 \let\l@dld@ta\l@dld@ta%
7291 \let\l@dlsn@te\l@dlsn@te%
7292 \fi%
7293 \l@dld@ta\csuse{LR}{\l@dlsn@te}%
7294 \endgroup%
7295 \fi%
7296 }%
7297
7298 %

```

XXXI.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.

```

7299 \AtBeginDocument{%
7300 \patchcmd{\chapter}{\clearforchapter}{%
7301 \if@eled@sectioning\else%
7302 \ifl@printingpages\else%
7303 \clearforchapter%
7304 \fi%
7305 \fi%
7306 }
7307 {}
7308 {}
7309
7310
7311 \pretocmd{\M@sect}
7312 {\let\old@edtext=\edtext%
7313 \let\edtext=\dummy@edtext@showlemma%
7314 }
7315 {}
7316 {}
7317
7318 \apptocmd{\M@sect}
7319 {\let\edtext=\old@edtext}
7320 {}
7321 {}

```

```

7322 \patchcmd{\M@sect}
7323 { #9}
7324 { #9%
7325 \print@rightmargin@eledsection%
7326 }
7327 {}
7328 {}
7329 {}
7330
7331 \patchcmd{\M@sect}
7332 {\hskip #3\relax}
7333 {\hskip #3\relax%
7334 \print@leftmargin@eledsection%
7335 }
7336 {}
7337 {}
7338
7339 \patchcmd{\@mem@old@ssect}
7340 {#5}
7341 {#5%
7342 \print@leftmargin@eledsection%
7343 }
7344 {}
7345 {}
7346
7347 \patchcmd{\@mem@old@ssect}
7348 {\hskip #1}
7349 {\hskip #1%
7350 \print@rightmargin@eledsection%
7351 }
7352 {}
7353 {}
7354
7355 \patchcmd{\chapter}{\if@openright\cleardoublepage\else\clearpage\fi}{%
7356 \if@eled@sectioning\else%
7357 \ifl@dprintingpages\else%
7358 \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
7359 \fi%
7360 \fi%
7361 }%
7362 {}%
7363 {}%
7364
7365 \patchcmd{\scr@startchapter}{\if@openright\cleardoublepage\else\clearpage\
fi}{%
7366 \if@eled@sectioning\else%
7367 \ifl@dprintingpages\else%
7368 \if@openright\cleardoublepage\else\clearpage\fi%No clearpage inside a

```

```

\Pages: will keep critical notes from printing on the title page. Here for
scrbook.
    \fi%
7369 \fi%
7370 \fi%
7371 }
7372 {}
7373 {}
7374
7375 \patchcmd{\@makechapterhead}
7376 {#1}
7377 {\print@leftmargin@eledsection%
7378   #1%
7379 \print@rightmargin@eledsection%
7380 }
7381 {}
7382 {}
7383
7384 \patchcmd{\@makechapterhead}% For BIDI
7385 {\if@RTL\raggedleft\else\raggedright\fi}%
7386 {\if@eled@sectioning\else%
7387   \if@RTL\raggedleft\else\raggedright\fi%
7388 \fi%
7389 }%
7390 {}%
7391 {}%
7392
7393 \patchcmd{\@makeschapterhead}
7394 {#1}
7395 {\print@leftmargin@eledsection%
7396   #1%
7397 \print@rightmargin@eledsection%
7398 }
7399 {}
7400 {}
7401
7402 \pretocmd{\@sect}
7403 {\let\old@edtext=\edtext
7404 \let\edtext=\dummy@edtext@showlemma%
7405 }
7406 {}
7407 {}
7408
7409 \apptocmd{\@sect}
7410 {\let\edtext=\old@edtext}
7411 {}
7412 {}
7413
7414 \pretocmd{\@ssect}
7415 {\let\old@edtext=\edtext%
7416 \let\edtext=\dummy@edtext@showlemma%

```

```

7417 }
7418 {}
7419 {}
7420
7421 \apptocmd{\@ssect}
7422 {\let\edtext=\old@edtext}
7423 {}
7424 {}
7425
7426 %

```

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.

```

7427 \@ifpackageloaded{nameref}{
7428
7429   \patchcmd{\NR@sect}
7430     {#8}
7431     {#8%
7432       \print@rightmargin@eledsection%
7433     }
7434     {}
7435     {}
7436
7437   \patchcmd{\NR@sect}
7438     {\hskip #3\relax}
7439     {\hskip #3\relax%
7440       \print@leftmargin@eledsection%
7441     }
7442     {}
7443     {}
7444
7445   \patchcmd{\NR@ssect}
7446     {#5}
7447     {#5%
7448       \print@rightmargin@eledsection%
7449     }
7450     {}
7451     {}
7452
7453   \patchcmd{\NR@ssect}
7454     {\hskip #1}
7455     {\hskip #1%
7456       \print@leftmargin@eledsection%
7457     }
7458     {}
7459     {}
7460   }%
7461   {
7462     \patchcmd{\@sect}
7463       {#8}

```

```

7464     {#8%
7465     \print@rightmargin@eledsection%
7466     }
7467     {}
7468     {}
7469
7470     \patchcmd{\@sect}
7471     {\hskip #3\relax}
7472     {\hskip #3\relax%
7473     \print@leftmargin@eledsection%
7474     }
7475     {}
7476     {}
7477
7478     \patchcmd{\@ssect}
7479     {#5}
7480     {#5%
7481     \print@rightmargin@eledsection%
7482     }
7483     {}
7484     {}
7485
7486     \patchcmd{\@ssect}
7487     {\hskip #1}
7488     {\hskip #1%
7489     \print@leftmargin@eledsection%
7490     }
7491     {}
7492     {}
7493     }%
7494 }
7495 %

```

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.

```

7496 {}}%
7497 \protect\catcode`\#=6 %Space NEEDS by \catcode
7498 \notbool{@noeled@sec}{%
7499 %

```

XXXI.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.

```

7500 \newwrite\eled@sectioning@out
7501 %

```

`\eledchapter` And now, the user sectioning commands, which write to the file, and also add content as a “normal” line.

`\eledsection`

`\eledsubsection`

```

7502 \newcommand{\eledchapter}[2] [] {%
7503   #2%
7504   \ifledRcol%
7505     \immediate\write\eled@sectioningR@out{%
7506       \string\eled@chapter{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR}{R}
7507     }%
7508   \else%
7509     \immediate\write\eled@sectioning@out{%
7510       \string\eled@chapter{#1}{\unexpanded{#2}}{\the\l@dnumpstartsL}{L}
7511     }%
7512   \fi%
7513 }
7514
7515 \newcommand{\eledsection}[2] [] {%
7516   #2%
7517   \ifledRcol%
7518     \immediate\write\eled@sectioningR@out{%
7519       \string\eled@section{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR}{R}
7520     }%
7521   \else%
7522     \immediate\write\eled@sectioning@out{%
7523       \string\eled@section{#1}{\unexpanded{#2}}{\the\l@dnumpstartsL}{L}
7524     }%
7525   \fi%
7526 }
7527
7528 \newcommand{\eledsubsection}[2] [] {%
7529   #2%
7530   \ifledRcol%
7531     \immediate\write\eled@sectioningR@out{%
7532       \string\eled@subsection{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR}{R}
7533     }%
7534   \else%
7535     \immediate\write\eled@sectioning@out{%
7536       \string\eled@subsection{#1}{\unexpanded{#2}}{\the\l@dnumpstartsL}{L}
7537     }%
7538   \fi%
7539 }
7540 \newcommand{\eledsubsubsection}[2] [] {%
7541   #2%
7542   \ifledRcol%
7543     \immediate\write\eled@sectioningR@out{%
7544       \string\eled@subsubsection{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR}
7545     }%
7546   \else%

```

```

7547 \immediate\write\eled@sectioning@out{%
7548 \string\eled@subsubsection{#1}{\unexpanded{#2}}{\the\l@dumpstartsL
7549 }{}{}
7549 }%
7550 \fi%
7551 }
7552
7553
7554 \WithSuffix\newcommand\eledchapter*[2][]{%
7555 #2%
7556 \ifledRcol%
7557 \immediate\write\eled@sectioningR@out{%
7558 \string\eled@chapter{#1}{\unexpanded{#2}}{\the\l@dumpstartsR}{*}{R}
7559 }%
7560 \else%
7561 \immediate\write\eled@sectioning@out{%
7562 \string\eled@chapter{#1}{\unexpanded{#2}}{\the\l@dumpstartsL}{*}{L}
7563 }%
7564 \fi%
7565 }
7566
7567 \WithSuffix\newcommand\eledsection*[2][]{%
7568 #2%
7569 \ifledRcol%
7570 \immediate\write\eled@sectioningR@out{%
7571 \string\eled@section{#1}{\unexpanded{#2}}{\the\l@dumpstartsR}{*}{R}
7572 }%
7573 \else%
7574 \immediate\write\eled@sectioning@out{%
7575 \string\eled@section{#1}{\unexpanded{#2}}{\the\l@dumpstartsL}{*}{L}
7576 }%
7577 \fi%
7578 }
7579
7580 \WithSuffix\newcommand\eledsubsection*[2][]{%
7581 #2%
7582 \ifledRcol%
7583 \immediate\write\eled@sectioningR@out{%
7584 \string\eled@subsection{#1}{\unexpanded{#2}}{\the\l@dumpstartsR}{*}{R}
7585 }%
7586 \else%
7587 \immediate\write\eled@sectioning@out{%
7588 \string\eled@subsection{#1}{\unexpanded{#2}}{\the\l@dumpstartsL}
7589 }{*}{L}
7589 }%
7590 \fi%
7591 }
7592
7593 \WithSuffix\newcommand\eledsubsubsection*[2][]{%

```

```

7594 #2%
7595 \ifledRcol%
7596 \immediate\write\eled@sectioningR@out{%
7597 \string\eled@subsubsection{#1}{\unexpanded{#2}}{\the\l@dumpstartsR
7598 }{*}{R}
7599 }%
7600 \else%
7601 \immediate\write\eled@sectioning@out{%
7602 \string\eled@subsubsection{#1}{\unexpanded{#2}}{\the\l@dumpstartsL
7603 }{*}{L}
7604 }%
7605 \fi%

```

XXXI.7 Macros written in the auxiliary file

`\eled@chapter`
`\eled@section`
`\eled@subsection`
`\eled@subsubsection`

The sectioning macros, called in the auxiliary file. They have five arguments:

1. Optional arguments of \LaTeX sectioning command.
2. Mandatory arguments of \LaTeX sectioning command.
3. Pstart number.
4. Side: R if right, nothing if left.
5. Starred or not.

```

7606 \def\eled@chapter#1#2#3#4#5{%
7607 \ifstrempy{#4}%
7608 {%
7609 \ifstrempy{#1}%
7610 {%
7611 \global\csdef{eled@sectioning@#3#5}{\let\edtext=\
dummy@edtext@showlemma\chapter{#2}}%
7612 \global\csdef{eled@sectmark@#3#5}{\let\edtext=\dummy@edtext{}\
chaptermark{#2}}%
7613 }%Need for \pairs, because of using parbox.
7614 {%
7615 \global\csdef{eled@sectioning@#3#5}{\let\edtext=\
dummy@edtext@showlemma\chapter[#1]{#2}}%
7616 \global\csdef{eled@sectmark@#3#5}{\let\edtext=\dummy@edtext{}\
chaptermark{#2}}%Need for \pairs, because of using parbox.
7617 }%
7618 }%
7619 {%
7620 \ifstrempy{#1}%
7621 {\global\csdef{eled@sectioning@#3#5}{\let\edtext=\
dummy@edtext@showlemma\chapter*{#2}}}%

```



```

7622     {\global\csdef{eled@sectioning@#3#5}{\let\edtext=\
dummy@edtext@showlemma\chapter*{#1}{#2}}}%Bug in LaTeX!
7623     }%
7624     \listcsadd{eled@sections#5@@}{#3}%
7625     }
7626 \def\eled@section#1#2#3#4#5{%
7627     \ifstrepty{#4}%
7628     {\ifstrepty{#1}%
7629     {%
7630     \global\csdef{eled@sectioning@#3#5}{\section{#2}}%
7631     \global\csdef{eled@sectmark@#3#5}{\let\edtext=\dummy@edtext{}\
sectionmark{#2}}}%Need for \pairs, because of using parbox.
7632     }%
7633     {%
7634     \global\csdef{eled@sectioning@#3#5}{\section[#1]{#2}}%
7635     \global\csdef{eled@sectmark@#3#5}{\let\edtext=\dummy@edtext{}\
sectionmark{#1}}}%Need for \pairs, because of using parbox.
7636     }%
7637     }%
7638     {\ifstrepty{#1}%
7639     {\global\csdef{eled@sectioning@#3#5}{\section*{#2}}}%
7640     {\global\csdef{eled@sectioning@#3#5}{\section*{#1}{#2}}}%Bug in
LaTeX!
7641     }
7642     \listcsadd{eled@sections#5@@}{#3}%
7643     }
7644 \def\eled@subsection#1#2#3#4#5{%
7645     \ifstrepty{#4}%
7646     {\ifstrepty{#1}%
7647     {%
7648     \global\csdef{eled@sectioning@#3#5}{\subsection{#2}}%
7649     \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)
7650     }%
7651     {%
7652     \global\csdef{eled@sectioning@#3#5}{\subsection[#1]{#2}}%
7653     \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)
7654     }%
7655     }%
7656     {\ifstrepty{#1}%
7657     {\global\csdef{eled@sectioning@#3#5}{\subsection*{#2}}}%
7658     {\global\csdef{eled@sectioning@#3#5}{\subsection*{#1}{#2}}}%Bug in
LaTeX!
7659     }
7660     \listcsadd{eled@sections#5@@}{#3}%
7661     }
7662 \def\eled@subsubsection#1#2#3#4#5{%

```

```

7663 \ifstrempy{#4}%
7664   {\ifstrempy{#1}%
7665     {\global\csdef{eled@sectioning@#3#5}{\subsubsection{#2}}}%
7666     {\global\csdef{eled@sectioning@#3#5}{\subsubsection{#1}{#2}}}%
7667   }%
7668   {\ifstrempy{#1}%
7669     {\global\csdef{eled@sectioning@#3#5}{\subsubsection*{#2}}}%
7670     {\global\csdef{eled@sectioning@#3#5}{\subsubsection*{#1}{#2}}}%Bug
in LaTeX!
7671   }
7672   \listcsgadd{eled@sections#5@@}{#3}%
7673   }
7674
7675 %

```

End of the conditional test about noeledsec option.

```

7676 }{}
7677 %

```

XXXII 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.

```

7678 \def\normal@page@break{}
7679 %

```

\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.

```

7680 \def\l@prev@pb{}
7681 \def\l@prev@nopb{}
7682 %

```

\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.

```

7683 \newcommand{\ledpb}{\write\linenum@out{\string\led@pb}}
7684 \newcommand{\ledpbnum}[1]{\write\linenum@out{\string\led@pbnum{#1}}}
7685 \newcommand{\lednopb}{\write\linenum@out{\string\led@nopb}}
7686 \newcommand{\lednopbnum}[1]{\write\linenum@out{\string\led@nopbnum{#1}}}
7687 %

```

\led@pb The \led@pb adds the absolute line number in the \prev@pb list. The \led@pbnum adds the argument in the \prev@pb list. The \led@nopb adds the absolute line number in the \prev@nopb list. The \led@nopbnum adds the argument in the \prev@nopb list.

```

\led@pbnum \newcommand{\led@pb}{\listxadd{\l@prev@pb}{\the\absline@num}}
7688 \newcommand{\led@pbnum}[1]{\listxadd{\l@prev@pb}{#1}}
7689 \newcommand{\led@nopb}{\listxadd{\l@prev@nopb}{\the\absline@num}}
7690 \newcommand{\led@nopbnum}[1]{\listxadd{\l@prev@nopb}{#1}}
7691 %
7692

```

\ledpbsetting The \ledpbsetting macro only changes the value of \led@pb@macro, for which the default value is before.

```

\led@pb@setting \def\led@pb@setting{before}
7693 \newcommand{\ledpbsetting}[1]{\gdef\led@pb@setting{#1}}
7694 %
7695

```

\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.

```

7696 \newcommand{\led@check@pb}{\xifinlist{\the\absline@num}{\l@prev@pb}{\pagebreak[4]}}
7697 \newcommand{\led@check@nopb}{%
7698   \IfStrEq{\led@pb@setting}{before}{%
7699     \xifinlist{\the\absline@num}{\l@prev@nopb}{%
7700       {\numdef{\abs@prevline}{\the\absline@num-1}%
7701       \xifinlist{\abs@prevline}{\normal@page@break}%
7702       {\nopagebreak[4]\enlargethispage{\baselineskip}}%
7703       }%
7704     }%
7705   }%
7706 }%
7707 \IfStrEq{\led@pb@setting}{after}{%
7708   \xifinlist{\the\absline@num}{\l@prev@nopb}{%
7709     \xifinlist{\the\absline@num}{\normal@page@break}%
7710     {\nopagebreak[4]\enlargethispage{\baselineskip}}%
7711     }%
7712 }%
7713 }%
7714 }%
7715 }%
7716 }
7717 %

```

XXXIII 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.

```

7718 \newcommand{\check@pb@in@verse}{%
7719   \ifinstanza\iflednopbinverse\ifinserthangingsymbol% Using stanzas and
enabling page breaks in verse control, while on a hanging verse.
7720   \ifnum\page@num=\last@page@num\else%If we have change page
7721     \IfStrEq{\led@pb@setting}{before}{%
7722       \numgdef{\abs@line@verse}{\the\absline@num-1}%
7723       \ledpbnum{\abs@line@verse}%
7724     }{}%
7725     \IfStrEq{\led@pb@setting}{after}{%
7726       \numgdef{\abs@line@verse}{\the\absline@num-1}%
7727       \lednopbnum{\abs@line@verse}%
7728     }{}%
7729   \fi%
7730 \fi\fi\fi%
7731 }
7732 %

```

XXXIV 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

```

7733 \def\Hy@raisedlink@left#1{%
7734   \ifvmode
7735     #1%
7736   \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-coupeure-de-ligne/801>.

```

7737 \Hy@SaveSpaceFactor
7738 \llap{\smash{%
7739 \begingroup
7740 \let\HyperRaiseLinkLength\@tempdima
7741 \setlength\HyperRaiseLinkLength\HyperRaiseLinkDefault
7742 \HyperRaiseLinkHook
7743 \expandafter\endgroup
7744 \expandafter\raise\the\HyperRaiseLinkLength\hbox{%
7745 \Hy@RestoreSpaceFactor
7746 #1%
7747 \Hy@SaveSpaceFactor
7748 }%
7749 }}%
7750 \Hy@RestoreSpaceFactor
7751 \penalty\@M\hskip\z@
7752 \fi
7753 }
7754 %

```

XXXV Compatibility with eledmac

Here, we define some commands for the eledmac-compat option.

```

7755 \ifeledmaccompat{%
7756
7757 \newcommand{\footnormalX}[1]{\arrangementX[#1]{normal}}%
7758 \newcommand{\footparagraphX}[1]{\arrangementX[#1]{paragraph}}%
7759 \newcommand{\foottwocolX}[1]{\arrangementX[#1]{twocol}}%
7760 \newcommand{\footthreecolX}[1]{\XarrangementX[#1]{threecol}}%
7761
7762 \unless\ifnocritical@
7763 \newcommand{\footnormal}[1]{\Xarrangement[#1]{normal}}%
7764 \newcommand{\footparagraph}[1]{\Xarrangement[#1]{paragraph}}%
7765 \newcommand{\foottwocol}[1]{\Xarrangement[#1]{twocol}}%
7766 \newcommand{\footthreecol}[1]{\Xarrangement[#1]{threecol}}%
7767 \let\hsizetwocol\Xhsizetwocol
7768 \let\hsizethreecol\Xhsizethreecol
7769 \let\bhookXnote\Xbhooknote
7770 \let\boxsymlinenum\Xboxsymlinenum
7771 \let\symlinenum\Xsymlinenum
7772 \let\beforenumberinfootnote\Xbeforenumber
7773 \let\afternumberinfootnote\Xafternumber
7774 \let\beforeXsymlinenum\XbeforeXsymlinenum
7775 \let\afterXsymlinenum\XafterXsymlinenum
7776 \let\inplaceofnumber\Xinplaceofnumber
7777 \let\Xlemmaseparator\lemmaseparator
7778 \let\afterlemmaseparator\Xafterlemmaseparator
7779 \let\beforelemmaseparator\Xbeforelemmaseparator
7780 \let\inplaceoflemmaseparator\Xinplaceoflemmaseparator

```

```

7781 \let\txtbeforeXnotes\Xtxtbeforenotes
7782 \let\afterXrule\Xafterrule
7783 \let\numberonlyfirstinline\Xnumberonlyfirstinline
7784 \let\numberonlyfirstintwlines\Xnumberonlyfirstintwlines
7785 \let\nonumberinfootnote\Xnonumberinfootnote
7786 \let\pstartinfootnote\Xpstart
7787 \let\pstartinfootnoteeverytime\Xpstarteverytime
7788 \let\onlyXpstart\Xonlypstart
7789 \let\Xnonumberinfootnote\Xnonumber
7790 \let\nonbreakableafternumber\Xnonbreakableafternumber
7791 \let\maxhXnotes\Xmaxhnotes
7792 \let\beforeXnotes\Xbeforenotes
7793 \let\boxlinenum\Xboxlinenum
7794 \let\boxlinenumalign\Xboxlinenumalign
7795 \let\boxstartlinenum\Xboxstartlinenum
7796 \let\boxendlinenum\Xboxendlinenum
7797 \let\twlines\Xtwlines
7798 \let\morethantwlines\Xmorethantwlines
7799 \let\twlinesbutnotmore\Xtwlinesbutnotmore
7800 \let\twlinesonlyinsamepage\Xtwlinesonlyinsamepage
7801 \fi
7802
7803 \unless\ifnofamiliar@
7804 \let\notesXwidthliketwocolumns\noteswidthliketwocolumnsX
7805 \fi
7806 \newcommandx{\parafootsep}[2][1,usedefault]{%
7807 \Xparafootsep[#1]{#2}%
7808 \parafootsepX[#1]{#2}
7809 }%
7810
7811 \newcommandx{\afternote}[2][1,usedefault]{%
7812 \Xafternote[#1]{#2}%
7813 \afternoteX[#1]{#2}%
7814 }%
7815
7816 \unless\ifnoend@
7817 \let\XendXtwlines\Xendtwlines
7818 \let\XendXmorethantwlines\Xendmorethantwlines
7819 \let\hookXendnote\Xendhooknote
7820 \let\boxXendlinenum\Xendboxlinenum%
7821 \let\boxXendlinenumalign\Xendboxlinenumalign%
7822 \let\boxXendstartlinenum\Xendboxstartlinenum%
7823 \let\boxXendendlinenum\Xendboxendlinenum%
7824 \let\XendXlemmaseparator\Xendlemmaseparator
7825 \let\XendXbeforelemmaseparator\Xendbeforelemmaseparator
7826 \let\XendXafterlemmaseparator\Xendafterlemmaseparator
7827 \let\XendXinplaceoflemmaseparator\Xendinplaceoflemmaseparator
7828 \fi
7829
7830 \AtBeginDocument{%

```

```
7831 \ifdef\lineref{}\let\lineref\edlineref}%  
7832 }%  
7833  
7834  
7835 \fi%  
7836 %
```

</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 \LaTeX 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. 108 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 5.5.1 p. 29), 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 (6 p. 30) or `\Xfootnote` options (5.2.2 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 `stanzaindentrepetition` (cf. 8.3 p. 42). This bug had two consequences:

1. `stanzaindentrepetition` did not work when its value was greater than 2.
2. `stanzaindentrepetition` worked wrong when its value was equal to 2.

So, if you used `stanzaindentrepetition` with a value equal to 2, you had to change your `\setstanzaindent`. Explanation:

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

to:

```
\setcounter{stanzaindentrepetition}{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 4.2.3 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 15.2 p. 57.

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 18.1.2 p. 60 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 `\Xlinerangeseparator` or `/` and `\Xendlinerangeseparator` (6.2.3 p. 31).

Appendix A.14 Migration to reledmac 2.5.0

It is strongly recommended to stop redefining `\printnpnum` and to use the hooks documented in 6.3 p. 35.

`\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 `edstanzas.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>	41
<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>\@fnpos</code>	1

<code>\@footnotemark</code>	1
<code>\@footnotetext</code>	1
<code>\@getfirstseries</code>	1
<code>\@gobblefive</code>	1
<code>\@gobblefour</code>	1
<code>\@gobblethree</code>	1
<code>\@h</code>	1
<code>\@hangingsymbol</code>	1
<code>\@iiiminipage</code>	1
<code>\@insertstanza</code>	1
<code>\@k</code>	1
<code>\@l@dtmptcnta</code>	1
<code>\@l@dtmptcntb</code>	1
<code>\@lab</code>	1
<code>\@led@testifnofoot</code>	1
<code>\@lemma</code>	1
<code>\@line@#num</code>	1
<code>\@lock</code>	1
<code>\@lopL</code>	1
<code>\@lopR</code>	1
<code>\@makechapterhead</code>	1
<code>\@makeschapterhead</code>	1
<code>\@mem@extranofeet</code>	1
<code>\@mem@old@ssect</code>	1
<code>\@mpfnpos</code>	1
<code>\@nl</code>	1
<code>\@nl@reg</code>	1
<code>\@opXfeet</code>	1
<code>\@pend</code>	1
<code>\@pendR</code>	1
<code>\@ref</code>	1
<code>\@ref@reg</code>	1
<code>\@sect</code>	1
<code>\@series</code>	1
<code>\@set</code>	1
<code>\@sidenotesep</code>	1
<code>\@ssect</code>	1
<code>\@startstanza</code>	1
<code>\@stopstanza</code>	1
<code>\@sw</code>	1
<code>\@tag</code>	1
<code>\@wredindex</code>	1
<code>\@xloop</code>	1
<code>\@xympar</code>	1
<code>CLASSbook</code>	297
<code>CLASSmemoir</code>	174, 175, 228, 229, 258, 297, 316, 370, 374
<code>CLASSscrbook</code>	374
<code>COMMAND*footnote</code>	61
<code>COMMAND\...@footnotemark...</code>	178
<code>COMMAND\...d@ta</code>	131

COMMAND\<hook	
@<series	217
COMMAND\<hookname	
<pseudoseries	219, 220
COMMAND\<type	
footfmt	166
COMMAND\@@line	158
COMMAND\@MM	146, 371
COMMAND\@Rlineflag	260, 371
COMMAND\@Serefprefix	240
COMMAND\@Serefprefixmore	240
COMMAND\@add@	287
COMMAND\@adv	95
COMMAND\@apprefprefixmore	240
COMMAND\@apprefprefixsingle	240
COMMAND\@bspback	231
COMMAND\@doclearpage	229, 364, 374
COMMAND\@doreinfeetX	374
COMMAND\@dprintingcolumns	371
COMMAND\@edindex@hyperref	260, 261
COMMAND\@edtext@level	112
COMMAND\@esphack	231
COMMAND\@fnpos	195
COMMAND\@footnotemark	175, 364, 374
COMMAND\@footnotetext	175, 176, 364
COMMAND\@gobble	110
COMMAND\@gobblefive	214, 372
COMMAND\@gobblefour	370
COMMAND\@gobblethree	363
COMMAND\@h	161
COMMAND\@hangingsymbol	265
COMMAND\@iiiminipage	250, 252, 363, 374
COMMAND\@iiiminipage	250
COMMAND\@l	369
COMMAND\@l@tempcnta	134, 135, 142
COMMAND\@l@tempcntb	135
COMMAND\@l@reg	369
COMMAND\@lab	92, 231, 233, 237, 363
COMMAND\@ldunboxmpfoot	252
COMMAND\@led@extranofeet	317
COMMAND\@ledinnote@command	256, 257
COMMAND\@lemma	114, 116
COMMAND\@lock	86, 265
COMMAND\@lopL	364
COMMAND\@lopR	364
COMMAND\@makecol	225, 226, 228, 374
COMMAND\@mpfnpos	195
COMMAND\@nl	92–95, 97, 104, 233, 363, 364
COMMAND\@nl@reg	93, 314, 364, 369
COMMAND\@opXfeet	364

COMMAND\@opfeetX	374
COMMAND\@opxtrafeeti	374
COMMAND\@page	94, 233
COMMAND\@pend	364
COMMAND\@pendR	364
COMMAND\@ref	92, 101, 102, 105, 109
COMMAND\@ref@reg	101, 364
COMMAND\@reinserts	225–228, 374
COMMAND\@secondoftwo	62
COMMAND\@sect	300
COMMAND\@series	216
COMMAND\@set	96
COMMAND\@sidenotesepp	249
COMMAND\@sw	102, 117, 120, 121
COMMAND\@tag	111, 112, 116
COMMAND\@tempcnta	73
COMMAND\@tempcntb	73
COMMAND\@toksa	78
COMMAND\@toksb	78
COMMAND\@xloop	143
COMMAND\@xympar	244, 374
COMMAND\Aendnote	15, 23
COMMAND\Afootfmt	146
COMMAND\Afootgroup	146
COMMAND\Afootnote	8, 14, 22, 23, 25, 113, 153, 174, 196, 209, 373
COMMAND\Afootstart	146
COMMAND\AtEveryPend	17, 126, 370, 372, 374
COMMAND\AtEveryPstart	17, 370, 372, 374, 377
COMMAND\Bendnote	15, 22
COMMAND\Bfootnote	8, 14, 174, 196, 209
COMMAND\Centering	38
COMMAND\Cfootnote	174
COMMAND\Columns	73, 150
COMMAND\Dfootnote	174
COMMAND\Efootnote	174
COMMAND\Gls	53
COMMAND\Hy@raisedlink	308
COMMAND\Hy@raisedlink@left	308
COMMAND\NR	300
COMMAND\Pages	73, 226, 227
COMMAND\ProcessOptionsX	65
COMMAND\RaggedLeft	38
COMMAND\RaggedRight	38
COMMAND\SEonlypage	238, 376
COMMAND\Seref	47–49, 238, 240, 377
COMMAND\Serefonlypage	48, 49, 320, 376
COMMAND\Serefwithpage	47–49, 238, 240, 320, 376
COMMAND\Stanza	369
COMMAND\Waklam	287
COMMAND\X@doreinfeet	227, 374

COMMAND\XR@prefix	244
COMMAND\XR@test	243, 244
COMMAND\XR@test@mac	243
COMMAND\XR@test@mac@test	244
COMMAND\XXXXXXfmt	313
COMMAND\XXXXXfmt	313
COMMAND\Xafterlemmaseparator	35, 317
COMMAND\Xafternote	38, 316, 317
COMMAND\Xafternumber	33, 317
COMMAND\Xafterrule	39, 196, 317, 369, 372
COMMAND\Xaftersymlinenum	33, 317
COMMAND\Xarrangement	30, 61, 146, 147, 218, 317
COMMAND\Xarrangement@footparagraph	151
COMMAND\Xarrangement@normal	147
COMMAND\Xarrangement@paragraph	151
COMMAND\Xbeforelemmaseparator	35, 317
COMMAND\Xbeforenotes	39, 196, 317, 369, 372
COMMAND\Xbeforenumber	31, 33, 317
COMMAND\Xbeforesymlinenum	33, 317
COMMAND\Xbhookendnote	318
COMMAND\Xbhookgroup	39, 376, 377
COMMAND\Xbhooknote	37, 318, 319, 374, 375
COMMAND\Xboxendlinenum	34, 318, 373
COMMAND\Xboxlinenum	34, 318
COMMAND\Xboxlinenumalign	34, 318, 373
COMMAND\Xboxstartlinenum	34, 318, 373
COMMAND\Xboxsymlinenum	34, 318
COMMAND\Xcolalign	37, 372
COMMAND\Xdo@feet	226, 364, 374
COMMAND\Xend	214
COMMAND\XendXafterlemmaseparator	318
COMMAND\XendXbeforelemmaseparator	318
COMMAND\XendXinplaceoflemmaseparator	318
COMMAND\XendXlemmaseparator	318
COMMAND\XendXmorethantwolines	318
COMMAND\XendXtwolines	318
COMMAND\Xendafterenumber	33, 375
COMMAND\Xendafterlemmaseparator	36, 318
COMMAND\Xendafternote	40, 377
COMMAND\Xendafternumber	35
COMMAND\Xendafterpagenumber	35, 49
COMMAND\XendafterpagenumberSErefonlypage	320
COMMAND\Xendaftersymlinenum	33, 35, 375
COMMAND\Xendahookinplaceofnumber	35, 375
COMMAND\Xendahooklinenum	35, 375
COMMAND\Xendbeforelemmaseparator	36, 318
COMMAND\Xendbeforelinenum	35
COMMAND\Xendbeforenumber	33, 375
COMMAND\Xendbeforepagenumber	35, 48, 49
COMMAND\XendbeforepagenumberSErefonlypage	48

COMMAND\Xendbeforesymlinenum	33, 35, 375
COMMAND\Xendbhookinplaceofnumber	35, 375
COMMAND\Xendbhooklinenum	35, 375
COMMAND\Xendbhooknote	37
COMMAND\Xendboxendlinum	34, 373
COMMAND\Xendboxlinenum	34, 318, 371
COMMAND\Xendboxlinenumalign	34, 318, 373
COMMAND\Xendboxstartlinenum	34, 373
COMMAND\Xendboxsymlinenum	34, 375
COMMAND\Xendhangindent	37, 375, 377
COMMAND\Xendinplaceoflemmaseparator	23, 36, 318
COMMAND\Xendinplaceofnumber	34, 374
COMMAND\Xendinsertsep@	202
COMMAND\Xendlemmadisablefontselection	36
COMMAND\Xendlemmafont	36, 376
COMMAND\Xendlemmaseparator	24, 36, 318
COMMAND\Xendletboxendlinum	318
COMMAND\Xendlineflag	49
COMMAND\Xendlineprefixmore	35, 49
COMMAND\Xendlineprefixsingle	35, 49
COMMAND\Xendlinangeseparator	31, 49, 320, 375
COMMAND\Xendmorethantwolines	23, 32, 49, 318, 372, 373
COMMAND\Xendnonumber	32, 374
COMMAND\Xendnote	199, 213, 214, 372
COMMAND\Xendnotefontsize	37, 317
COMMAND\Xendnotenunfont	35, 36, 317
COMMAND\Xendnumberonlyfirstinline	31, 375
COMMAND\Xendnumberonlyfirstintwolines	31, 375
COMMAND\Xendparagraph	40, 369
COMMAND\Xendsep	40
COMMAND\Xendsublinesep	33, 49
COMMAND\Xendsymlinenum	31, 375
COMMAND\Xendtwolines	23, 32, 49, 318, 372, 373
COMMAND\Xendtwolinesbutnotmore	32, 49, 372, 373
COMMAND\Xendtwolinesonlyinsamepage	32, 49, 372, 373
COMMAND\Xfootfmt	319
COMMAND\Xfootgroup	150
COMMAND\Xfootins	149
COMMAND\Xfootnote	46, 52, 111, 313, 366, 370–372, 376
COMMAND\Xfootstarts	150
COMMAND\Xhangindent	37, 375
COMMAND\Xhsize	320, 376, 377
COMMAND\Xsizethreecol	38, 40, 318
COMMAND\Xsizetwocol	38, 40, 219, 318
COMMAND\Xinplaceoflemmaseparator	23, 35, 318
COMMAND\Xinplaceofnumber	34, 318, 371, 373
COMMAND\Xinsertparafootsep	156, 158
COMMAND\Xledsetnormalparstuff	315, 316, 372
COMMAND\Xlemmadisablefontselection	36
COMMAND\Xlemmafont	36, 376

COMMAND\Xlemmaseparator	35, 220, 223, 225, 318
COMMAND\Xlineflag	49
COMMAND\Xlinerangeseparator	31, 49, 320, 375
COMMAND\Xmaxhnotes	39, 61, 317, 318, 369, 371
COMMAND\Xmorethantwolines	23, 32, 49, 318, 371
COMMAND\Xnoindent	375
COMMAND\Xnolemmaseparator	35, 225, 318
COMMAND\Xnonbreakableafternumber	33, 318, 367
COMMAND\Xnonumber	32, 318
COMMAND\Xnonumberinfootnote	318
COMMAND\Xnotefontsize	37, 317
COMMAND\Xnotefontsize@<s>	156, 160, 161
COMMAND\Xnotenumberfont	36, 317
COMMAND\Xnoteswidthliketwocolumns	40, 370
COMMAND\Xnumberonlyfirstinline	31, 89, 219, 220, 223, 318, 366, 371
COMMAND\Xnumberonlyfirstintwolines	31, 318, 366
COMMAND\Xonlypstart	33, 318, 366, 371
COMMAND\Xparafootsep	38, 89, 317, 318
COMMAND\Xparafootsep@series	156
COMMAND\Xparinden	320
COMMAND\Xparindent	37, 372, 375, 377
COMMAND\Xpstart	32, 33, 315, 318, 366, 371
COMMAND\Xpstarteverytime	32, 315, 318, 371
COMMAND\Xragged	38
COMMAND\Xrigidbalance	158, 320, 376
COMMAND\Xstanza	33, 44
COMMAND\Xstanzaseparator	33
COMMAND\Xsublinesep	20, 33, 49, 320
COMMAND\Xsublinesepside	20, 33
COMMAND\Xsymlinenum	31, 38, 317, 318, 373
COMMAND\Xtwolines	23, 32, 49, 171, 219, 318, 371
COMMAND\Xtwolinesappref	219
COMMAND\Xtwolinesbutnotmore	32, 49, 318, 372
COMMAND\Xtwolinesbutnotmoreappref	220
COMMAND\Xtwolinesonlyinsamepage	32, 49, 318, 372
COMMAND\Xtxtbeforenotes	39, 318
COMMAND\Xunvxh	154, 319
COMMAND\Xwidth	40, 320, 377
COMMAND\&	317
COMMAND\absline@num	86, 133
COMMAND\accent	110
COMMAND\actionlines@list	87, 134
COMMAND\actions@list	87
COMMAND\add@inserts	87, 140
COMMAND\add@inserts@next	140, 141
COMMAND\add@penalties	132, 141
COMMAND\addcontentsline	294
COMMAND\addfootins	313, 316
COMMAND\addfootinsX	313, 316
COMMAND\advancelabel@refs	232

COMMAND\advanceline	21, 88, 95, 106, 374
COMMAND\advancepageno	225
COMMAND\affixlin@num	249
COMMAND\affixline@num	135, 138, 139, 364
COMMAND\affixpstart@num	139
COMMAND\afterXrule	317
COMMAND\afterXsymlinenum	317
COMMAND\afterrenumber	33
COMMAND\aftergroup	109, 113
COMMAND\afterlemmaseparator	317
COMMAND\afternote	317
COMMAND\afternoteX	38, 316, 317
COMMAND\afternumberinfootnote	317
COMMAND\afterruleX	39, 369, 372
COMMAND\applabel	48, 234, 235, 372
COMMAND\appref	46, 48, 49, 238, 240, 376, 377
COMMAND\apprefprefixmore	48, 317
COMMAND\apprefprefixsingle	48, 317
COMMAND\apprefwithpage	48, 49, 238, 240, 373, 376
COMMAND\arrangementX	30, 61, 178, 218, 317
COMMAND\arrangementX@normal	183
COMMAND\at@every@pend	126
COMMAND\autopar	16, 17, 124, 127, 128, 193, 365, 367, 368, 372
COMMAND\ballast	60
COMMAND\ballast@count	132, 142
COMMAND\baselineskip	30, 152, 156
COMMAND\beforeXnotes	317
COMMAND\beforeXsymlinenum	317
COMMAND\beforeelectedchapter	9, 58, 295, 296
COMMAND\beforelemmaseparator	317
COMMAND\beforenotesX	39, 319, 368, 369, 372
COMMAND\beforenumberinfootnote	317
COMMAND\begin	272
COMMAND\beginnumbering	15, 17, 18, 74, 75, 77, 85, 90, 104, 127, 199, 366, 369, 373, 374
COMMAND\bf	366
COMMAND\bfseries	36, 366
COMMAND\bhookXnote	318
COMMAND\bhookgroupX	39, 376
COMMAND\bhooknoteX	37, 319, 374, 375
COMMAND\body	266
COMMAND\bodyfootmarkA	28
COMMAND\boxXendlinenum	318
COMMAND\boxXendlinenumalign	318
COMMAND\boxXendstartlinenum	318
COMMAND\boxendlinenum	318
COMMAND\boxlinefootnote	168
COMMAND\boxlinenum	318
COMMAND\boxlinenumalign	318
COMMAND\boxstartlinenum	318
COMMAND\boxsymlinenum	318

COMMAND\break	31, 154
COMMAND\brokenpenalty	141
COMMAND\centering	38
COMMAND\ch@ck@l@ck	365
COMMAND\ch@cksub@l@ck	138, 365
COMMAND\chapter	57, 297, 369, 372, 374
COMMAND\chaptermark	295
COMMAND\check@pb@in@verse	308
COMMAND\colalignX	37, 372
COMMAND\collect@body	273
COMMAND\colorbox	61
COMMAND\columns	40
COMMAND\columnwidth	152, 370
COMMAND\command names	219
COMMAND\copyright	110
COMMAND\correct@Xfootins@box	371
COMMAND\correct@footinsX@box	371
COMMAND\count	160
COMMAND\critex	365
COMMAND\critext	116, 312, 313, 316
COMMAND\csize	65, 119
COMMAND\ctab	288, 293
COMMAND\ctabtext	293
COMMAND\dccl	282
COMMAND\def	63
COMMAND\detokenize	119
COMMAND\dimen	160
COMMAND\dimexpr	40
COMMAND\discretionary	153
COMMAND\displaywidowpenalty	141
COMMAND\do@actions	132–134, 365
COMMAND\do@actions@fixedcode	364
COMMAND\do@actions@next	133, 134
COMMAND\do@ballast	132, 133, 142
COMMAND\do@feetX	374
COMMAND\do@insidelinehook	367
COMMAND\do@line	87, 109, 125, 128, 131, 132, 140, 141, 265, 365, 367, 369
COMMAND\do@linehook	365
COMMAND\do@lockoff	88
COMMAND\do@lockon	88
COMMAND\dodoreintrafeet	363
COMMAND\doendnotes	24, 202, 319, 372
COMMAND\doendnotesbysection	24, 202, 214, 373
COMMAND\doinsidelinehook	21, 370
COMMAND\dolinehook	21, 370
COMMAND\doreintrafeeti	374
COMMAND\doreintrafeetii	374
COMMAND\doxtrafeet	225, 363
COMMAND\doxtrafeeti	374
COMMAND\doxtrafeetii	374

COMMAND\dummy@ref	109
COMMAND\edaftertab	56, 288, 289
COMMAND\edatleft	56, 286
COMMAND\edatright	56, 286
COMMAND\edbeforetab	56, 288
COMMAND\edfilldimen	287
COMMAND\edfont@info	115
COMMAND\edgls	53, 255
COMMAND\edgls...	377
COMMAND\edindex	51–53, 255, 256, 259, 261, 276, 367, 370, 371, 374, 375
COMMAND\edindexlab	53
COMMAND\edlabel	45–48, 50, 110, 230–233, 236, 243, 255, 276, 308, 363, 366–368, 371, 376
COMMAND\edlabelE	47, 235
COMMAND\edlabelS	47, 235
COMMAND\edlabelSE	47
COMMAND\edlineref	46, 230, 318, 371, 373, 376
COMMAND\edmakelabel	47, 243
COMMAND\edpageref	46, 230, 235, 236, 243
COMMAND\edrowfill	287
COMMAND\edtabcolsep	281
COMMAND\edtext	6, 22–25, 27, 28, 41, 46–48, 51, 54, 61, 87, 101, 102, 105, 108– 116, 118–120, 122, 234, 235, 238, 276, 278, 297, 308, 312, 313, 316, 364, 365, 367, 369–373
COMMAND\edtext@level	373
COMMAND\edvertdots	57, 286
COMMAND\edvertline	56, 57, 286
COMMAND\elechapter	58
COMMAND\eled@sectioning@out	301
COMMAND\eledchapter	57, 316, 370, 374
COMMAND\eledchapter*	57
COMMAND\eledmac@error	363
COMMAND\eledsection	6, 15, 57, 110, 130, 295, 316, 371
COMMAND\eledsection*	57
COMMAND\eledsubsection	57, 316
COMMAND\eledsubsection*	57
COMMAND\eledsubsubsection	57, 316
COMMAND\eledsubsubsection*	57
COMMAND\eledxxx	9, 58, 296, 301, 369
COMMAND\eledxxxx	295
COMMAND\else	254, 295
COMMAND\empty	73, 136, 231
COMMAND\end	272
COMMAND\end@lemmas	109
COMMAND\endashchar	41, 165
COMMAND\endgraf	125, 156, 193
COMMAND\endlock	20, 88, 107, 270
COMMAND\endminipage	250, 252, 363, 374
COMMAND\endnotes	372, 376, 377
COMMAND\endnumbering	15, 18, 74, 76, 77, 365, 373
COMMAND\endprint	199, 201, 214, 315
COMMAND\endstanzaextra	317

COMMAND\endsub	20, 88, 106
COMMAND\endverse	319
COMMAND\everypar	127
COMMAND\extensionchars	59, 74
COMMAND\externaldocument	50, 243
COMMAND\fix@l@cks	365
COMMAND\falseverse	316, 367, 369
COMMAND\fi	295
COMMAND\firstlinenum	19, 135, 365
COMMAND\firstsublinenum	19, 365
COMMAND\fix@page	93, 94, 364
COMMAND\flag@end	105, 115, 369
COMMAND\flag@start	105, 115, 369, 370
COMMAND\flagstanza	44
COMMAND\floatingpenalty	146, 371
COMMAND\flush@notes	142, 143
COMMAND\fnpos	195, 368
COMMAND\footfmt	145, 148
COMMAND\footfmt...	179
COMMAND\footfootmarkA	28
COMMAND\footfudgefactor	154
COMMAND\footfudgefiddle	61, 152, 363
COMMAND\footgroup	145
COMMAND\footins	149
COMMAND\footnormal	218, 317, 364
COMMAND\footnormalX	317
COMMAND\footnote	28, 60, 174–176, 314, 364
COMMAND\footnote@lang	165
COMMAND\footnoteA	15, 28
COMMAND\footnoteB	15
COMMAND\footnoteC	22
COMMAND\footnoteE	28
COMMAND\footnoteX	8, 212
COMMAND\footnoteX@reading	213
COMMAND\footnoteXmk	225
COMMAND\footnotelang@lua	144
COMMAND\footnotelang@poly	145
COMMAND\footnoteoption@	144, 375
COMMAND\footnoterule	159
COMMAND\footnotesize	37
COMMAND\footparagraph	152, 218, 317, 369
COMMAND\footparagraphX	188, 317, 369
COMMAND\footplitskips	365, 371
COMMAND\footstart	145, 149, 159
COMMAND\footstrut	156
COMMAND\footthreecol	317
COMMAND\footthreecolX	317, 372
COMMAND\foottwocol	317
COMMAND\foottwocolX	317, 372
COMMAND\fullstop	41, 320

COMMAND\get@edindex@hyperref	260
COMMAND\get@edindex@ledinnote@command	256
COMMAND\get@fnmark	176
COMMAND\get@index@command	368
COMMAND\get@linelistfile	365
COMMAND\get@thisfootnote	182
COMMAND\getline@num	132, 133
COMMAND\gl@p	79
COMMAND\global	92
COMMAND\globaldefs	92
COMMAND\gls	53, 263
COMMAND\hangindentX	37, 372, 375
COMMAND\hangingsymbol	317, 365
COMMAND\hbox	153, 154
COMMAND\hfill	368
COMMAND\hidenumbering	21, 100, 372
COMMAND\hline	54
COMMAND\hrulefill	287
COMMAND\hsize	31, 149, 152–154, 160, 163, 194, 364, 370
COMMAND\hsizeX	320, 376, 377
COMMAND\hsizethreecol	318
COMMAND\hsizethreecolX	38, 40
COMMAND\hsizetwocol	318
COMMAND\hsizetwocolX	38, 40
COMMAND\hyperlinkR	260
COMMAND\hyperlinkformat	259
COMMAND\hyperlinkformatR	260
COMMAND\if@RTL	67
COMMAND\if@edtext@	370, 373
COMMAND\if@eled@sectioning	296
COMMAND\if@nobreak	126
COMMAND\if@noneed@Footnote	105
COMMAND\ifXnote@	74
COMMAND\ifbypage@	79
COMMAND\ifbypage@R	79
COMMAND\ifbypstart@	79
COMMAND\ifbypstart@R	79
COMMAND\iffirst@linenum@out@	103, 104
COMMAND\ifndtl@innote	74
COMMAND\ifndtl@notenumber	74
COMMAND\ifinserthangingsymbol	265
COMMAND\ifinstanza	265
COMMAND\ifstwofollowinglines	171, 172
COMMAND\ifl@d@Xmorethantwolines	169, 372
COMMAND\ifl@d@Xtwolines	169
COMMAND\ifl@d@dash	169
COMMAND\ifl@d@elin	169
COMMAND\ifl@d@esl	169
COMMAND\ifl@d@pnum	169
COMMAND\ifl@d@ssub	169

COMMAND\ifl@dend@X	213
COMMAND\ifl@dmemoir	363
COMMAND\ifl@dpaging	370
COMMAND\ifl@dpairing	73, 365
COMMAND\ifl@dprintingpages	371
COMMAND\ifl@dskipnumber	135
COMMAND\ifl@dstartendok	287
COMMAND\ifl@imakeidx	67
COMMAND\ifledRcol	73, 366
COMMAND\ifledRcol@	73, 369
COMMAND\iflemmacommand@	371
COMMAND\ifnoend@	203
COMMAND\ifnoledgroup@	254
COMMAND\ifnoteschanged@	89
COMMAND\ifnumberedpar@	123
COMMAND\ifnumbering	74, 77
COMMAND\ifnumberingR	73, 366
COMMAND\ifnumberline	115, 135
COMMAND\ifpst@rted	365
COMMAND\ifpst@rtedL	75
COMMAND\ifseriesbefore	217
COMMAND\ifsublines@	86, 98
COMMAND\iftrue	373
COMMAND\ifvmode	232
COMMAND\ifxxx	295
COMMAND\ignorespaces	113
COMMAND\imki@wrindexentry	67
COMMAND\immediate	103, 104, 198
COMMAND\indent	17, 127, 319
COMMAND\index	262, 263
COMMAND\indtl@wrindexentry	67
COMMAND\initnumbering@quote	293, 374
COMMAND\initnumbering@reg	365
COMMAND\initnumbering@sectcmd	374
COMMAND\inplaceoflemmaseparator	318
COMMAND\inplaceofnumber	318
COMMAND\insert	140, 145, 148, 179
COMMAND\insert@count	101, 105, 113
COMMAND\insert@countR	113
COMMAND\insertthangingsymbol	368
COMMAND\insertlines@list	87, 101
COMMAND\insertparafootsepX	192
COMMAND\inserts@list	109, 124, 140, 141, 153
COMMAND\interAfootnotelinepenalty	364
COMMAND\interfootnotelinepenalty	364
COMMAND\interlinepenalty	146
COMMAND\interparanoteglue	316
COMMAND\justifying	38
COMMAND\l@advance@parledegroupp@beforenormalnotes	374
COMMAND\l@d@@@wrindexhyp	370

COMMAND\l@d@add	117
COMMAND\l@d@end	199, 213
COMMAND\l@d@nums	112, 115–117, 169
COMMAND\l@d@section	199
COMMAND\l@d@set	96, 107
COMMAND\l@dampcount	278
COMMAND\l@dbfnote	176, 364
COMMAND\l@dcheckstartend	287
COMMAND\l@dchset@num	97
COMMAND\l@dcolcount	278, 279
COMMAND\l@dcollect@@body	272
COMMAND\l@dcollect@body	272
COMMAND\l@dcsnote	369
COMMAND\l@dcsnotetext	131, 247, 248
COMMAND\l@dcsnotetext@l	131, 247, 248
COMMAND\l@dcsnotetext@r	131, 247, 248
COMMAND\l@ddodoreinxtrafeet	226, 363
COMMAND\l@ddoxtrafeet	226, 363
COMMAND\l@demptyd@ta	365
COMMAND\l@dend@close	198
COMMAND\l@dend@open	198
COMMAND\l@dend@stuff	199
COMMAND\l@denbody	272
COMMAND\l@dfeetbeginmini	364
COMMAND\l@dfeetendmini	364
COMMAND\l@dgetline@margin	365
COMMAND\l@dgetlock@disp	365
COMMAND\l@dgetref@num	237
COMMAND\l@dgetsidenote@margin	244, 365
COMMAND\l@dgobbeloptarg	370
COMMAND\l@dgonblearg	370
COMMAND\l@dgonbleoptarg	277
COMMAND\l@dlabel@parse	237
COMMAND\l@dld@ta	135, 137
COMMAND\l@dlp@rbox	249
COMMAND\l@dlsn@te	365
COMMAND\l@dlsnote	369
COMMAND\l@dmake@labels	233, 244
COMMAND\l@dmake@labelsR	244
COMMAND\l@dnumpstartsL	75, 365
COMMAND\l@dp@rsefootspec	169
COMMAND\l@dpush@begins	272
COMMAND\l@drd@ta	135, 137
COMMAND\l@dref@undefined	236
COMMAND\l@drsn@te	365
COMMAND\l@drsnote	369
COMMAND\l@dtabaddcols	287
COMMAND\l@dtabnoexpands	363
COMMAND\l@dumboxmpfoot	374
COMMAND\l@dunboxmpfoot	365

COMMAND\l@dzeropenalties	365, 370
COMMAND\l@pb	307
COMMAND\l@prev@nopb	306
COMMAND\l@prev@pb	306
COMMAND\l@reg	314
COMMAND\label	17, 47, 50, 53, 231, 237
COMMAND\label@refs	231
COMMAND\labelstarttrue	17, 366
COMMAND\labelref@list	230, 231, 233
COMMAND\language	153
COMMAND\last@page@num	364
COMMAND\lastbox	127
COMMAND\lastskip	106
COMMAND\leavevmode	17, 127
COMMAND\led@check@nopb	307
COMMAND\led@check@pb	307
COMMAND\led@nopb	306–308
COMMAND\led@nopbnum	306, 307
COMMAND\led@pb	306–308
COMMAND\led@pb@macro	307
COMMAND\led@pbnum	306, 307
COMMAND\led@reinit@index@fornote	263
COMMAND\led@set@index@fornote	262
COMMAND\ledRflag	260
COMMAND\ledchapter	316, 367
COMMAND\ledfootinsdim	317
COMMAND\ledinnernote	50, 246, 369
COMMAND\ledinnote	258, 315, 373
COMMAND\ledinnotemark	52, 315, 372
COMMAND\ledleftnote	50, 246
COMMAND\ledlinenum	85, 365
COMMAND\ledllfill	132
COMMAND\ledlsnotesep	51
COMMAND\ledlsnotewidth	50
COMMAND\lednopb	59, 306
COMMAND\lednopbinverse	308
COMMAND\lednopbinversetrue	44, 59
COMMAND\lednopbnum	306
COMMAND\ledouternote	50, 246, 369
COMMAND\ledpb	59, 306, 316
COMMAND\ledpbnum	306
COMMAND\ledpbsetting	59, 307, 375
COMMAND\ledrightnote	50, 246
COMMAND\ledrsnotesep	51
COMMAND\ledrsnotewidth	50
COMMAND\ledsection	316
COMMAND\ledsectnomark	295
COMMAND\ledsectnotoc	294
COMMAND\ledsetnormalparstuff	315, 316, 372
COMMAND\ledsetnormalparstuff@common	193

COMMAND\ledsetnormalparstuffX	315, 316, 372
COMMAND\ledsidenote	50, 246, 248
COMMAND\ledsubsection	316
COMMAND\ledsubsubsection	316
COMMAND\ledxxx	369
COMMAND\left	56
COMMAND\leftctab	288
COMMAND\leftheadline	84
COMMAND\leftlinenum	20, 84, 363, 365
COMMAND\leftltab	288
COMMAND\leftnoteupfalse	50
COMMAND\leftpstartnum	139
COMMAND\leftrtab	288
COMMAND\leftsidenote	248
COMMAND\leftskip	149, 153, 154
COMMAND\lemma	2, 23–25, 27, 28, 108, 112, 113, 116, 118, 312, 365, 366, 373, 374, 376
COMMAND\lemmaseparator	318
COMMAND\let	25, 41, 270, 363
COMMAND\letboxXendendlinenum	318
COMMAND\line	158, 161
COMMAND\line@list	87, 102, 115
COMMAND\line@list@stuff	75, 90, 104, 363, 365
COMMAND\line@list@version	92
COMMAND\line@margin	81, 137, 244
COMMAND\line@num	86, 88, 135, 363
COMMAND\line@set	117
COMMAND\lineation	19, 80
COMMAND\linebreak	31
COMMAND\linenum	23, 25, 46–48, 108, 116, 117, 236, 237, 243, 312
COMMAND\linenum@out	103, 231, 233
COMMAND\linenumberlist	19, 73, 136, 363
COMMAND\linenumberstyle	21, 84, 363
COMMAND\linenumincrement	19, 365
COMMAND\linenummargin	19, 81, 244
COMMAND\linenumr@p	84, 363, 365
COMMAND\linenumrep	84, 365
COMMAND\linenumsep	20, 51, 84, 246
COMMAND\linerangesep@	225
COMMAND\lineref	230, 236, 243, 318, 371
COMMAND\list@clear	78
COMMAND\list@clearing@reg	365
COMMAND\list@create	78
COMMAND\lock@disp	83
COMMAND\lock@off	99
COMMAND\lock@on	98
COMMAND\lockdisp	20, 83
COMMAND\loop	143, 266
COMMAND\ltab	288, 289, 293
COMMAND\ltabtext	293
COMMAND\m@mmf@prepare	175

COMMAND\makeatletter	131
COMMAND\makehboxofhboxes	155, 156
COMMAND\makeindex	51, 259
COMMAND\makelabel	243
COMMAND\managestanza@modulo	267
COMMAND\marginpar	50, 60, 244, 364
COMMAND\marginparwidth	50, 246
COMMAND\markboth	131
COMMAND\mathchardef	266
COMMAND\maxhXnotes	318
COMMAND\maxhnotesX	40, 61, 317, 368, 369, 371, 372
COMMAND\maxlinesinpar@list	90
COMMAND\measurembody	290
COMMAND\measuretbody	290
COMMAND\memorybreak	18
COMMAND\morenoexpands	61, 62, 109, 110
COMMAND\morethantwolines	318
COMMAND\mpfnpos	195, 368
COMMAND\mpnormalfootgroup	364
COMMAND\mpnormalvfootnote	364
COMMAND\multfootsep	29, 175
COMMAND\multiplefootnotemarker	175
COMMAND\musixtex	369
COMMAND\n@num	365, 372
COMMAND\n@num@ref	372
COMMAND\new@line	104, 364
COMMAND\newcommand	25, 63, 174, 233
COMMAND\newcommandx	25
COMMAND\newhookarg@specific	224
COMMAND\newhookcommand@series	219, 220, 372
COMMAND\newhookcommand@series@reload	220
COMMAND\newhookcommand@toggle@reload	220, 370
COMMAND\newhooktoggle@series	219, 220, 372
COMMAND\newhooktoggle@specific	224
COMMAND\newif	372
COMMAND\newline	31
COMMAND\newlinechar	214
COMMAND\newseries	29, 313, 316, 317
COMMAND\newseries@	206, 207, 217
COMMAND\newverse	44, 45, 316, 369
COMMAND\next	266
COMMAND\next@action	91
COMMAND\next@actionline	91
COMMAND\next@insert	141
COMMAND\nl@regR	93
COMMAND\no@expands	61, 116, 363
COMMAND\noXlemmaseparator	318
COMMAND\nobreak	168
COMMAND\nocritical	207
COMMAND\noeledsec	58, 316

COMMAND\noendnotes	316
COMMAND\noexpand	314
COMMAND\nofamiliar	222
COMMAND\noindent	17, 127, 375
COMMAND\noindentX	375
COMMAND\nomk@	225
COMMAND\nonbreakableafternumber	318
COMMAND\nonumberinfootnote	318
COMMAND\normal@footnotemarkX	178
COMMAND\normal@page@break	306
COMMAND\normal@pars	193
COMMAND\normalbfnoteX	365
COMMAND\normalbodyfootmarkX	179
COMMAND\normalfootfmt	41, 149, 156, 165, 199
COMMAND\normalfootfmtX	179, 180
COMMAND\normalfootfootmarkX	180
COMMAND\normalfootgroup	150
COMMAND\normalfootgroupX	181
COMMAND\normalfootnoterule	146
COMMAND\normalfootstart	149, 153
COMMAND\normalfootstartX	180
COMMAND\normalvfootnote	148
COMMAND\normalvfootnoteX	179
COMMAND\notbool	295
COMMAND\notfontsetup	317
COMMAND\notfontsizeX	37, 317
COMMAND\notenumfont	317
COMMAND\notenumfontX	36, 317
COMMAND\notesXwidthliketwocolumns	318
COMMAND\noteswidthliketwocolumnsX	40, 318, 370, 372
COMMAND\num@lines	123, 142
COMMAND\numberlinefalse	18
COMMAND\numberlinetrue	18
COMMAND\numberonlyfirstinline	217, 318
COMMAND\numberonlyfirstintwolines	318
COMMAND\numberpstartfalse	17
COMMAND\numberpstarttrue	17, 32, 315, 365, 374
COMMAND\numberstanza	33
COMMAND\numberstanzafalse	44
COMMAND\numberstanzatrue	44
COMMAND\numlabfont	20, 41, 85
COMMAND\one@line	123
COMMAND\onehalfspacing	375
COMMAND\onlyXpstart	318
COMMAND\page@action	88, 97
COMMAND\page@start	88, 365
COMMAND\pagecontents	88
COMMAND\pagelinesep	52
COMMAND\pageno	225
COMMAND\pageparbreak	316

COMMAND\pageref	47, 236
COMMAND\par	24, 31, 127, 193
COMMAND\par@line	123, 142
COMMAND\para@footgroup	153
COMMAND\para@footgroupX	191
COMMAND\para@footsetup	152, 363
COMMAND\para@footsetupX	189, 363, 370
COMMAND\para@vfootnote	156
COMMAND\para@vfootnoteX	190
COMMAND\parafootfmt	155, 156
COMMAND\parafootfmtX	191
COMMAND\parafootftm	158
COMMAND\parafootftmX	192
COMMAND\parafootftmsep	317
COMMAND\parafootsep	318, 368, 373
COMMAND\parafootsepX	38, 89, 317, 318
COMMAND\parafootstart	153
COMMAND\parafootstartX	189
COMMAND\paravfootnote	153
COMMAND\parfillskip	155
COMMAND\parindent	375
COMMAND\parindentX	37, 320, 375, 377
COMMAND\parshape	60
COMMAND\parskip	127
COMMAND\pausenumbering	18, 77, 90, 92, 128, 368, 370
COMMAND\penalty	155
COMMAND\pend	2, 6, 16–19, 21, 58, 106, 109, 111, 117, 123–128, 139, 140, 314, 368, 369
COMMAND\preXnotes	39, 197, 372
COMMAND\preXnotes@	149, 197, 366
COMMAND\prenotesX	39, 197, 372
COMMAND\prepare@preXnotes	196
COMMAND\prev@nopb	307
COMMAND\prev@pb	307
COMMAND\prevlineX	89
COMMAND\prevpageX@num	89
COMMAND\print@Xfootnoterule	373
COMMAND\print@Xnotes	226, 227
COMMAND\print@Xnotes@forpages	371
COMMAND\print@eledsection	130
COMMAND\print@footnoteXrule	373
COMMAND\print@leftmargin@eledsection	296
COMMAND\print@line	129
COMMAND\print@notesX@forpages	371
COMMAND\print@rightmargin@eledsection	296
COMMAND\printendlines	203, 240, 320, 363, 365
COMMAND\printlinefootnote	166, 167, 371
COMMAND\printlinefootnotearea	167, 168, 371
COMMAND\printlinefootnotenumbers	166
COMMAND\printlines	149, 165, 168, 170, 203, 240, 320, 363, 365, 372, 376
COMMAND\printnnum	315, 320

COMMAND\printpstart	165
COMMAND\protect	110, 314
COMMAND\providecommand	174, 363
COMMAND\pstart	2, 6, 16–19, 21, 57, 58, 96, 106, 107, 111, 117, 123–127, 130, 140, 314, 365, 366, 368–370, 372–374, 377
COMMAND\pstartinfootnote	318
COMMAND\pstartinfootnoteeverytime	318
COMMAND\pstartnum	139
COMMAND\pstartref	46, 230, 236, 368
COMMAND\pstarts	366
COMMAND\raggedX	38
COMMAND\raggedleft	38
COMMAND\raggedright	38
COMMAND\raw@text	123, 124
COMMAND\rbracket	35, 36, 41
COMMAND\read@linelist	90–92
COMMAND\ref	47, 50, 53
COMMAND\reformatted@	240
COMMAND\reformattedwithpage	240
COMMAND\relax	17, 97, 133, 140, 270, 277, 314
COMMAND\renewcommand	61, 317, 320
COMMAND\resetprevline@	89
COMMAND\resetprevpage@	89
COMMAND\resumenumbering	18, 74, 77, 90, 92, 128, 365, 369, 370
COMMAND\right	56
COMMAND\rightctab	288
COMMAND\rightlinenum	20, 84, 363, 365
COMMAND\rightltab	289
COMMAND\rightnoteupfalse	50
COMMAND\rightrtab	289
COMMAND\rightrightnote	248
COMMAND\rightrightskip	149, 153–155
COMMAND\rightrightstartnum	139
COMMAND\rigidbalance	158, 160, 161, 320, 376
COMMAND\rigidbalanceX	158, 320, 376
COMMAND\robustify	31
COMMAND\roman	282, 376
COMMAND\rtab	288, 289, 293
COMMAND\rtabtext	290, 293
COMMAND\sameword	26–28, 117–120, 122, 371, 373, 375
COMMAND\sameword@inedtext	119
COMMAND\saweword	118
COMMAND\scriptsize	85
COMMAND\section	57, 365
COMMAND\section@num	74
COMMAND\sectionmark	295
COMMAND\select@lemmafont	41, 143
COMMAND\series	206, 207
COMMAND\series@	207
COMMAND\seriesatbegin	29, 216, 372

COMMAND\seriesatend	29, 216, 373
COMMAND\set@line	115
COMMAND\set@line@action	88, 97
COMMAND\setSErefonlypageprefixmore	48, 240, 376
COMMAND\setSErefonlypageprefixsingle	48, 240, 376
COMMAND\setSErefprefixmore	48
COMMAND\setSErefprefixsingle	48
COMMAND\setapprefprefixmore	48, 317
COMMAND\setapprefprefixsingle	48, 317, 376
COMMAND\setcommand@series	218
COMMAND\sethangingsymbol	43, 265, 317, 375
COMMAND\sethangingsymbol	42
COMMAND\setistwofollowinglines	171
COMMAND\setl@dlprbox	249
COMMAND\setline	21, 88, 93, 96, 106, 110, 125, 374
COMMAND\setlinenum	21, 92, 96, 107, 363
COMMAND\setprintendlines	203, 204, 365
COMMAND\setprintlines	170, 171, 203, 365
COMMAND\setsidenoteseq	51
COMMAND\setsidenotsep	317
COMMAND\setstanzaindent	267
COMMAND\setstanzaindents	43, 266, 314
COMMAND\setstanzapenalties	266
COMMAND\setstanzavalues	266
COMMAND\settoggle@series	217, 366, 370
COMMAND\showlemma	110, 364
COMMAND\showwordrank	28, 119
COMMAND\sidenote@margin	364
COMMAND\sidenotemargin	50, 364, 369
COMMAND\sidenoteseq	317
COMMAND\sidepstartnumtrue	17
COMMAND\skip	149
COMMAND\skipnumbering	21, 100, 107, 365, 373
COMMAND\skipnumbering@reg	373
COMMAND\small	37
COMMAND\special	12
COMMAND\splitmaxdepth	146, 160
COMMAND\splitoff	158
COMMAND\splittopskip	146, 160, 161
COMMAND\stanza	20, 21, 44, 45, 269, 317, 319, 375
COMMAND\stanza@hang	269
COMMAND\stanza@line	269
COMMAND\stanzaindent	43, 267, 371
COMMAND\stanzaindent*	43
COMMAND\stanzaindentbase	266
COMMAND\stanzanumwrapper	44
COMMAND\startlock	20, 88, 107, 270
COMMAND\startstanzahook	317
COMMAND\startsub	20, 88, 106
COMMAND\strip@pt	153

COMMAND\strutbox	160
COMMAND\sub@action	88, 98
COMMAND\sub@lock	86
COMMAND\sub@off	95, 233
COMMAND\sub@on	95, 233
COMMAND\subline@num	86, 88
COMMAND\sublinenum@rep	363
COMMAND\sublinenumberstyle	21, 84, 363
COMMAND\sublinenumincrement	19
COMMAND\sublinenumr@p	84, 363, 365
COMMAND\sublinenumrep	84, 365
COMMAND\sublineref	46, 230, 236
COMMAND\subsectionmark	295
COMMAND\sw@inthisedtext	112
COMMAND\sw@list@inedtext	116, 122
COMMAND\symlinenum	318
COMMAND\symplinum	317
COMMAND\sza@penalty	269
COMMAND>tag	371
COMMAND\text	312
COMMAND\textcolor	62
COMMAND\textheight	61
COMMAND\the	363
COMMAND\thefootnoteA	28
COMMAND\thefootnoteX	367
COMMAND\thelabidx	261
COMMAND\thepage	93
COMMAND\thepstart	17
COMMAND\thepstartL	366
COMMAND\thepstartR	366
COMMAND\thestanza	44
COMMAND\this@line@list@version	103
COMMAND\thisfootnote	182
COMMAND\threecolfootfmt	160
COMMAND\threecolfootfmtX	187
COMMAND\threecolfootgroup	160
COMMAND\threecolfootgroupX	188
COMMAND\threecolfootsetup	159
COMMAND\threecolfootsetupX	187
COMMAND\threecolvfootnote	160
COMMAND\threecolvfootnoteX	187
COMMAND\twocolfootfmtX	185
COMMAND\twocolfootgroupX	185
COMMAND\twocolfootsetupX	184
COMMAND\twocolvfootnoteX	185
COMMAND\twolines	217, 318
COMMAND\twolines@A	217
COMMAND\twolines@B	217
COMMAND\twolines@C	217
COMMAND\twolinesbutnotmore	318

COMMAND\twolinesonlyinsamepage	318
COMMAND\txtbeforeXnotes	318
COMMAND\unhbox	153
COMMAND\unpenalty	155, 156
COMMAND\unskip	155
COMMAND\unvxh	155, 319
COMMAND\unvxhX	319
COMMAND\upbracefill	287
COMMAND\usingcritext	313, 316
COMMAND\usingedtext	313, 316
COMMAND\vAfootnote	146
COMMAND\variant	25
COMMAND\ vbox	125, 127, 154, 158, 195
COMMAND\vfootnote	145, 149, 153, 160
COMMAND\vl@dbfnote	176, 364
COMMAND\ vnumfootnoteX	365
COMMAND\ vsize	39, 40, 61
COMMAND\ vsplit	141
COMMAND\ waklam	287
COMMAND\ waklamec	287
COMMAND\ wapunktel	287
COMMAND\ wastricht	287
COMMAND\ widthX	40, 320, 377
COMMAND\ wrap@edcrossref	235, 370
COMMAND\ wrapped@bodyfootmarkX	193
COMMAND\ wrapped@footfootmarkX	192
COMMAND\ x...	46
COMMAND\ xdef	78, 270
COMMAND\ xflagref	47, 236, 320, 376
COMMAND\ xleft@appenditem	79, 109
COMMAND\ xlineref	46, 47, 320, 376
COMMAND\ xpagerref	46
COMMAND\ xpstartref	46, 368
COMMAND\ xr	50
COMMAND\ xright@appenditem	78, 79
COMMAND\ xsublineref	46
COMMAND\ xxref	47, 237, 238, 243, 368, 371, 372
COMMAND\ zz@@@	363
ENVIRONMENTastanza	376
ENVIRONMENTedarrayc	293
ENVIRONMENTedarrayl	293
ENVIRONMENTedarrayr	293
ENVIRONMENTedtabularc	293
ENVIRONMENTedtabularl	293
ENVIRONMENTedtabularr	293
ENVIRONMENTledgroup	66, 253, 320, 376
ENVIRONMENTledgroup sized	254
PACKAGE(r)(e)ledmac	29
PACKAGEEledmac	11, 63, 88, 258, 315, 316, 371–373
PACKAGEEledpar	372, 373

PACKAGEetoolbox	65
PACKAGEParallel	321
PACKAGEReledmac	319, 320
PACKAGEamsgen	271
PACKAGEamsmath	271
PACKAGEbabel	62, 282, 376
PACKAGEbiblatex	60
PACKAGEbidi	67, 375
PACKAGEcaption	73
PACKAGEcolor	61
PACKAGEedmac	1, 5, 10–13, 63, 169, 174, 231, 266, 312, 321, 363
PACKAGEedstanza	1, 13, 265
PACKAGEeledmac	1, 10, 13–15, 52, 117, 174, 255, 258, 275, 297, 309, 313, 315–317, 367, 369, 371
PACKAGEeledpar	73, 146, 295, 321, 366, 369–372
PACKAGEetex	375
PACKAGEetoolbox	78, 117, 207, 217, 225, 247, 295, 306
PACKAGEfloatrow	60, 320
PACKAGEfootmisc	29, 62, 174, 321
PACKAGEglossaries	53, 263, 376
PACKAGEhandout	370
PACKAGEhyperlink	213
PACKAGEhyperref	47, 112, 192, 193, 231, 260, 300, 308, 368–370, 377
PACKAGEifluatex	65
PACKAGEifxetex	65
PACKAGEimakeidx	51, 60, 66, 67, 255, 258, 316, 367–369, 371
PACKAGEindextols	262
PACKAGEindextool	316
PACKAGEindextools	51, 60, 66, 67, 74, 255, 258, 262, 316, 371, 376
PACKAGEinputenc	119
PACKAGEledarab	62
PACKAGEledmac	1, 10, 13, 62, 78, 258, 312, 313, 316, 319
PACKAGEledpar	62
PACKAGEMemoir	66, 258, 316, 321, 370
PACKAGEMorewrites	60
PACKAGEMusixtex	369
PACKAGEperpage	376
PACKAGEpolyglossia	35, 62, 113, 145, 165, 376
PACKAGERagged2e	38, 65
PACKAGEREledmac	1, 2, 10, 11, 13–15, 18, 19, 21–23, 25–27, 29, 31, 34, 37–39, 41, 43, 45–48, 50–54, 58, 60–64, 79, 82, 87, 88, 91, 92, 95, 103, 110, 140, 147, 149, 154, 174, 199, 207, 211, 217, 225, 235, 238, 243, 258, 275, 295, 296, 308, 316, 317, 319, 320, 374, 377
PACKAGEREledpar	1, 4, 5, 8, 14, 17, 40, 47, 49, 58–60, 62, 64, 73, 79, 90, 95, 113, 147, 150, 194, 195, 207, 212, 225–227, 255, 265, 375, 376
PACKAGESuffix	65
PACKAGETabmac	1, 13, 321
PACKAGEuninormalize	26
PACKAGEXargs	25, 65
PACKAGEXkeyval	64, 225
PACKAGEXref	4, 50, 243, 244, 377
PACKAGEXref	243

PACKAGEstring 65, 260

A

\absline@num 1
 Abu Kamil Shuja' b. Aslam 12
 \actionlines@list 1
 \actions@list 1
 \add@inserts 1
 \add@inserts@next 1
 \add@penalties 1
 \addtol@denbody 1
 Adelard II 12
 \advancelabel@refs 1
 \advanceline 1, 20
 \advancepageno 1
 \Aendnote 23
 \affixline@num 1
 \affixpstart@num 1
 \affixside@note 1
 \Afootnote 23
 \afternoteX 38
 \afterruleX 39
 \ampersand 1, 44
 \applabel 1, 48
 \appref 1, 48
 \apprefwithpage 1, 48
 \arrangementX 1, 30
 \arrangementX@normal 1
 \arrangementX@threecol 1
 \arrangementX@twocol 1
 \at@every@pend 1
 \AtEveryPend 1, 17
 \AtEveryPstart 1, 17
 \autopar 1, 16

B

\ballast 60
 \ballast@count 1
 Beeton, Barbara Ann Neuhaus Friend 17
 \beforeeledchapter 1
 \beforenotesX 39
 \beginnumbering 1, 15
 \Bendnote 23
 \Bfootnote 23
 \bhookgroupX 39
 \bhooknoteX 37
 \bodyfootmarkA 28
 \boxfootnotenumbers 1
 Bredon, Simon 12
 Breger, Herbert 12, 13, 274

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	23
\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.	12
\colalignX	37
Copernicus, Nicolaus	12
\critext	312
\ctab	1
\ctabtext	1

D

Dekker, Dirk-Jan	61
\Dendnote	23
\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
\do@linehook	1
\do@lockoff	1
\do@lockoffL	1
\do@lockon	1
\do@lockonL	1
\doedindexlabel	1

<code>\doendnotes</code>	1, 24
<code>\doendnotesbysection</code>	1, 24
<code>\doinsidelinehook</code>	1, 21
<code>\dolinehook</code>	1, 21
<code>\dosplits</code>	1
Downes, Michael	61, 153, 155
<code>\doextrafeet</code>	1
<code>\dummy@edtext</code>	1
<code>\dummy@edtext@showlemma</code>	1
<code>\dummy@ref</code>	1

E

<code>\edaftertab</code>	1, 56, 288
<code>edarrayc</code> (environment)	54
<code>edarrayl</code> (environment)	54
<code>edarrayr</code> (environment)	54
<code>\edatleft</code>	1, 56
<code>\edatright</code>	1, 56
<code>\edbeforetab</code>	1, 56, 288
<code>\edfilldimen</code>	1
<code>\edfont@info</code>	1
<code>\EDINDEX</code>	1
<code>\edindex</code>	1, 51
<code>\edindexlab</code>	1, 53
<code>\EDLABEL</code>	1
<code>\edlabel</code>	1, 45
<code>\edlabelE</code>	1, 47
<code>\edlabelS</code>	1, 47
<code>\edlabelSE</code>	1, 47
<code>\edlineref</code>	1, 46
<code>\edmakelabel</code>	1, 47
<code>\edpageref</code>	1, 46
<code>\edrowfill</code>	1, 55
<code>\EDTAB</code>	1
<code>\edtabcolsep</code>	1, 54
<code>\EDTABINDENT</code>	1
<code>\edtabindent</code>	1
<code>\EDTABtext</code>	1
<code>edtabularc</code> (environment)	54
<code>edtabularl</code> (environment)	54
<code>edtabularr</code> (environment)	54
<code>\EDTEXT</code>	1
<code>\edtext</code>	1, 22
<code>\edvertdots</code>	1, 56
<code>\edvertline</code>	1, 56
<code>\Eendnote</code>	23
<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, 20
<code>\endminipage</code>	1
<code>\endnumbering</code>	1, 15
<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>	54
<code>edarrayl</code>	54
<code>edarrayr</code>	54
<code>edtabularc</code>	54
<code>edtabularl</code>	54
<code>edtabularr</code>	54
<code>ledgroup</code>	45
<code>ledgroupsize</code>	45
<code>minipage</code>	45
<code>Euclid</code>	12
<code>\extensionchars</code>	1, 59

F

<code>\f@x@l@cks</code>	1
Fairbairns, Robin	29
<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, 44
<code>\flush@notes</code>	1

<code>\fnpos</code>	1, 29
Folkerts, Menso	12
<code>\footfootmarkA</code>	28
<code>\footfudgefiddle</code>	1, 61
<code>\footnote</code>	1
<code>\footnoteA</code>	28
<code>\footnoteB</code>	28
<code>\footnoteC</code>	28
<code>\footnoteD</code>	28
<code>\footnoteE</code>	28
<code>\footnotelang@lua</code>	1
<code>\footnotelang@poly</code>	1
<code>\footnoteoptions@</code>	1
<code>\footsplitskips</code>	1
<code>\fullstop</code>	1

G

Gädeke, Nora	12
<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>	37
<code>\hidenumbering</code>	1, 21
<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>	38
<code>\hsizetwocolX</code>	38
<code>\Hy@raisedlink@left</code>	1
<code>\hyperlinkformat</code>	1
<code>\hyperlinkformatR</code>	1
<code>\hyperlinkR</code>	1

I

<code>\if@addsw</code>	1
<code>\if@edindex@fornote@true</code>	1

<code>\if@eled@sectioning</code>	1
<code>\if@led@nofoot</code>	1
<code>\if@ledgroup</code>	1
<code>\if@lemmacommand@</code>	1
<code>\if@noeled@sec</code>	1
<code>\if@noneed@Footnote</code>	1
<code>\if@RTL</code>	1
<code>\ifautopar@pause</code>	1
<code>\ifbypage@</code>	1
<code>\ifbypage@R</code>	1
<code>\ifbypstart@</code>	1
<code>\ifbypstart@R</code>	1
<code>\ifeledmaccompat@</code>	1
<code>\iffirst@linenum@out@</code>	1
<code>\ifindtl@innote</code>	1
<code>\ifindtl@notenumber</code>	1
<code>\ifinserthangingsymbol</code>	1
<code>\ifinstanza</code>	1
<code>\ifl@d@dash</code>	1
<code>\ifl@d@elin</code>	1
<code>\ifl@d@esl</code>	1
<code>\ifl@d@pnum</code>	1
<code>\ifl@d@ssub</code>	1
<code>\ifl@d@Xmorethantwolines</code>	1
<code>\ifl@d@Xtwolines</code>	1
<code>\ifl@dend@X</code>	1
<code>\ifl@dhidenumber</code>	1
<code>\ifl@dmemoir</code>	1
<code>\ifl@dpaging</code>	1
<code>\ifl@dpairing</code>	1
<code>\ifl@dprintingcolumns</code>	1
<code>\ifl@dprintingpages</code>	1
<code>\ifl@dskipnumber</code>	1
<code>\ifl@dskipversenumber</code>	1
<code>\ifl@dstartendok</code>	1
<code>\ifl@imakeidx</code>	1
<code>\ifl@indextools</code>	1
<code>\ifledfinal</code>	1, 59
<code>\ifledgroupnotesL@</code>	1
<code>\ifledgroupnotesR@</code>	1
<code>\iflednopbinverse</code>	1
<code>\ifledRcol</code>	1
<code>\ifledRcol@</code>	1
<code>\ifnocritical@</code>	1
<code>\ifnoend@</code>	1
<code>\ifnofamiliar@</code>	1
<code>\ifnoledgroup@</code>	1
<code>\ifnoquotation@</code>	1
<code>\ifnoteschanged@</code>	1
<code>\ifnumberedpar@</code>	1

<code>\ifnumbering</code>	1
<code>\ifnumberingR</code>	1
<code>\ifnumberline</code>	1
<code>\ifnumberstanza</code>	1
<code>\ifparapparatus@</code>	1
<code>\ifparledgroup</code>	1
<code>\ifpst@rtedL</code>	1
<code>\ifseriesbefore</code>	1
<code>\ifsidepstartnum</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>\inserthangingymbol</code>	1
<code>\insertlines@list</code>	1
<code>\inserts@list</code>	1

J

Jayaditya	12
-----------------	----

K

Kabelschacht, Alois	143
---------------------------	-----

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@ddodoreinxtrafeet</code>	1
<code>\l@dedbeginmini</code>	1
<code>\l@dedendmini</code>	1

<code>\l@emptyd@ta</code>	1
<code>\l@dend@close</code>	1
<code>\l@dend@open</code>	1
<code>\l@dend@stuff</code>	1
<code>\l@dend@Xfalse</code>	1
<code>\l@dend@Xtrue</code>	1
<code>\l@denbody</code>	1
<code>\l@dfambeginmini</code>	1
<code>\l@dfamendmini</code>	1
<code>\l@dfeetbeginmini</code>	1
<code>\l@dfeetendmini</code>	1
<code>\l@getline@margin</code>	1
<code>\l@getlock@disp</code>	1
<code>\l@getref@num</code>	1
<code>\l@getsidenote@margin</code>	1
<code>\l@gobblearg</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>\l@drp@rbox</code>	1
<code>\l@drrsn@te</code>	1
<code>\l@drrsnote</code>	1
<code>\l@dsetmaxcolwidth</code>	1
<code>\l@dskipnumberfalse</code>	1
<code>\l@dskipnumbertrue</code>	1
<code>\l@dtabaddcols</code>	1
<code>\l@dtabnoexpands</code>	1
<code>\l@dunboxmpfoot</code>	1
<code>\l@dunhbox@line</code>	1
<code>\l@dzeropenalties</code>	1
<code>\label</code>	47

\labelstartfalse	1
\labelstarttrue	1, 17
\labelref@list	1
\labelrefsparseline	1
\labelrefsparsesubline	1
\last@page@num	1
Lavagnino, John	11
\led@check@nopb	1
\led@check@pb	1
\led@err@AutoparNotNumbered	1
\led@err@edtextoutsidepstart	1
\led@err@EdtextWithoutFootnote	1
\led@err@FootnoteWithoutEdtext	1
\led@err@HighEndColumn	1
\led@err@LineationInNumbered	1
\led@err@LowStartColumn	1
\led@err@ManyLeftnotes	1
\led@err@ManyRightnotes	1
\led@err@ManySidenotes	1
\led@err@NumberingNotStarted	1
\led@err@NumberingShouldHaveStarted	1
\led@err@NumberingStarted	1
\led@err@NumberingWithoutPstart	1
\led@err@PendNoPstart	1
\led@err@PendNotNumbered	1
\led@err@PstartInPstart	1
\led@err@PstartNotNumbered	1
\led@err@ReverseColumns	1
\led@err@TooManyColumns	1
\led@err@UnequalColumns	1
\led@error@fail@patch@@doclearpage	1
\led@error@fail@patch@@iiiminipage	1
\led@error@fail@patch@@makecol	1
\led@error@fail@patch@@reinserts	1
\led@error@fail@patch@endminipage	1
\led@error@ImakeidxAfterEledmac	1
\led@error@IndextoolsAfterEledmac	1
\led@mess@NotesChanged	1
\led@mess@SectionContinued	1
\led@nopb	1
\led@nopbnum	1
\led@pb	1
\led@pb@setting	1
\led@pbnum	1
\led@reinit@index@fornote	1
\led@set@index@fornote	1
\led@toksa	1
\led@toksb	1
\led@warn@AppLabelOutEdtext	1
\led@warn@BadAction	1

<code>\led@warn@BadAdvancelineLine</code>	1
<code>\led@warn@BadAdvancelineSubline</code>	1
<code>\led@warn@BadLineation</code>	1
<code>\led@warn@BadLinenummargin</code>	1
<code>\led@warn@BadLockdisp</code>	1
<code>\led@warn@BadSetline</code>	1
<code>\led@warn@BadSetlinenum</code>	1
<code>\led@warn@BadSidenotemargin</code>	1
<code>\led@warn@BadSublockdisp</code>	1
<code>\led@warn@DuplicateLabel</code>	1
<code>\led@warn@LineFileObsolete</code>	1
<code>\led@warn@NoIndexFile</code>	1
<code>\led@warn@NoLineFile</code>	1
<code>\led@warn@NoMarginpars</code>	1
<code>\led@warn@RefUndefined</code>	1
<code>\led@warn@SeriesStillExist</code>	1
<code>\led@warning@hsizeX@deprecated</code>	1
<code>\led@warning@Xhsize@deprecated</code>	1
<code>ledgroup (environment)</code>	45
<code>ledgroupsize (environment)</code>	45
<code>\ledinnernote</code>	1, 50
<code>\ledinnote</code>	1
<code>\ledinnotehyperpage</code>	1
<code>\ledinnotemark</code>	1
<code>\ledleftnote</code>	1, 50
<code>\ledlinenum</code>	1
<code>\ledllfill</code>	1
<code>\ledlsnotefontsetup</code>	1, 51
<code>\ledlsnotesep</code>	1, 51
<code>\ledlsnotewidth</code>	1, 50
<code>\lednopb</code>	1, 58
<code>\lednopbinversetrue</code>	59
<code>\lednopbnum</code>	1
<code>\ledouternote</code>	50
<code>\ledouterote</code>	1
<code>\ledpb</code>	1, 58
<code>\ledpbnum</code>	1
<code>\ledpbsetting</code>	1, 59
<code>\ledrightnote</code>	1, 50
<code>\ledrlfill</code>	1
<code>\ledrsnotefontsetup</code>	1, 51
<code>\ledrsnotesep</code>	1, 51
<code>\ledrsnotewidth</code>	1, 50
<code>\ledsectnomark</code>	1
<code>\ledsectnotoc</code>	1
<code>\ledsetnormalparstuff@common</code>	1
<code>\ledsetnormalparstuffX</code>	1
<code>\ledsidenote</code>	1, 50
<code>\leftctab</code>	1
<code>\leftlinenum</code>	1, 20

<code>\leftltab</code>	1
<code>\leftnoteupfalse</code>	50
<code>\leftpstartnum</code>	1
<code>\leftrtab</code>	1
Leibniz	12
<code>\lemma</code>	1, 24
<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, 19
<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, 19
<code>\linenumr@p</code>	1
<code>\linenumrep</code>	1
<code>\linenumsep</code>	1, 20
<code>\linerangesep@</code>	1
<code>\list@clear</code>	1
<code>\list@clearing@reg</code>	1
<code>\list@create</code>	1
<code>\lock@disp</code>	1
<code>\lock@off</code>	1
<code>\lock@on</code>	1
<code>\lockdisp</code>	1, 20
Lorch, Richard	12
<code>\ltab</code>	1
<code>\ltabtext</code>	1
Luecking, Dan	65

M

<code>\m@mmf@check</code>	1
<code>\m@mmf@prepare</code>	1
<code>\M@sect</code>	1
<code>\makehboxofhboxes</code>	1
<code>\managestanza@modulo</code>	1
<code>\maxhnotesX</code>	40
Mayer, Gyula	12
<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	12, 86
minipage (environment)	45
Mittelbach, Frank	12
\morenoexpands	<u>1</u> , 61
\mpfnpos	<u>1</u> , 29
\mpnormalfootgroup	<u>1</u>
\mpnormalfootgroupX	<u>1</u>
\mpnormalvfootnote	<u>1</u>
\mpnormalvfootnoteX	<u>1</u>
\mppara@footgroupX	<u>1</u>
\mppara@vfootnoteX	<u>1</u>
\mpparafootgroup	<u>1</u>
\mpparavfootnote	<u>1</u>
\mpthreecolfootgroup	<u>1</u>
\mpthreecolfootgroupX	<u>1</u>
\mpthreecolfootsetup	<u>1</u>
\mpthreecolfootsetupX	<u>1</u>
\mptwocolfootgroup	<u>1</u>
\mptwocolfootgroupX	<u>1</u>
\mptwocolfootsetup	<u>1</u>
\mptwocolfootsetupX	<u>1</u>
\multfootsep	<u>1</u> , 29
\multiplefootnotemarker	<u>1</u>

N

\n@num	<u>1</u>
\n@num@stanza	<u>1</u>
\new@line	<u>1</u>
\newhookarg@specific	<u>1</u>
\newhookcommand@series	<u>1</u>
\newhookcommand@series@reload	<u>1</u>
\newhooktoggle@series	<u>1</u>
\newhooktoggle@series@reload	<u>1</u>
\newhooktoggle@specific	<u>1</u>
\newseries@	<u>1</u>
\newverse	<u>1</u>
\NEXT	<u>1</u>
\no@expands	<u>1</u>
\noeledsec	58
\nomk@	<u>1</u>
\normal@footnotemarkX	<u>1</u>
\normal@page@break	<u>1</u>
\normal@pars	<u>1</u>
\normalbfnoteX	<u>1</u>
\normalbodyfootmarkX	<u>1</u>
\normalfootfmt	<u>1</u>
\normalfootfmtX	<u>1</u>
\normalfootfootmarkX	<u>1</u>
\normalfootgroup	<u>1</u>
\normalfootgroupX	<u>1</u>

<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>	37
<code>\notenumfontX</code>	36
<code>\noteschanged@false</code>	1
<code>\noteschanged@true</code>	1
<code>\noteswidthliketwocolumnsX</code>	40
<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>	18
<code>\numberlinetrue</code>	18
<code>\numberpstartfalse</code>	1, 17
<code>\numberpstarttrue</code>	1, 17
<code>\numberstanzafalse</code>	44
<code>\numberstanzatrue</code>	44
<code>\numlabfont</code>	1, 41

O

<code>\old@hsize</code>	1
<code>\one@line</code>	1
<code>optioninnnote</code>	376
<code>optioninnote</code>	376
<code>optionlinangesep</code>	225
<code>optionnocritical</code>	376
<code>optionnoend</code>	376
<code>optionnotenumber</code>	376

P

<code>\page@action</code>	1
<code>\page@num</code>	1
<code>\pagelinesep</code>	1, 52
<code>\pageno</code>	1
<code>\pageref</code>	47
<code>\par@line</code>	1
<code>\para@footgroupX</code>	1
<code>\para@footsetup</code>	1
<code>\para@footsetupX</code>	1
<code>\para@vfootnoteX</code>	1
<code>\parafootfmt</code>	1
<code>\parafootfmtX</code>	1
<code>\parafootgroup</code>	1

<code>\parafootsepX</code>	38
<code>\parafootstart</code>	1
<code>\parafootstartX</code>	1
<code>\paravfootnote</code>	1
<code>\parindentX</code>	37
<code>\pausenumbering</code>	1, 18
<code>\pend</code>	1, 16
Plato of Tivoli	12
<code>\postbodyfootmark</code>	1
<code>\prebodyfootmark</code>	1
<code>\prenotesX</code>	39
<code>\prepare@edindex@fornote</code>	1
<code>\prepare@prenotesX</code>	1
<code>\prepare@preXnotes</code>	1
<code>\prev@nopb</code>	1
<code>\prev@pb</code>	1
<code>\prevpage@num</code>	1
<code>\preXnotes</code>	1, 39
<code>\preXnotes@</code>	1
<code>\print@eledsection</code>	1
<code>\print@footnoteXrule</code>	1
<code>\print@leftmargin@eledsection</code>	1
<code>\print@line</code>	1
<code>\print@notesX</code>	1
<code>\print@rightmargin@eledsection</code>	1
<code>\print@Xfootnoterule</code>	1
<code>\print@Xnotes</code>	1
<code>\printendlines</code>	1
<code>\printlineendnote</code>	1
<code>\printlineendnotearea</code>	1
<code>\printlinefootnote</code>	1
<code>\printlinefootnotearea</code>	1
<code>\printlinefootnotenumbers</code>	1
<code>\printlines</code>	1
<code>\printnpnum</code>	1
<code>\printpstart</code>	1
<code>\printsymlineendnotearea</code>	1
<code>\printsymlinefootnotearea</code>	1
<code>\printXafternumber</code>	1
<code>\printXbeforenumber</code>	1
<code>\pstart</code>	1, 16
<code>\pstarteref</code>	1
<code>\pstartnum</code>	1
<code>\pstartref</code>	46

Q

<code>\quotation</code>	1
<code>\quote</code>	1

R

<code>\raggedX</code>	38
<code>\raw@text</code>	<u>1</u>
<code>\rbracket</code>	<u>1</u>
<code>\read@linelist</code>	<u>1</u>
<code>\ref</code>	47
<code>\Relax</code>	<u>1</u>
<code>\reledmac@error</code>	<u>1</u>
<code>\reledmac@warning</code>	<u>1</u>
<code>\removehboxes</code>	<u>1</u>
<code>\resetprevline@</code>	<u>1</u> , 89
<code>\resetprevpage@</code>	<u>1</u>
<code>\resetprevpage@num</code>	89
<code>\restore@familiarnotes</code>	<u>1</u>
<code>\restore@notes</code>	<u>1</u>
<code>\restore@sidenotes</code>	<u>1</u>
<code>\resumenumbering</code>	<u>1</u> , 18
<code>\rightctab</code>	<u>1</u>
<code>\rightlinenum</code>	<u>1</u> , 20
<code>\rightltab</code>	<u>1</u>
<code>\rightnoteupfalse</code>	50
<code>\rightrtab</code>	<u>1</u>
<code>\rightstartnum</code>	<u>1</u>
<code>\rigidbalance</code>	<u>1</u>
<code>\rigidbalanceX</code>	<u>1</u>
<code>\rtab</code>	<u>1</u>
<code>\rtabtext</code>	<u>1</u>

S

Sacrobosco	12
<code>\sameword</code>	<u>1</u> , 26
<code>\sameword@inedtext</code>	<u>1</u>
Schöpf, Rainer	12
<code>\section@num</code>	<u>1</u>
<code>\select@lemmafонт</code>	<u>1</u>
<code>\select@lemmafонт</code>	<u>1</u> , 41
<code>\SEref</code>	<u>1</u> , 47
<code>\SErefonlypage</code>	48
<code>\SErefwithpage</code>	<u>1</u> , 48
<code>\series</code>	<u>1</u>
<code>\seriesatbegin</code>	<u>1</u> , 29
<code>\seriesatend</code>	<u>1</u> , 29
<code>\set@line</code>	<u>1</u>
<code>\set@line@action</code>	<u>1</u>
<code>\setapprefprefixmore</code>	48
<code>\setapprefprefixsingle</code>	48
<code>\setcommand@series</code>	<u>1</u>
<code>\sethangingsymbol</code>	<u>1</u> , 43
<code>\setistwofollowinglines</code>	<u>1</u>
<code>\setl@dlp@rbox</code>	<u>1</u>

<code>\setl@drpr@box</code>	1
<code>\setline</code>	1, 20
<code>\setlinenum</code>	1, 21
<code>\setmcellcenter</code>	1
<code>\setmcellleft</code>	1
<code>\setmcellright</code>	1
<code>\setmrowcenter</code>	1
<code>\setmrowleft</code>	1
<code>\setmrowright</code>	1
<code>\setnoteswidthliketwocolumnsX@</code>	1
<code>\setnotesXpositionliketwocolumns@</code>	1
<code>\setprintendlines</code>	1
<code>\setprintlines</code>	1
<code>\setSErefonlypageprefixmore</code>	48
<code>\setSErefonlypageprefixsingle</code>	48
<code>\setSErefprefixmore</code>	48
<code>\setSErefprefixsingle</code>	48
<code>\setsidenotessep</code>	51
<code>\setstanzaindents</code>	1, 41
<code>\setstanzapenalties</code>	1, 43
<code>\setstanzavalues</code>	1
<code>\settccllcenter</code>	1
<code>\settccllleft</code>	1
<code>\settccllright</code>	1
<code>\settoggle@series</code>	1
<code>\setthrowcenter</code>	1
<code>\setthrowleft</code>	1
<code>\setthrowright</code>	1
<code>\setXnotespositionliketwocolumns@</code>	1
<code>\setXnoteswidthliketwocolumns@</code>	1
<code>\showlemma</code>	1, 59
<code>\showwordrank</code>	1, 28
<code>\sidenote@margin</code>	1
<code>\sidenotemargin</code>	1, 50
<code>\sidepstartnumtrue</code>	17
<code>\skip@lockoff</code>	1
<code>\skipnumbering</code>	1, 21
<code>\splitoff</code>	1
<code>\spreadmath</code>	1, 55
<code>\spreadtext</code>	1, 55
<code>\stanza</code>	1, 41
<code>\stanza@count</code>	1
<code>\stanza@chang</code>	1
<code>\stanza@line</code>	1
<code>\stanzaindent</code>	1, 43
<code>\stanzaindent*</code>	1, 43
<code>\stanzaindentbase</code>	1, 41
<code>\stanzanumwrapper</code>	1, 44
<code>\startlock</code>	1, 20
<code>\startsub</code>	1, 20

<code>\stepl@dcolcount</code>	1
<code>\strip@szacnt</code>	1
<code>\sub@action</code>	1
<code>\sub@lock</code>	1
<code>\sub@off</code>	1
<code>\sub@on</code>	1
<code>\subline@num</code>	1
<code>\sublinenumberstyle</code>	1, 21
<code>\sublinenumincrement</code>	1, 19
<code>\sublinenumr@p</code>	1
<code>\sublinenumrep</code>	1
<code>\sublineref</code>	1, 46
<code>\sublines@false</code>	1
<code>\sublines@true</code>	1
<code>\sublock@disp</code>	1
<code>\sublockdisp</code>	1
Sullivan, Wayne	12, 13, 60, 73, 77, 153, 155, 231, 265
<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>	28
Theodosius	12
<code>\thepageline</code>	1
<code>\thepstart</code>	1, 17
<code>\thestanza</code>	1, 44
<code>\thestartpageline</code>	1
<code>\this@line@list@version</code>	1
<code>\threecolfootfmt</code>	1
<code>\threecolfootfmtX</code>	1
<code>\threecolfootgroup</code>	1
<code>\threecolfootgroupX</code>	1
<code>\threecolfootsetup</code>	1
<code>\threecolfootsetupX</code>	1
<code>\threecolvfootnote</code>	1
<code>\threecolvfootnoteX</code>	1
<code>\twocolfootfmt</code>	1
<code>\twocolfootfmtX</code>	1
<code>\twocolfootgroup</code>	1
<code>\twocolfootgroupX</code>	1
<code>\twocolfootsetup</code>	1
<code>\twocolfootsetupX</code>	1
<code>\twocolvfootnote</code>	1
<code>\twocolvfootnoteX</code>	1

U

\unvxxX 1

V

Vamana 12
 \variab 1
 \vbfnoteX 1
 \vl@dbfnote 1
 \vl@dcsnote 1
 \vl@dlsnote 1
 \vl@drsnote 1
 \vnumfootnoteX 1

W

Whitney, Ron 12
 \widthX 40
 \wrap@edcrossref 1
 \wrapped@bodyfootmarkX 1
 \wrapped@footfootmarkX 1
 Wujastyk, Dominik 11

X

\X@doreinfeet 1
 \Xafterlemmaseparator 35
 \Xafternote 38
 \Xafternumber 33
 \Xafterrule 39
 \Xaftersymmlinenum 33
 \Xarrangement 1, 30
 \Xarrangement@normal 1
 \Xarrangement@paragraph 1
 \Xarrangement@threecol 1
 \Xarrangement@twocol 1
 \Xbeforelemmaseparator 35
 \Xbeforenotes 39
 \Xbeforenumber 31, 33
 \Xbeforesymmlinenum 33
 \Xbhookgroup 39
 \Xbhooknote 37
 \Xboxlinenum 34
 \Xboxlinenumalign 34
 \Xboxsymmlinenum 34
 \Xcolalign 37
 \Xdo@feet 1
 \xedindex 1
 \xedlabel 1
 \xedtext 1
 \Xendaftererenum 33
 \Xendafterlemmaseparator 36
 \Xendafternote 40

\Xendafterpagenumber	35
\Xendaftersymlinenum	33
\Xendahookinplaceofnumber	35
\Xendahooklinenum	35
\Xendbeforelemmaseparator	36
\Xendbeforenumber	33
\Xendbeforepagenumber	35
\Xendbeforesymlinenum	33
\Xendbhookinplaceofnumber	35
\Xendbhooklinenum	35
\Xendbhooknote	37
\Xendboxendlinenumalign	34
\Xendboxlinenum	34
\Xendboxlinenumalign	34
\Xendboxstartlinenumalign	34
\Xendboxsymlinenum	34
\Xendhangindent	37
\Xendinplaceofflemmaseparator	36
\Xendinplaceofnumber	34
\Xendlemmadisablefontselection	36
\Xendlemmafont	36
\Xendlemmaseparator	36
\Xendlineprefixmore	35
\Xendlineprefixsingle	35
\Xendlinerrangeseparator	31
\Xendmorethantwolines	32
\Xendnonumber	32
\Xendnotefontsize	37
\Xendnotenumfont	36
\Xendnumberonlyfirstinline	31
\Xendnumberonlyfirstintwolines	31
\Xendparagraph	40
\Xendsep	40
\Xendsublinesep	33
\Xendsymlinenum	31
\Xendtwolines	32
\Xendtwolinesbutnotmore	32
\xflagref	1
\Xhangindent	37
\Xhsizethreecol	38
\Xhsizetwocol	38
\Xinplaceofflemmaseparator	35
\Xinplaceofnumber	34
\Xinsertparafootsep	1
\Xledsetnormalparstuff	1
\xleft@appenditem	1
\Xlemmadisablefontselection	36
\Xlemmafont	36
\Xlemmaseparator	35
\Xlinerrangeseparator	31

\xlineref	1, 46
\Xmaxhnotes	39
\Xmorethantwolines	31
\Xnolemmaseparator	1, 35
\Xnonbreakableafternumber	33
\Xnonumber	32
\Xnotefontsize	37
\Xnotenumfont	36
\Xnoteswidthliketwocolumns	40
\Xnumberonlyfirstinline	31
\Xnumberonlyfirstintwolines	31
\Xonlypstart	33
\xpageref	1, 46
\Xparafootsep	38
\Xparindent	37
\Xpstart	32
\Xpstarteverytime	32
\xpstartref	1, 46
\XR@test	1
\XR@test@mac	1
\XR@test@mac@test	1
\Xragged	38
\xright@appenditem	1
\Xrigidbalance	1
\Xstanza	33
\Xstanzaseparator	33
\xsublineref	1, 46
\Xsublinesep	33
\Xsymlinenum	31
\Xtwolines	31
\Xtwolinesonlyinsamepage	32
\Xtxtbeforenotes	39
\Xunvxh	1
\Xwidth	40
\xxref	1, 47

Z

\zz@@@	1
--------------	---

Change History

v0.1.0.	
General: First public release	1
v0.2.0.	
General: Added tabmac code, and extended indexing	1
\ifl@dmemoir: Added \ifl@dmemoir for memoir class having been used	66
\morenoexpands: Added \l@dtabnoexpands to \no@expands	111
\reledmac@error: Added \eledmac@error and replaced error messages	67
v0.2.1.	
\@lab: Removed page setting from \@lab	233
General: Added text about normal labeling	47
Bug fixes and match with mempatch v1.8	1
Major changes to insert code when memoir is loaded	228
\dextrafeet: Renamed \dextrafeet to \l@ddextrafeet	225
\edlabel: Tweaked \edlabel to get correct page numbers	231
\l@ddodoreinxtrafeet: Renamed \dodoreinxtrafeet to \l@ddodoreinxtrafeet	226
\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	111
\zz@@@: Minor change to \zz@@@	230
v0.2.2.	
General: Improved paragraph footnotes	1
New Dekker example	1
Used \providecommand for \@gobblethree to avoid clash with the amsfonts pack- age	72
\footfudgefiddle: Added \footfudgefiddle	152
\line@list@stuff: Added initial write of page number in \line@list@stuff	104
\para@footsetup: Added \footfudgefiddle to \para@footsetup	152
\para@footsetupX: Added \footfudgefiddle to \para@footsetupX	189
v0.3.0.	
\@lab: Replaced \the\line@num by \linenumr@p\line@num in \@lab, and similar for sub-lines	233
\@nl@reg: Added a bunch of code to \@nl for handling \setlinenum	93
General: Includes edstanza and more	1
\ledlinenum: Added \linenumr@p and \sublinenumr@p to \leftlinenum and \rightlinenum	85
\linenumberlist: Added \linenumberlist mechanism	73
\printendlines: Added \linenumr@p and \sublinenumr@p to \printendlines	205
\printlines: Added \linenumr@p and \sublinenumr@p to \printlines	173
\sublinenumr@p: Added \linenumberstyle and \sublinenumberstyle	84
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 criti- cal footnotes	252
General: Added final/draft options	64
Added ledgroup environment	253
Added ledgroupsized environment	254
Added minipage, etc., support	1

\edtext: Added \showlemma to \edtext	111
\l@dfeetendmini: Added \l@dfeetbeginmini, \l@dfeetendmini and all their supporting code	251
\mpnormalfootgroup: Added \mpnormalfootgroup	150
\mpnormalvfootnote: Added \mpnormalvfootnote	148
\showlemma: Added \showlemma	72
\Xarrangement@normal: Added minpage footnote setup to \footnormal	148
v0.4.1.	
General: Added code for changing \@docclearpage	229
Not released. Minor editorial improvements and code tweaks	1
Only change \@footnotetext and \@footnotemark if memoir not used	175
\edindex: Let eledmac take advantage of memoir's indexing	258
\print@Xnotes: Added \@opXfeet	226
\Xdo@feet: Changed \Xdo@feet code for easier extensions	226
v0.5.0.	
\@footnotetext: Enabled regular \footnote in numbered text	176
\@xympar: Eliminated \marginpar disturbance	244
General: Added left and right side notes	244
Added sidenotes, familiar footnotes in numbered text	1
v0.5.1.	
General: Added moveable side note	244
Fixed right line numbers killed in v0.5	1
Only change \hsize in ledgroupsized environment otherwise page number can be in wrong place	254
\affixline@num: Changed \affixline@num to cater for sidenotes	135
\l@dgetsidenote@margin: Added \sidenotemargin and \sidenote@margin	245
v0.6.0.	
\@lopR: Added \@pend, \@pendR, \@lopL and \@lopR in anticipation of parallel processing	95
\@nl@reg: Added \fix@page to \@nl	93
Extended \@nl to include the page number	93
General: Fixed long paragraphs looping	1
Fixed minor typos	1
Prepared for eledpar package	1
\fix@page: Added \last@page@num and \fix@page	94
\get@thisfootnote: Changed \l@dbfnote and \vl@dbfnote as originals could give incorrect markers in the footnotes	176
\new@line: Extended \new@line to output page numbers	104
v0.7.0.	
\@nl@reg: Added \@nl@reg	93
\@ref@reg: Added \@ref@reg	101
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	67
Replaced all \interAfootnotelinepenalty, etc., by just \interfootnotelinepenalty	1
Tidying up for eledpar and ledarab packages	1
\affixline@num: Added skipnumering to \affixline@num	135
\do@actions@fixedcode: Added \do@actions@fixedcode	134

\do@actions@next: Added number skipping to \do@actions	133
\do@insidelinehook: Added \do@linehook for use in \do@line	131
\endnumbering: Changed \endnumbering foreledpar	76
\f@x@l@cks: Added \ch@cksub@l@ck, \ch@ck@l@ck and \f@x@l@cks	138
\footsplitskips: Added \footsplitskips for use in many footnote styles	146
\get@linelistfile: Added \get@linelistfile	91
\initnumbering@reg: Added \initnumbering@reg	75
\l@advance@parledgroup@beforenormalnotes: Added \l@dunboxmpfoot containing some common code	252
\l@dcstotext@r: Added \l@emptyd@ta	131
\l@dgetline@margin: Added \l@dgetline@margin	81
\l@dgetlock@disp: Added \l@dgetlock@disp	83
\l@dgetsidenote@margin: Added \l@dgetsidenote@margin	245
\l@dnumpstartsL: Added \l@dnumpstartsL, \ifl@dpairing and \ifpst@rted for/fromeledpar	73
\l@drsn@te: Added \l@dlsn@te and \l@drsn@te for use in \do@line	131
\l@dzeropenalties: Added \l@dzeropenalties	127
\ledlinenum: Added \ledlinenum for use by \leftlinenum and \rightlinenum	85
\line@list@stuff: Deleted \page@start from \line@list@stuff	104
\list@clearing@reg: Added \list@clearing@reg	91
\n@num: Added \n@num	100
\normalbfnoteX: Removed extraneous space from \normalbfnoteX	181
\resumenummering: Changed \resumenummering foreledpar	77
\setprintendlines: Added \setprintendlines for use by \printendlines	203
\setprintlines: Added \setprintlines for use by \printlines	170
\skipnumbering: Added \skipnumbering and supports	107
\sublinenumincrement: Added \firstlinenum, \linenumincrement, \firstsublinenum and \linenumincrement	82
\sublinenumr@p: Using \linenumrep instead of \linenumr@p	84
Using \sublinenumrep instead of \sublinenumr@p	84
\vnumfootnoteX: Removed extraneous space from \vnumfootnoteX	183
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.	
General: For compatibility witheledpar, possibility to use \autopar on the right side.	1
Possibility to number \pstart.	17
Possibility to number the pstart with the commands \numberpstarttrue.	1

\l@dnumpstartsL: Added \ifledRcol and \ifnumberingR for/from eledpar	73
v0.12.1.	
General: Don't number \pstarts of stanza.	1
The numbering of \pstarts restarts on each \beginnumbering.	1
v0.13.0.	
General: New stanzaindentsrepetition counter to repeat stanza indents every n verses. . .	42
New stanzaindentsrepetition counter: to repeat stanza indents every n verses.	1
\managestanza@modulo: New stanzaindentsrepetition counter to repeat stanza indents every n verses.	267
v0.13.1.	
General: \thepstartL and \thepstartR use now \bfseries and not \bf, which is deprecated and makes conflicts with memoir class.	1
v0.14.0.	
General: Tweaked \edlabel to get correct line number if the command is first element of a paragraph.	1
\edlabel: Tweaked \edlabel to get correct line number if the command is first element of a paragraph.	231
v0.15.0.	
General: Line numbering can be reset at each pstart.	79
Possibility to print \pstart number inside.	17
\affixline@num: Line numbering can be disabled.	135
\ifinserthangingsymbol: New management of hangingsymbol insertion, preventing undesirable insertions.	265
\printlines: Line numbering can be reset at each pstart.	172
v0.17.0.	
\ifinserthangingsymbol: New new management of hangingsymbol insertion, pre- venting undesirable insertions.	265
v1.0.0.	
General: \lemma can contain commands.	24
Debug in lineation command	19
New generic commands to customize footnote display.	30, 217
Options nonum and noseup in \Xfootnote.	23
Options of \Xfootnotes.	144
Possibility to have commands in sidenotes.	50
Some compatibility break with eledmac. Change of name: eledmac.	1
\morenoexpands: Change to be compatible with new features	111
v1.0.1.	
General: Correction on \Xnumberonlyfirstinline with lineation by pstart or by page. 31	
v1.1.0.	
General: Add \labelpstarttrue.	17
Add \Xnumberonlyfirstintwolines	31
Add \Xpstart and \Xonlypstart	32
New hook to add arbitrary code at the beginning of the notes	37
New options for block of notes.	39
New package option: parapparatus.	1
New tools to change order of series	216
Sectioning commands.	57
\preXnotes: New skip \preXnotes@	197
\settoggle@series: \settoggle@series switch the global value of the toggle, not only the local value.	217

v1.2.0.	
\endquote:	Compatibility of \ledchapter with the <i>memoir</i> class. 293
\preXnotes:	Debug in familiar footnotes (bug introduced by v1.1). 197
v1.3.0.	
\endquote:	<i>Quotation</i> and quote environment inside numbered sections. 293
v1.4.0.	
General:	Compatibility with LuaTeX of RTL notes. 1
\edtext:	Compatibility of \edtext with the right-to-left direction (with Polyglossia). 111
\ledsetnormalparstuffX:	Direction of footnotes with polyglossia. 193
\newseries@:	Remembers the language of the lemma, in order to create a correct direction for the footnote separator. 209
\rbracket:	Switch the right bracket to a left bracket when the lemma is RTL (needs polyglossia or LuaTeX). 165
v1.4.1.	
\affixside@note:	Remove spurious spaces. 249
\endquote:	New option <i>noquotation</i> 293
\get@thisfootnote:	Compatibility of standard footnotes with <i>eledmac</i> when these footnotes contain any commands. 176
\labelrefsparsesubline:	Fix bug with \edlabel. 232
v1.4.2.	
General:	Debug with some special classes. 1
v1.4.3.	
General:	Add \Xnonbreakableafternumber. 33
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. 293
v1.4.8.	
General:	Corrects a bug with parallel texts introduced by 1.1. 1
v1.4.9.	
\normalbfnoteX:	Allow to redefine \thefootnoteX with <i>alph</i> when some packages are loaded. 181
v1.5.0.	
General:	Correct indexing when the call is made in critical notes. 255
\do@insidelinehook:	Added \do@insidelinehook for use in \do@line 131
\edindex:	Compatibility with <i>imakeidx</i> package, and possibility to use multiple index with \edindex. 258
v1.5.1.	
\managestanza@modulo:	Correct stanzaindentsrepetition counter 267
\normalvfootnoteX:	Fix bug with normal familiar footnotes when mixing RTL and LTR text. 179
v1.6.0.	
\newverse:	Add \falseverse macro. 269
v1.6.1.	
General:	Corrects a false hanging verse when a verse is exactly the length of a line. 1

\AtEveryPstart: Spurious space in \pstart.	124
\ifinserthangingsymbol: Hang verse is now not automatically flush right.	265
\l@dunhbox@line: Move the call to \inserthangingsymbol to allow use \hfill in-	
side.	128
\pend: Spurious space in \pend.	125
v1.7.0.	
General: New features for managing page breaks.	58
v1.8.0.	
General: Compatibility with parledgroup option of eledpar package.	1
If imakeidx and hyperref are loaded, adds hyperref in the index.	255
\endquote: Correction of sectioning commands in parallel texts.	293
\get@index@command: Debug \get@index@command and compatibility with hyperref	
package.	257
\newhookcommand@series@reload: Debug \beforenotesX and \maxhnotesX which	
did not work.	220
\prevpage@num: Correct \parafootsep when using with ledgroup.	158
v1.8.1.	
General: Debug endnotes when more than one series is used (change the position where	
tools for endnotes are defined).	198
v1.8.2.	
General: Debug compatibility problem with hebrew option of babel package.	1
v1.8.3.	
General: Fixes spurious spaces added by v1.7.0.	1
v1.8.5.	
General: Debug indexing in right column, with eledpar.	255
v1.9.0.	
\doxtrafeet: Add \fnpos to choice the order of footnotes.	225
\l@dfeetendmini: Add \mpfnpos to choice the order of footnotes in minipage / led-	
group.	250
v1.10.0.	
General: Add \pstartref and \xpstartref to refer to a pstart number (extension of	
\edlabel).	1
\endquote: Correction of sectioning commands in parallel texts.	293
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.	36
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.	92
General: Add \ledinnernote and \ledouternote commands.	50
Add \Xendparagraph and related settings.	40
Add hyperlink to crossref (needs hyperref package).	45
Compatibility with musixtex.	1
Debug eledmac sectioning command after using \resumenumbering.	1
Ensure that imakeidx is loaded <i>before</i> eledmac	255
New hooks: \Xafterrule and \afterruleX	39
New options for ragged-paragraph notes	38
New sectioning commands.	57
Optional arguments for \pstart and \pend.	17
\AtEveryPstart: New optional argument for \pstart, to execute code before it. . .	124
\edindex: Use correctly default index when imakeidx is loaded.	258
\endquote: \ledxxx sectioning commands are deprecated and replaced by \eledxxx commands.	293
\initnumbering@reg: \beginnumbering is defined only on eledmac, not on eledpar. .	75
\l@dcnote: \l@dlsnote, \l@drsnote and \l@dcnote defined only one time, in eledmac, including needs for eledpar case.	246
\l@dgetsidenote@margin: \sidenotemargin is now directly defined in eledmac to be able to manage eledpar.	245
\l@dnumpstartsL: Add \ifledRcol@ for eledpar	73
\l@dunhbox@line: \do@line is split in more little commands.	129
\newhookcommand@series@reload: Debug \beforenotesX and \maxhnotesX which did not work when called after \footparagraphX.	220
Debug \Xbeforenotes and \Xmaxhnotes which did not work when called after \footparagraph.	220
\pend: New optional argument for \pend, to execute code after it.	125
\stanza: &can have an optional argument: content to be printed after.	269
\Stanza can have an optional argument: content to be printed before.	269
Add \newverse macro, \falseverse deprecated.	269

v1.12.1.

\wrap@edcrossref: Fix spurious spaces.	235
--	-----

v1.12.2.

\l@dunhbox@line: Fix a bug with critical notes at the tops of pages (added by v12.0.0)	128
--	-----

v1.12.3.

General: Add macros for new messages since v0.7	67
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.	50
Debug left and right notes (bugs added by 1.12.0)	1
Underline lemma in \eledxxx when using draft mode.	1
\flag@end: \flag@start and \flag@end are now defined only one time for eledmac and eledpar	105
\flag@start send a error message when a \edtext is done without insert (note)	105
\reledmac@error: Replaced error messages	67

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	260
\@ssect: Debug \eledchapter in parallel with memoir	297
\doinsidelinehook: Added \dolinehook and \doinsidelinehook	131
\endnumbering: Allow to mix parallel columns and normal text when using \pausenumbering	76
\l@dgobblearg: \l@dgobblearg becomes \l@gobbeloptarg	277
\l@drestoreforedtext: Debug optional arguments of \Xfootnote in tabular context	277
\resumenumbering: Debug \resumenumbering	77
v1.12.7.	
\wrap@edcrossref: \wrap@edcrossref is now robust	235
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	105
v1.13.0.	
General: Add \Xnoteswidthliketwocolumns and \noteswidthliketwocolumnsX	40
Added widthliketwocolumns option	64
\newhooktoggle@series@reload: Add \newhookcommand@toggle@reload	220
\para@footsetupX: In \para@footsetupX, use \columnwidth instead of \hsize	189
\settoggle@series: \settoggle@series can take an optional arguments to reload series setup.	217
v1.13.1.	
General: Coming back of page and line breaking penalties's management, deleted by error in v0.17.	1
Debug quotation environment inside of a \pstart preceded by a sectioning command.	1
\thepstart: Add \l@dzeropenalties in \pstart	124
v1.13.2.	
General: Fix bug with normal footnotes, added by v1.13.0.	1
\l@dnumpstartsL: Add \ifl@dpaging for eledpar	73
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.	
General: Debug spurious characters before endnotes.	198
Delete previous override of \l@d@@wrindexhyp at the beginning of a document when hyperref is not loaded.	262
Move gobbling command	72
Provide \@gobblefour	72
\edindex: Let eledmac take advantage of imakeidx even when memoir class is used	258
v1.14.1.	
\@ssect: Debug sectioning commands when using both handout and hyperref package.	300
v1.14.2.	
\@ssect: Debug \edtext after starred sectioning commands when using memoir class.	297
v1.15.0.	
\@edtext@level: New boolean \if@edtext@.	111
General: Fix bug with footnotes layout when using some options of the geometry package (bug add by v1.13.0).	1
New commands \AtEveryPstart and \AtEveryPend.	17

New tools to prevent ambiguous references in lemma	26
<code>\arrangementX@threecol</code> : Correct bug with paragraphed familiar footnotes setting.	188
<code>\endsub</code> : Restore subline feature (disabled by mistake in v1.8.0).	106
<code>\if@lemmacommand@</code> : New boolean <code>\iflemmacommand@</code>	116
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.	104
v1.16.0.	
General: <code>\edtext</code> is now defined only in <code>eledmac</code> , not in <code>eledpar</code> . Debug wrong num- bering when using <code>\sameword + eledpar + \tag</code> command.	111
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> . <code>Eledmac</code> provides <code>\edlineref</code> instead.	236
v1.17.0.	
<code>\edtext</code> : Error message when calling <code>\edtext</code> outside of a numbered paragraph.	111
v1.18.0.	
<code>@edindex@hyperref</code> : Fix spurious space with <code>\edindex</code> when using <code>imakeidx/indextools</code> + <code>hyperref</code>	260
General: Add <code>\Xstarteverytime</code>	32
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.	32
<code>\edlabel</code> : <code>\edlabel</code> is now defined only one time for both <code>eledmac</code> and <code>eledpar</code>	231
<code>\l@d@section</code> : Option <code>parapparatus</code> works for endnotes.	199
<code>\l@dnumpstartsl</code> : Add <code>\ifl@dprintingpages</code> and <code>\dprintingcolumns</code> for <code>eledpar</code>	73
<code>\print@line</code> : Compatibility with Lua \TeX RTL languages.	129
<code>\printlinefootnote</code> : Code refactoring in <code>\printlinefootnote</code> : the printing of the numbers are factorized in <code>\printlinefootnotearea</code>	166
<code>\printpstart</code> : Debug <code>\Xpstart</code> with parallel pages and columns (<code>eledpar</code>)	165
v1.19.0.	
General: <code>\Xmaxhnotes</code> and <code>\maxhnotesX</code> work now for both two-columns and three- columns setting.	1
Compatibility with <code>eledpar</code> v1.13.0.	1
<code>\footsplitskips</code> : <code>\footsplitskips</code> doesn't set <code>\floatingpenalty</code> to <code>\@MM</code> when processing parallel pages.	146
<code>\xxref</code> : <code>\xxref</code> works also with right side numbers, when <code>\@Rlineflag</code> is not empty.	237
v1.19.1.	
General: Call <code>\correct@footinsX@box</code> and <code>\correct@Xfootins@box</code> directly in <code>\print@notesX@forpages</code> and <code>\print@Xnotes@forpages</code> , that is in <code>eledpar</code>	1
v1.20.0.	
General: Add <code>\Xendboxlinenum</code>	34
Add <code>\Xtwolines</code> and <code>\Xmorethantwolines</code> hooks	31
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 Lua \TeX	1

Fix issues with RTL text in notes when using Lua \TeX .	1
Options fulllines in \backslash Xfootnote.	23
The \backslash newifs are not followed by boolean values set to false, because it is the \TeX default setting.	1
\backslash printlines: Added \backslash ifl@d@Xmorethantwoline and \backslash ifl@d@Xmorethantwoline to \backslash printlines	173
\backslash stanza: & and \backslash & can be preceded by spaces.	269
\backslash xxref: Debug \backslash xxref when not loading eledpar (fix bug added in 1.19.0).	237
v1.21.0.	
\backslash @edindex@hyperref: Look at the hyperindex option of hyperref before inserting hyperref	260
General: \backslash AtEveryPstart and \backslash AtEveryPend are now compatible with \backslash autopar	1
\backslash Xafterrule and \backslash afterruleX features no longer create problems of overflowing at the bottom of the page.	1
\backslash chapter inside optional argument of \backslash pstart works when typesetting parallel pages	1
\backslash preXnotes and \backslash prenotesX features no longer create problems of overflowing at the bottom of the page.	1
\backslash seriesatbegin and \backslash seriesatbegin more efficient	216
Add \backslash applabel and related	48
Add \backslash beforenotesX and \backslash Xbeforenotes features for notes set in two and three column.	1
Add \backslash hidenumbering	21
Add \backslash Xcolalign and \backslash colalignX hooks	37
Add \backslash Xendtwoline, \backslash Xendmorethantwoline, \backslash Xendtwolinebutnotmore and \backslash Xendtwolineonlyinsamepage.	32
Add \backslash Xparindent and \backslash hangindentX	37
Add \backslash Xtwolinebutnotmore and \backslash Xtwolineonlyinsamepage.	1
Add nocritical, noend, nofamiliar and noledgroup options.	1
Add noeledsec package option	1
Debug \backslash beforenotesX \backslash maxnotesX \backslash noteswidthliketwocolumnsX and \backslash afterruleX with footnotes set in two and three columns.	1
Fix bug when a \backslash Xfootnote follows a \backslash Xendnote in the second argument of \backslash edtext (bug added in eledmac 1.0.0).	1
Fix bug with \backslash maxnotesX when using \backslash foottwocolX or \backslash footthreecolX.	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 \backslash doendnotes. oldprintnpnumspace option allows to come back to previous setting	1
parapparatus option works now with familiar footnotes.	1
Provide \backslash @gobblefive	72
\backslash l@d@section: \backslash endnotes take five arguments.	199
\backslash ledinnotemark: Add \backslash ledinnotemark.	258
\backslash ledsetnormalparstuffX: \backslash ledsetnormalparstuff is deprecated and becomes \backslash ledsetnormalparstuffX and \backslash Xledsetnormalparstuff.	193
\backslash n@num: \backslash n@num@ref deleted	100
\backslash n@num defined only one time for both Eledmac and Eledpar	100
\backslash newhookcommand@series: \backslash newhookcommand@series can take an optional argument.	219
\backslash newhooktoggle@series: \backslash newhooktoggle@series can take an optional argument.	220

\print@footnoteXrule: Code refactoring: the spaces after the footnote rules are directly managed in \print@Xfootnoterule and \print@footnoteXrule	195
\seriesatend: Fix spurious space in \seriesatend	216
\skipnumbering: \skipnumbering defined only one time for both Eledmac and Eledpar.	107
Correct \skipnumbering for stanza.	107
Delete \skipnumbering@reg.	107
v1.22.0.	
General: Add \doendnotesbysection command.	24
Add option for lemma separator inside endnotes	36
Adds hyperlink for references to notes in indices.	1
Fix conflict between noend package option and edtabularx environments	1
Provides support for xindy.	1
Standardize endnotes handbook.	24
When using hyperref package, internal links in index or with \edlineref are now targeted to the top and not longer to the bottom of the lines they refer to.	1
\ledinnote: \ledinnote takes a first optional argument, which is the label for hyperlinks.	258
v1.22.1.	
General: Fix bug (added on v1.22.0) with \Xinplaceofnumber hook.	1
\prevpage@num: Correct double symbol when using both \parafootsep and \Xsymlinenum.	158
v1.23.0.	
\@edtext@level: The boolean \if@edtext@ becomes the counter \edtext@level.	111
General: Add \Xboxlinenumalign and \Xendboxlinenumalign.	34
Add \Xboxstartlinenum, \Xendboxstartlinenum, \Xboxendlinenum, \Xendboxendlinenum.	34
Allow use of \sameword with inputenc managing of UTF-8.	1
Compatibility between nofamiliar/nocriticals option and minipage/ledgroup.	1
Error message when using \beginnumbering... \endnumbering without \pstart.	1
Fix bug with \sameword when the lemma overlaps multiple line.	26
Fix bug with \sameword when the same lemma is used for multiple notes or for nested \edtexts.	26
Fix bug with \skipnumbering called immediately after a \pstart.	1
Fix error of \iftrue not closed.	1
Fix spurious space with \skipnumbering (bug added on v1.21.0).	1
New tools to ensure the line-list file uses the right version of commands when upgrading the eledmac version.	1
Optional argument of \sameword can be a comma-separated list of \edtext depth.	26
\lemma: Fix spurious space after \lemma command	116
\newseries@: Prevent spurious spaces when \Afootnote and similar commands are followed by spaces (bug added on 1.0.0).	209
\sameword: In order to allow use of \sameword with inputenc, we detokenize its mandatory argument before using it in control sequence names.	120
\Serefwithpage: Debug \Xendtwolines, \Xendmoreethantwolines, \Xendtwolinesbutnotmore and \Xendtwolinesonlyinsamepage when using \apprefwithpage.	240
v1.23.1.	
General: Fix bug with \lemma command in the right side.	1
v1.23.2.	
General: Compatibility with L ^A T _E X'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 ‘nocritical’ option.	1
v1.24.2.	
General: Fix incompatibility between ‘nofamiliar’ option and ‘memoir’ package.	1
v1.24.3.	
General: Restore marginal numbers and notes with sectioning command (bug introduced in v1.21.0)	1
v1.24.4.	
General: Fix spurious space with <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.	252
<code>\@xympar</code> : Patching <code>\@xympar</code> instead of redefining it	244
General: <code>\@makecol</code> , <code>\@reinserts</code> and <code>\@doclearpage</code> are patched instead of begin redefined	228
<code>\doextrafeeti</code> becomes <code>\do@feetX</code> ; <code>\doextrafeetii</code> becomes <code>\Xdo@feet</code> ; <code>\@opextrafeeti</code> becomes <code>\@opfeetX</code> ; <code>\doreinxtrafeetii</code> becomes <code>\X@doreinfeet</code> ; <code>\doreinxtrafeeti</code> becomes <code>\@doreinfeetX</code>	228
Add <code>\Xendinplaceofnumber</code> hook.	1
Add <code>\Xendnonumber</code> hook.	1
Add nonum 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	175
Suppress indexing command specific to <code>memoir</code>	258
<code>\endminipage</code> : Patch <code>\endminipage</code> instead of redefining it.	252
<code>\initnumbering@quote</code> : <code>\initnumbering@sectcmd</code> becomes <code>\initnumbering@quote</code>	293
<code>\l@advance@parledgroup@beforenormalnotes</code> : Some conde of <code>\l@dumboxmpfoot</code> moved to <code>\l@advance@parledegroupp@beforenormalnotes</code>	252
<code>\newseries@</code> : One endnotes file by series.	213
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.	
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
<code>\ledsetnormalparstuffX</code> : Replaced <code>\noindent</code> with <code>\parindent</code> set to 0pt.	193
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>bidi</code> v17.9 and following.	1
Warning message when using some setting commands inside <code>rightside</code> environment (deprecated behavior)	1
v2.3.1.	
General: Fix spurious space when using optional argument of <code>\stanza</code> (introduced in v2.3.0).	1
v2.4.0.	
General: <code>\Xbhooknote</code> and <code>\bhooknoteX</code> work with notes in columns.	1
Fix 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>\Xlinerangeseparator</code> and <code>\Xendlinerangeseparator</code>	31
Option <code>linerangesep</code> for critical footnotes and endnotes.	31
<code>\footnoteoptions@</code> : First argument of <code>\footnoteoption@</code> is now mandatory, not optional.	144

v2.4.1.	
General: Fix bug with <code>\appref</code> and <code>\apprefwithpage</code> (introduced in v2.4.0).	1
Fix bug with tabular environments when using <code>babel</code> or <code>polyglossia</code> languages that override \TeX <code>\roman</code> command, like Greek language.	1
Fix bug with tabular environments when using <code>babel</code> or <code>polyglossia</code> languages that override \TeX <code>\roman</code> command, like Greek.	1
v2.5.0.	
General: <code>\apprefwithpage</code> and <code>\appref</code> print double quotation mark when the label was not defined.	1
<code>\apprefwithpage</code> and <code>\appref</code> work with right side crossref.	1
<code>\apprefwithpage</code> works also when <code>noend</code> option is enabled.	1
<code>\appref</code> and <code>\apprefwithpage</code> can take <code>linrangesep</code> optional argument.	1
<code>\edlabel</code> works now in <code>\Xfootnote</code> .	1
<code>\lemma</code> can be used even when the <code>nocritical</code> is enabled.	1
Compatibility with new hook and tools of <code>reledpar</code> 2.6.0.	1
Fix spurious vertical space in <code>astanza</code> environment (<code>reledpar</code>)	1
Log now states ‘There were undefined references’ when using wrong references in <code>\edlineref</code> or <code>edpageref</code> .	1
New hooks to customize page and line number appearance in endnotes.	1
New hooks: <code>\Xbhookgroup</code> and <code>\bhookgroupX</code> .	1
New tools to easily make cross-reference to a passage defined by a start and an end line	47
<code>\edlabel</code> : Fix bug when calling <code>\edlabel</code> in a footnotes of the rightside	231
<code>\l@d@section</code> : <code>\endnotes</code> take six arguments.	199
<code>\printlines</code> : <code>\printlines</code> takes an eighth argument: the line flag	172
<code>\Serefwithpage</code> : Debug <code>\setapprefprefixsingle</code>	240
<code>\xlineref</code> : <code>\xlineref</code> does not include anymore the side flag. Use <code>\xflagref</code> to get it. Not that <code>\edlineref</code> still contains the flag.	236
v2.6.0.	
General: Adds compatibility with <code>innnote</code> and <code>notenumber</code> options of <code>indextools</code> package.	1
Fix bug with footnote counter in <code>ledgroup</code> (added in v2.5.0).	1
Fix bug, introduced in v2.5.0, with footnote numbering in parallel typesetting when using <code>perpage</code> package.	1
v2.7.0.	
<code>\@k</code> : <code>\rigidbalance</code> is split in <code>\Xrigidbalance</code> and <code>\rigidbalanceX</code> .	158
General: Add dash as default page range separator for <code>\SEonlypage</code>	1
Debug <code>\SErefonlypage</code> when referring to only one page.	1
Delete parenthesis after <code>\SErefonlypage</code> .	1
Fix (again) bugs with footnote numbering in parallel typesetting while using <code>ledgroup</code> environments (bug added in v2.5.0).	1
Fix bug with <code>\Serefwithpage</code> .	1
Fix bugs in compatibility with <code>innote</code> and <code>notenumber</code> options of <code>indextools</code> package, when indexing outside of a <code>ledgroup</code> .	1
New commands to make glossaries connected to page and linenumber with the <code>glossaries</code> package	1
New hooks: <code>\Xhsize</code> and <code>\hsizeX</code>	40
New hooks: <code>\Xlemmafont</code> and <code>\Xendlemmafont</code>	36
New setting commands: <code>\setSErefonlypageprefixsingle</code> and <code>\setSErefonlypageprefixmore</code>	1
Warning for duplicate and undefined labels are parsable by <code>latexmk</code>	1

Warning for duplicate labels does not send any more a false line and page number . . .	1
When using <code>hyperref</code> package, add link in familiar footnotes between the footnote marks in the text and the footnote marks in the footnote	1
When using <code>hyperref</code> package, add links for <code>\SEref</code> and related, <code>\appref</code> and related.	1
When using <code>hyperref</code> package, add links from critical footnotes and critical endnotes to the line of text they refers	1
<code>\l@ddsection: \endnotes</code> take seven arguments.	199
v2.7.1.	
General: Debug <code>\Xhookgroup</code> hooks executed on columnar footnotes (moved to a larger group, to take effect).	1
v2.7.2.	
General: <code>\Xhsize</code> and <code>\hsizeX</code> become <code>\Xwidth</code> and <code>\widthX</code>	40
Fix problem of hyphenation when using <code>hyperref</code> package (added in v2.7.0).	1
v2.8.0.	
General: <code>reledmac</code> cross-referencing can take advantage of <code>xr</code> package.	1
More <code>\edgls...</code> commands.	1
No indentation for paragraphed notes in <code>ledgroup</code> . Can be changed with <code>\Xparindent</code> and <code>\parindentX</code>	1
<code>\l@dsection: \Xendhangindent</code> and <code>\Xendafternote</code> can take values which are relative to the font size of the endnote.	199
v2.8.1.	
General: Warning for undefined labels are really parsable by <code>latexmk</code>	1
v2.8.2.	
General: Fix bug concerning indent in a paragraph immediatly 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 between sectioning command as optional argument of a <code>\pstart</code> and <code>\pstart</code> content	1