

reledmac

Typeset scholarly editions with L^AT_EX^{*}

Maïeul Rouquette[†]

based on the original ledmac by

Peter Wilson

Herries Press

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

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

Abstract

The **reledmac** provides many tools in order to typeset scholarly editions. It is based on the **eledmac** package, which was based on the **ledmac** package, which was based on the **edmac** T_EX package.

It can be used in combination with **reledpar** in order to typeset two texts in parallel, like an original text and its translation in a modern language.

reledmac provides many tools and options. Normally, they are all documented in this file. Also provided is a help folder, “examples”. The folder contains additional examples (although not for every possible case). Examples starting with “1-” are for basic uses, those starting with “2-” are for advanced uses.

To report bugs or request a new feature, please go to ledmac GitHub page and click on “New Issue”: <https://github.com/maieul/ledmac/issues/>. You must create an account on github.com to access my page (maieul/ledmac). GitHub accounts are free for open-source users. You can post messages in English or in French (preferred).

You can subscribe to the **reledmac** mail list at:
<http://geekographie.maieul.net/146>

Contents

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

^{*}This file (**reledmac.dtx**) has version number v2.10.0, last revised 2016/05/08.

[†]maieul at maieul dot net

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

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

7.12 Options for paragraphed footnotes	39
7.12.1 Mark separation of notes	39
7.12.2 Ragged text	40
7.13 Options for block of notes	40
7.13.1 Text before notes	40
7.13.2 Code before notes	40
7.13.3 Spacing	40
7.13.4 Rule	40
7.13.5 Maximum height	41
7.13.6 Width	41
7.14 Footnotes and the <code>reledpar</code> columns	41
7.15 Endnotes in one paragraph	42
8 Fonts	42
9 Verse	43
9.1 Basic	43
9.2 Define stanza indents	43
9.3 Repeating stanza indents	43
9.4 Manual stanza indent	44
9.5 Stanza breaking	44
9.6 Hanging symbol	44
9.7 Long verse and page break	45
9.8 Content before/after verses	45
9.9 Numbering stanza	45
9.10 Various tools	46
9.11 Notes on empty lines	46
10 Grouping	46
11 Cross referencing	47
11.1 Basic use	47
11.2 Cross-referencing to a critical note	47
11.3 Cross-referencing which return a number in any case	47
11.3.1 Cross-referencing in order to define line number of a critical note	48
11.4 Not automatic cross-referencing	48
11.5 Normal \LaTeX cross-referencing	49
11.6 References to start and end lines	49
11.6.1 Reference to main text lines	49
11.6.2 References to lines that are commented on in the apparatus	49
11.6.3 Settings	49
11.7 Compatibility with <code>xr</code> package	51

12 Side notes	51
12.1 Basics	51
12.2 Setting	52
12.2.1 Width	52
12.2.2 Vertical position	52
12.2.3 Distance to the main text	52
12.2.4 Separator between notes	52
13 Indexing	52
13.1 Basics	52
13.2 Referring to critical notes	53
13.3 Separator between page and line numbers	53
13.4 Using xindy	54
13.5 Advanced setting	54
14 Glossary	54
14.1 Preamble setting	55
14.2 Commands	55
15 Tabular material	55
16 Sectioning commands	58
16.1 Sectioning commands without line numbers or critical notes	58
16.2 Sectioning commands with line numbering and critical notes	59
16.3 Optimization	59
17 Quotation environments	60
18 Page breaks	60
18.1 Control page breaking	60
18.2 Prevent page break in a long verses	60
19 Miscellaneous	61
19.1 Known and suspected limitations	61
19.1.1 Non-standard geometry	61
19.1.2 floatrow package compatibility	61
19.1.3 ‘No room for a new’	61
19.1.4 Marginal notes	62
19.1.5 Paragraph shape	62
19.1.6 Paragraphed footnotes	62
19.1.7 Use with other packages	63
19.1.8 Parallel typesetting	64
I Implementation overview	65

II Preliminaries	65
II.1 Links with original edmac	65
II.2 Package declaration	65
II.3 Package options	66
II.4 Loading packages	67
II.5 Compatibility with Lua \TeX	68
II.6 Boolean flags	68
II.7 Messages	69
II.8 Gobbling	74
II.9 Miscellaneous commands	75
II.10 Prepare reledpar	75
II.11 Booleans provided by other optional packages which are required in any case	76
III Sectioning commands	76
IV List macros	80
V Line counting	81
V.1 Choosing the system of lineation	81
V.2 Line number margin	83
V.3 Line number initialization and increment	84
V.4 Line number locking	85
V.5 Line number style	86
V.6 Line number printing	87
V.7 Line number counters and lists	88
V.8 Line number locking counter	89
V.9 Line number associated to lemma	89
V.10 Reading the line-list file	92
V.11 Commands within the line-list file	94
V.12 Writing to the line-list file	105
VI Marking text for notes	110
VI.1 <code>\edtext</code> itself	111
VI.2 Substitute lemma	118
VI.3 Substitute line numbers	119
VI.4 Lemma disambiguation	120
VII Paragraph decomposition and reassembly	126
VII.1 Boxes, counters, <code>\pstart</code> and <code>\pend</code>	126
VII.2 Processing one line	131
VII.2.1 General process	131
VII.2.2 Process for “normal” line	131
VII.2.3 Process for line containing <code>\eledsection</code> command	133
VII.2.4 Hooks	133
VII.2.5 Sidenotes and marginal line number initialization	134

VIII Line and page number computation	134
IX Line number printing	137
X Pstart number printing in side	141
XI Restoring footnotes and penalties	143
XI.1 Add insertions to the vertical list	143
XI.2 Penalties	144
XI.3 Printing leftover notes	145
XII Critical footnotes	146
XII.1 Fonts	146
XII.2 Individual note options	146
XII.3 Notes language	147
XII.4 General survey of the way we manage notes	148
XII.5 General setup	148
XII.6 Footnotes arrangement	149
XII.6.1 User level macro	149
XII.6.2 Normal footnote	149
XII.6.3 Paragraphed footnotes	153
XII.6.4 Columnar footnotes	160
XII.7 Critical notes presentation	166
XII.7.1 Font tools	167
XII.7.2 Pstart number in footnote	167
XII.7.3 Lemma printing	168
XII.7.4 Line number printing	168
XIII Familiar footnotes	177
XIII.1 Adjacent footnotes	177
XIII.2 Regular footnotes for numbered texts	178
XIII.3 Footnote formats	180
XIII.4 Footnote arrangement	181
XIII.4.1 User level macro	181
XIII.4.2 Normal footnotes	181
XIII.4.3 Two columns footnotes	187
XIII.4.4 Three columns footnotes	189
XIII.4.5 Paragraphed footnotes	191
XIII.5 Wrapping footnote marks in hyperlink	195
XIV Code common to both critical and familiar footnote in normal arrangement	196
XV Footnotes' width for two columns	197
XVI Footnotes' order	198

XVII Footnotes' rule	198
XVIII Specific skip for first series of footnotes	199
XVIII.0.1 Overview	199
XVIII.0.2 User level command	200
XVIII.0.3 Internal commands	200
XIX Endnotes	201
XX Generate series of notes	209
XX.1 Test if series is still existing	210
XX.2 Init specific to <code>reledpar</code>	210
XX.3 For critical footnotes	210
XX.3.1 Options	210
XX.3.2 Create inserts, needed to add notes in foot	212
XX.3.3 Create commands for critical apparatus, <code>\Afootnote</code> , <code>\Bfootnote</code> etc.	212
XX.3.4 Set standard display	214
XX.4 For familiar footnotes	214
XX.4.1 Options	215
XX.4.2 Create tools for familiar footnotes (<code>\footnotex</code>)	215
XX.5 The endnotes	217
XX.5.1 The auxiliary file	217
XX.5.2 The main macro	217
XX.5.3 The options	218
XX.6 Init standards series (A,B,C,D,E)	219
XXI Setting series display	219
XXI.1 Change series order	219
XXI.2 Test series order	220
XXI.2.1 Get the first series	220
XXI.3 Series setting	220
XXI.3.1 General way of working	220
XXI.3.2 Tools to set options	221
XXI.3.3 Tools to generate options commands	222
XXI.3.4 Options for critical notes	224
XXI.3.5 Options for familiar notes	225
XXI.3.6 Options for endnotes	226
XXI.4 Hooks for a particular footnote	227
XXI.5 Alias	228
XXII Output routine	229
XXII.0.1 Page number management	229
XXII.0.2 Extra footnotes output	229
XXII.0.3 Standard output's commands patching	231

XXIII Cross referencing	234
XXIII.1 Compatibility with <code>xref</code>	247
XXIV Side notes	247
XXV Minipages and such	254
XXVI Indexing	259
XXVI.1 Looking on package order	259
XXVI.2 Auxiliary macros for <code>\edindex</code>	259
XXVI.3 Code specific to <code>\edindexin</code> critical footnotes	260
XXVI.4 Analysis of command in indexed text	261
XXVI.5 Code for the formatted index	262
XXVI.6 Main code	262
XXVI.7 Hyperlink	263
XXVI.8 ‘innote’ and ‘notenumber’ option of <code>indextols</code> package	266
XXVII Glossaries	267
XXVIII Verse	268
XXVIII.1 Hanging symbol management	268
XXVIII.2 Using <code>&</code> character	269
XXVIII.3 Code category setting	269
XXVIII.4 Stanza count and indent	270
XXVIII.5 Numbering stanza	271
XXVIII.6 Stanza number in note	272
XXVIII.7 Main work	273
XXVIII.8 Restore catcode and penalties	275
XXIX Arrays and tables	275
XXIX.1 Preamble: macro as environment	275
XXIX.2 Tabular environments	279
XXIX.2.1 Disabling and restoring commands	279
XXIX.2.2 Counters, boxes and lengths	282
XXIX.2.3 Tabular typesetting	286
XXIX.2.4 Environments	297
XXX Quotation’s commands	297
XXXI Section’s title commands	299
XXXI.1 Commands to disable some feature	299
XXXI.2 General overview	299
XXXI.3 <code>\beforeeledchapter</code> command	300
XXXI.4 Auxiliary commands	300
XXXI.5 Patching standard commands	301
XXXI.6 Main code of <code>\eledxxx</code> commands	306
XXXI.7 Macros written in the auxiliary file	308

XXXII Page breaking or no page breaking depending of specific lines	310
XXXIII Long verse: prevents being separated by a page break	312
XXXIV Tools for hyperref package	313
XXXV Compatibility with eledmac	313
Appendix A Things to do when changing versions	316
Appendix A.1 Migrating from edmac to ledmac	316
Appendix A.2 Migration from ledmac to eledmac	317
Appendix A.3 Migration to eledmac 1.5.1	318
Appendix A.4 Migration to eledmac 1.12.0	318
Appendix A.5 Migration to eledmac 1.7.1	319
Appendix A.6 Migration to eledmac 1.21.0	319
Appendix A.6.1 \Xledsetnormalparstuffand\ledsetnormalparstuffX	319
Appendix A.6.2 Endnotes	319
Appendix A.7 Migration to eledmac 1.22.0	319
Appendix A.8 Migration to eledmac 1.23.0	319
Appendix A.9 Migration from eledmac to reledmac	320
Appendix A.9.1 Risk of ‘no room for a new’	320
Appendix A.9.2 Multiple indices with memoir	320
Appendix A.9.3 Deprecated commands and options	320
Appendix A.9.4 \renewcommandreplaced by command	321
Appendix A.9.5 Commands the names of which have been changed . . .	321
Appendix A.9.6 Endnotes	323
Appendix A.9.7 Z Series	323
Appendix A.9.8 Internal commands	323
Appendix A.10 Migration to reledmac 2.1.0	323
Appendix A.11 Migration to reledmac 2.1.3	323
Appendix A.12 Migration to reledmac 2.3.0	323
Appendix A.13 Migration to reledmac 2.4.0	324
Appendix A.14 Migration to reledmac 2.5.0	324
Appendix A.15 Migration to reledmac 2.7.0	324
Appendix A.16 Migration to reledmac 2.7.2	324
Appendix A.17 Migration to reledmac 2.8.0	324
References	325
Index	325
Change History	370

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 \TeX format for critical editions', *TUGboat* 11 (1990), pp. 623–643.

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

As of 1994, we were pleased to be able to say that `edmac` was being used for the real-life book production of several interesting editions, such as the Latin texts of Euclid's *Elements*,² an edition of the letters of Nicolaus Copernicus,³ Simon Bredon's *Arithmetica*,⁴ a Latin translation by Plato of Tivoli of an Arabic astrolabe text,⁵ a Latin translation of part II of the Arabic *Algebra* by Abū Kāmil Shujā' b. Aslam,⁶ the Latin *Rithmarchia* of Werinher von Tegernsee,⁷ a middle-Dutch romance epic on the Crusades,⁸ a seventeenth-century Hungarian politico-philosophical tract,⁹ an anonymous Latin compilation from Hun-

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).

gary entitled *Sermones Compilati in Studio Generali Quinqeeclesiensi in Regno Ungarie*,¹⁰ the collected letters and papers of Leibniz,¹¹ Theodosius's *Spherics*, the German *Algorismus* of Sacrobosco, the Sanskrit text of the *Kāśikāvṛtti* of Vāmana and Jayāditya,¹² and the English texts of Thomas Middleton's collected works.

1.2.2 ledmac

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

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

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

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

1.2.3 eledmac

Important changes were put in version 1.0, to make `ledmac` more easily extensible (see 7 p. 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. 317.

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. 320.

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.

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

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

2 How the package works

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

3 Compatibility warning

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

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

Please read 7.13.5 p. 41.

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

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

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

4 Options

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

4.1 Specific features

draft underlines lemmas in the main text.

eledmac-compat help to migrate from eledmac to reledmac (see Appendix A.9.5 p. 321).

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 17 p. 60).

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 (13.4 p. 54).

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.

4.2 Optimizing package performance

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

noeledsec disables tools for `\eledsection` and related commands (16.2 p. 59).

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

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

noledgroup reledmac allows use of a series of critical notes and a new series of normal notes inside minipage and ledgroup environments (see 10 p. 46). However, such features use up computer memory, at the expense of other processing needs. So if you do not need this feature, use noledgroup option. This should make reledmac faster.

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

5 Text lines and paragraphs numbering

5.1 Text lines numbering

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

```
\beginnumbering
Text
\endnumbering
```

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

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

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

5.2 Paragraphs

5.2.1 Basics

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

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

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

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


```

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

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

The lines of this paragraph are
not numbered.

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

```

5.2.2 Automatically producing \pstart ... \pend

`\autopar` You can use `\autopar` to avoid the nuisance of this paragraph markup and still have every paragraph automatically numbered. The scope of the `\autopar` command needs to be limited by keeping it within a group, as follows:

```

\begingroup
\beginnumbering
\autopar

A paragraph of numbered text.

Another paragraph of numbered
text.

\endnumbering
\endgroup

```

`\autopar` fails, however, on paragraphs that start with a `{` or with any other command that starts a new group before it generates any text. Such paragraphs need to be started explicitly, before the new group is opened, using `\indent`, `\noindent`, or `\leavevmode`, or using `\pstart` itself.¹³

5.2.3 Content before specific \pstart and after specific \pend

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

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

brackets, or to add brackets after a `\pend`, just add a `\relax` between `\pstart ... \pend` and the brackets.

This feature is also useful when typesetting verses (see 9 p. 43) or `reledpar` (see 19.1.8 p. 64).

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

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

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

`\AtEveryPend`

5.2.5 Numbering paragraphs (`\pstart`)

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

`\numberpstarttrue`

`\numberpstartfalse`

`\thepstart`

You can redefine the command `\thepstart` to change style. You can change the value of the `pstart` number by using *after* `\beginnumbering`:

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

On each `\beginnumbering` the numbering restarts.

`\sidepstartnumtrue`

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

`\labelpstarttrue`

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

5.2.6 Languages written in Right to Left

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

5.2.7 Memory limits

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

`\pausenumbering`

`\resumenumbering`

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

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

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

```

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

\resumenummering
\pstart
Another paragraph.
\pend
\endnumbering

```

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

```
\newcommand{\memorybreak}{\pausenumbering\resumenummering}
```

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

5.3 Lineation commands

5.3.1 Disabling lineation

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

5.3.2 Setting lineation start and step

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

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

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

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

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

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

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

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

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

5.3.3 Setting lineation reset

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

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

5.3.4 Setting line number margin

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

`\linenummargin{left}`

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

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

5.3.5 Other settings

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

5.4 Changing the line numbers

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

5.4.1 Sublineation

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

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

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

You can change the separator between line number and subline number or using `\Xsublinesep` without any option argument (7.2.8 p. 33 or using `\Xsublinesepside`. But in the second case, it will change the separator only for line number in side, not for the footnotes.

5.4.2 Locking lineation

`\startlock` The `\startlock` command, used in running text, locks the line number at its current value, until you insert `\endlock`. It can tell for itself whether you are in a patch of line or sub-line numbering. One use for line-number locking is in printing poetry: there the line numbers should be those of verse lines rather than of printed lines, even when a verse line requires several printed lines. But in this case you may use the `\stanza` mechanism, see 9 p. 43.

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

5.4.3 Setting and changing line number

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

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

5.4.4 Line number style

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

`\sublinenumberstyle`

Alph Uppercase letters (A ... Z).

alph Lowercase letters (a ... z).

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

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

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

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

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

5.4.5 Skipping and hiding number

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

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

5.4.6 Execute code at each line

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

6 Apparatus commands

6.1 Terminology

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

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

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

So :

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

6.2 Critical notes

6.2.1 The lemma

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

```
\edtext{⟨lemma⟩}{⟨commands⟩}
```

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

For example:

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

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

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

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

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

6.2.2 Footnotes

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

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

If you need more series of critical notes, please look at 6.5.1 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. 7.2.4 p. 32).

- `nonum` disables line numbering for this note. A horizontal blank space is added instead. You can use `\Xinplaceoflemmaseparator` to set it (7.5.1 p. 36).
- `nosep` to disable the lemma separator for this note.
- `linangesep=<c>` to change to `<c>` the separator between start line and end line for this particular note.

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

6.2.3 Endnotes

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

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

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

`\Dendnote`

`\Eendnote`

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

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

- `fulllines` to disable `\Xendtwolines` and `\Xendmoreethantwolines` features for this particular note (cf. 7.2.4 p. 32).
- `nonum` to disable line number for this particular note.
- `nosep` to disable the lemma separator for this particular note. A horizontal blank space is added instead. You can use `\Xendinplaceoflemmaseparator` to set it (7.5.2 p. 36).
- `linangesep=<c>` to change to `<c>` the separator between start line and end line for this particular note.

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

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

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



```
\doendnotesbysection{B}
```

Note that by default inside endnotes no separator is used between the lemma and the content. However you can use the `\Xendlemmaseparator` macro to define one (7.5.2 p. 36).

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

6.2.4 Paragraph in critical apparatus

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

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

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

6.2.5 Change lemma and line number

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

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

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

1 Smith] Jones C, D.

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

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

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

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

This command does not change the marginal line numbers in any way; it just changes the numbers passed to the notes. Its use comes in situations that `\edtext` has trouble dealing with for whatever reason. If you need notes for overlapping passages that aren't nested, for instance, you can use `\lemma` and `\linenum` to generate such notes despite

the limitations of `\edtext`. If the $\langle lemma \rangle$ argument to `\edtext` is extremely long, you may run out of memory; here again you can specify a note with an abbreviated lemma using `\lemma` and `\linenum`. The numbers used in `\linenum` need not be entered manually; you can use the ‘x-’ symbolic cross-referencing commands below (11 p. 47) to compute them automatically.

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

6.2.6 Changing the names of commands for critical apparatus

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

We recommend that you create a series of such aliases and use them instead of the names chosen here; all you have to do is put commands of this form at the start of your file:¹⁴

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

6.3 Disambiguation of identical words in the apparatus

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

6.3.1 Basic use

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

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

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

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

6.3.2 Notes about input encoding with UTF-8 processor

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

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

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

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

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

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

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

6.3.3 Use with `\lemma` command

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

For example in the following example:

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

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

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

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

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

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

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

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

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

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

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

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

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

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

6.3.4 Customizing

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

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

6.4 Familiar notes

6.4.1 Basic use

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

6.4.2 Customizing mark

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

6.4.3 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.

6.5 Changing series

6.5.1 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}`.

6.5.2 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}
```

6.5.3 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.

6.6 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}
```

7 Critical apparatus appearance

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

When a length, noted `<l>`, is used, it can be stretchable: `a plus b minus c`. The final length `m` is calculated by \TeX to have: $a - c \leq m \leq a + b$. If you use some relative unit¹⁶, it will be relative to font size of the footnote, except for commands concerning the place kept by the notes — including blank space.

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

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

There is also name convention:

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

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

7.1 Notes arrangement in a series

`\Xarrangement`
`\arrangementX`

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

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

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

- `paragraph` formats all the footnotes of a series as a single paragraph. If you use this arrangement, you are strongly encouraged to read 19.1.6 p. 62.
- `twocol` formats them as separate paragraphs, but in two columns;
- `threecol`, in three columns.
- `normal`, restore normal arrangement.

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

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

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

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

7.2 Control line number printing

7.2.1 Print line number only at first time

`\Xnumberonlyfirstinline`

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

`\numberonlyfirstintwolines`

Suppose you have a lemma on line 2 and a lemma between line 2 and line 3. With `\Xnumberonlyfirstinline`, the second lemma is considered to be on the same

line as the first lemma. But if you use both `\Xnumberonlyfirstinline[⟨s⟩]` and `\Xnumberonlyfirstintwelines[⟨s⟩]`, a distinction is made.

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

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

7.2.2 Arbitrary text before line number

`\Xbeforenumber` `\Xbeforenumber[⟨s⟩]{⟨txt⟩}` allow to insert `⟨txt⟩` before the line number, only when the line number is printed, so taking into account `\Xnumberonlyfirstinline` and similar.

7.2.3 Separator for line range

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

7.2.4 Abbreviate line range

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

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

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

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

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

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

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

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

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

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

7.2.5 Disable line number

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

7.2.6 Printing pstart number

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

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

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

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

7.2.7 Printing stanza number

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

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

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

7.2.8 Separator between line and subline numbers

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

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

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

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

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

7.2.9 Space around number

<code>\Xbeforenumber</code>	With <code>\Xbeforenumber[⟨s⟩]{⟨l⟩}</code> , 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.
<code>\Xafternumber</code>	With <code>\Xafternumber[⟨s⟩]{⟨l⟩}</code> 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.
<code>\Xendbeforenumber</code> <code>Xendafternumber</code>	<code>\Xendbeforenumber</code> and <code>\Xendafternumber</code> are the equivalents of <code>\Xbeforenumber</code> and <code>\afternumber</code> for endnotes.
<code>\Xnonbreakableafternumber</code>	By default, the space defined by <code>\Xafternumber</code> is breakable. With <code>\Xnonbreakableafternumber[⟨s⟩]{⟨l⟩}</code> it becomes nonbreakable.

7.2.10 Space around line symbol

<code>\Xbeforemsynl</code>	With <code>\Xbeforemsynl[⟨s⟩]{⟨l⟩}</code> you can add some space before the line symbol in a footnote. The default value is value set by <code>\Xbeforenumber</code> .
<code>\Xaftermsynl</code>	With <code>\Xaftermsynl[⟨s⟩]{⟨l⟩}</code> you can add some space after the line symbol in a footnote. The default value is value set by <code>\Xafternumber</code> .
<code>\Xendbeforemsynl</code> <code>\Xendaftermsynl</code>	<code>\Xendbeforemsynl</code> and <code>\Xendaftermsynl</code> are the equivalents of <code>\Xbeforemsynl</code> and <code>\Xaftermsynl</code> for the endnotes.

7.2.11 Space in place of number

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

7.2.12 Boxing line number and line symbol

<code>\Xboxl</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>\Xboxl[⟨s⟩]{⟨l⟩}</code> to do that. To subsequently disable this feature, use <code>\Xboxl</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} \Xboxl{1em}</pre>
<code>\Xboxmsynl</code> <code>\Xendboxmsynl</code>	<code>\Xboxmsynl[⟨s⟩]{⟨l⟩}</code> is the same as <code>\Xboxl</code> but for the line number symbol. <code>\Xendboxmsynl[⟨s⟩]{⟨l⟩}</code> is the same as <code>\Xboxmsynl</code> but for endnotes.
<code>\Xboxlalign</code>	If you put line number in box, it will be aligned left inside the box. However, you can change it using <code>\Xboxlalign[⟨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>\Xendboxlinenumalign</code> <code>\Xendboxstartlinenumalign</code> <code>\Xendboxendlinenumalign</code>	<code>\Xendboxlinenum[⟨s⟩]{⟨l⟩}</code> , <code>\Xendboxlinenumalign[⟨s⟩]{⟨text⟩}</code> , <code>\Xendboxstartlinenum[⟨s⟩]{⟨l⟩}</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.
---	---

7.3 For endnotes

<code>\Xendbeforepagenumber</code> <code>\Xendafterpagenumber</code> <code>\Xendlineprefixsingle</code> <code>\Xendlineprefixmore</code>	<code>\Xendbeforepagenumber[⟨s⟩]{⟨text⟩}</code> defines the text before the page number in endnotes. Default value is <code>p.</code> (“p” followed by a dot). <code>\Xendafterpagenumber[⟨s⟩]{⟨text⟩}</code> defines the text after the page number in endnotes. Default value is <code>)</code> (open parenthesis followed by a single space). <code>\Xendlineprefixsingle[⟨s⟩]{⟨text⟩}</code> defines the text before the line number in endnotes, when there is only one line. Default value is empty. <code>\Xendlineprefixmore[⟨s⟩]{⟨text⟩}</code> 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 <code>\Xendlineprefixsingle</code> .
---	--

7.4 Arbitrary code around line number

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

7.5 Separator between the lemma and the note

7.5.1 For footnotes

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

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

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

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

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

7.5.2 For endnotes

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

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

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

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

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

7.6 Font style

7.6.1 For line number

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

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

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

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

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

7.6.2 For the lemma

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

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

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

`\Xendlemmafont`

```
\Xlemmafont{\bfseries}
```

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

7.6.3 For all notes

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

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

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

7.7 Wrapping notes

7.7.1 Wrapping lemmas

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

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

7.7.2 Wrapping contents

`\Xwrapcontent` `\Xwrapcontent[⟨s⟩]{⟨cmd⟩}` is used to wrap the footnote contents — excluding the lemma — in a \TeX command.

For example, if the language of your note is not the same as the language of the lemma, use `\Xwrapcontent{\foreignlanguage{⟨language⟩}}` (with `babel`) or `\Xwrapcontent{\text{⟨language⟩}}` (for `babel`).

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

`\wrapcontentX` `\wrapcontentX[⟨s⟩]{⟨cmd⟩}` is the same for critical footnotes.

7.8 Indent of notes content

`\Xparindent` By default, `reledmac` does not add indentation before the paragraphs inside critical footnotes. Use `\Xparindent[⟨s⟩]` to enable indentation.

`\parindentX` By default, `reledmac` does not add indentation before the paragraphs inside familiar footnotes. Use `\parindentX[⟨s⟩]` to enable indentation.

`\Xhangindent` For critical notes NOT paragraphed you can define an indent with `\Xhangindent[⟨s⟩]{⟨l⟩}`, 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.

`\hangindentX` For familiar notes NOT paragraphed you can define an indentation with `\hangindentX[⟨s⟩]{⟨l⟩}`, 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.

`\Xendhangindent` For critical endnotes NOT paragraphed you can define an indentation with `\Xendhangindent[⟨s⟩]{⟨l⟩}`, which will be applied in the second line of notes. It can help to make a distinction between a new note and a break in a note.

7.9 Arbitrary code at the beginning of notes

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

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

`\Xbhooknote` `\Xbhooknote[⟨s⟩]{⟨code⟩}` is to be used at the beginning of the critical footnotes.

`\bhooknoteX` `\bhooknoteX[⟨s⟩]{⟨code⟩}` is to be used at the beginning of the familiar footnotes.

`\Xendbhooknote` `\Xendbhooknote[⟨s⟩]{⟨code⟩}` is to be used at the beginning of the endnotes.

7.10 Arbitrary code before inserting note

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

They allow one to add any arbitrary code just before the footnotes are added in the list of footnotes. The main use is to insert text direction code. For example, if you edit right-to-left text with `bidl`, but want your critical footnote be left-to-right, use `\Xbeforeinserting\LTR`. You should also use `\Xwraplemma` to ensure your lemmas are right-to-left in a left-to-right paragraph (7.7.1 p. 37)).

Note that the changes are local to the footnote.

7.11 Options for footnotes in columns

7.11.1 Alignment

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

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

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

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

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

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

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

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

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

7.11.2 Size of the columns

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

`\Xhsizetwocol` `\Xhsizetwocol[⟨s⟩]{⟨l⟩}` is used to change width of a column when critical notes are displaying in two columns. Default value is `.45 \hspace`.

`\Xhsizethreecol` `\Xhsizethreecol[⟨s⟩]{⟨l⟩}` is used to change width of a column when critical notes are displaying in three columns. Default value is `.3 \hspace`.

`\hsizetwocolX` `\hsizetwocolX[⟨s⟩]{⟨l⟩}` is used to change width of a column when familiar notes are displaying in two columns. Default value is `.45 \hspace`.

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

7.12 Options for paragraphed footnotes

7.12.1 Mark separation of notes

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

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

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

7.12.2 Ragged text

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

7.13 Options for block of notes

7.13.1 Text before notes

- `\Xtxtbeforenotes` You can add text before critical notes with `\Xtxtbeforenotes[⟨s⟩]{⟨text⟩}`.

7.13.2 Code before notes

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

7.13.3 Spacing

- `\Xbeforenotes` You can change the vertical space before the rule of the critical notes with `\Xbeforenotes[⟨s⟩]{⟨l⟩}`. The default value is 1.2em plus .6em minus .6em.
- Be careful, the standard L^AT_EX footnote rule used by reledmac decreases by 3pt. This 3pt decrease is not changed by this command.**
- `\beforenotesX` You can change the vertical space printed before the rule of the familiar notes with `\beforenotesX[⟨s⟩]{⟨l⟩}`. The default value is 1.2em plus .6em minus .6em.
- Be careful, the standard L^AT_EX footnote rule, which is used by reledmac, decreases 3pt. These 3pt are not changed by this command.**
- `\preXnotes` You can set the space before the first series of critical notes printed on each page and set a different amount of space for each subsequent series on the page. You can do it with `\preXnotes{⟨l⟩}`. The default value is 0pt. You can disable this feature by setting the length to 0pt.
- `\prenotesX` You can set the space before the first printed (in a page) series of familiar notes to be different from the space before other series. The default value is 0pt. You can do this with `\prenotesX{⟨l⟩}`. You can disable this feature by setting the length to 0pt.

7.13.4 Rule

- `\Xafterrule` You can change the vertical space printed after the rule of the critical notes with `\Xafterrule[⟨s⟩]{⟨l⟩}`. The default value is 0pt.
- Be careful, the standard L^AT_EX footnote rule, which is used by reledmac, adds 2.6pt. These 2.6pt are not changed by this command.**
- `\afterruleX` You can change the vertical space printed after the rule of the familiar notes with `\afterruleX[⟨s⟩]{⟨l⟩}`. The default value is 0pt.
- Be careful, the standard L^AT_EX footnote rule, which is used by reledmac, adds 2.6pt. These 2.6pt are not changed by this command.**

7.13.5 Maximum height

`\Xmaxhnotes` By default, one series of critical notes can take up to 80% of `\vsize`, before being broken to the next page. If you want to change the size use `\Xmaxhnotes[⟨s⟩]{⟨l⟩}`. Be careful : the length can't be flexible, and is relative to the the current font. For example, if you want the note to take, at most, 33% of the text height, do `\Xmaxhnotes{.33\textheight}`.

`\maxhnotesX` `\maxhnotesX[⟨s⟩]{⟨l⟩}` is the same as previous, but for familiar footnotes. Note that in many cases, you should call these commands in the begin of the document, because the `\vsize` in the preamble is not the same as `\vsize` after the preamble. That why we recommend to you to add in your preamble

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

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

7.13.6 Width

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

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

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

Note that changes the width of the block of notes. If you want to change the width of each column when typesetting notes in columns, use `\Xhsizetwocol`, `\Xhsizethreecol`, `\hsizetwocolX`, `\hsizethreecolX`, see 7.11.2 p. 39.

7.14 Footnotes and the reledpar columns

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

`\noteswidthliketwocolumnsX`

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

7.15 Endnotes in one paragraph

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

8 Fonts

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

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

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

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

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

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

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

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

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

9 Verse

9.1 Basic

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

9.2 Define stanza indents

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

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

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

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

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

If you want the hanging verse to be flush right, you can use `\sethanginsymbol:` see p. 9.6 p. 45.

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

9.3 Repeating stanza indents

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

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

is like

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

Be careful: the feature is changed in eledmac 1.5.1. See Appendix A.3 p. 318.

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

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

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

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

9.4 Manual stanza indent

`\stanzaindent`
`\stanzaindent*`

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

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

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

9.5 Stanza breaking

`\setstanzapenalties`

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

The command

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

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

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

9.6 Hanging symbol

It is possible to insert a symbol in each line of hanging verse, as in French typography; for example, the opening bracket ‘[’. To insert it in `reledmac`, use macro

`\sethangingsymbol`

`\sethangingsymbol{⟨h⟩}` with this code. In the example of French typography, do

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

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

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

9.7 Long verse and page break

If you want to prevent page breaks inside long verses, use the option `nopbinverse` when loading package, or use `\lednopbinversetrue`. Read 18.2 p. 60 for further details.

9.8 Content before/after verses

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

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

Use `\AtEveryStanza{⟨arg⟩}` to automatically add content at the begining of stanza.

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

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

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

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

9.9 Numbering stanza

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

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

You can use this feature in combination with `\Xstanza` (7.2.7 p. 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 \TeX .
`\stanzanumwrapper` You can redefine `\stanzanumwrapper` in order to modify the way the stanza number is inserted in the flow of text. Default value is:

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

9.10 Various tools

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

9.11 Notes on empty lines

Since v2.3.0 of `reledmac`, empty lines when typesetting verses no longer produce new paragraphs, and consequently, do not insert vertical spaces. Use optional argument of `\stanza` or `\newverse` to insert vertical space (9.8 p. 45).

10 Grouping

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

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

You can also put familiar footnotes (see section 6.4) 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 `ledroup` except that you must specify a width for the environment, as with a `minipage`.
`\begin{ledgroupsize}[⟨pos⟩]{⟨width⟩}`.

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

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

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

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

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

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

11 Cross referencing

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

11.1 Basic use

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

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

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

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

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

11.2 Cross-referencing to a critical note

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

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

If you want to refer to starting and ending lines, you should use `\appref` and related tools (11.6.2 p. 49).

11.3 Cross-referencing which return a number in any case

`\xpageref` Where #1 stands for the reference.

`\xlineref`

`\xsublineref`

`\xpstartref`

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

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

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

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

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

`\xxref` The macros `\xxref` and `\edmakelabel` let you manipulate numbers and labels in ways which you may find helpful in tricky situations.

The `\xxref{<lab1>}{<lab2>}` command generates a reference to a sequence of lines, for use in the second argument of `\edtext`. It takes two arguments, both of which are labels: e.g., `\xxref{mouse}{elephant}`. It calls `\linenum` (q.v., 6.2.5 p. 25 above) and sets the beginning page, line and subline numbers to those of the place where `\edlabel{mouse}` was placed, and the ending numbers to those where `\edlabel{elephant}` occurs.

11.4 Not automatic cross-referencing

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

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

11.5 Normal \LaTeX cross-referencing

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

11.6 References to start and end lines

11.6.1 Reference to main text lines

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

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

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

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

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

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

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

11.6.2 References to lines that are commented on in the apparatus

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

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

If you use `\linenum` (6.2.5 p. 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
`\apprefwithpage` lines. The first one will print the lines as they are printed in the critical footnotes, while the second will print the lines as they are printed in endnotes.

11.6.3 Settings

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

For example, you may use:

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

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

```
\setSErefprefixsingle
  setSErefprefixmore
\setSErefonlypageprefixsingle
  \setSErefonlypageprefixmore
```

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

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

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

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

Also note that `\setSErefonlypageprefixsingle` is only a shortcut for `\Xendbeforepagenumber` (see 11.6.3 p. 50). So if you use `\Xendbeforepagenumber` without any optional argument, it will override this setting.

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

These commands are the following:

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

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

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

- `\Xendafterpagenumber` (not for `\SErefonlypage`)
- `\Xendbeforepagenumber`
- `\Xendlineflag` (for `reledpar`), enabled by default.
- `\Xendlineprefixmore`

- `\Xendlineprefixsingle`
- `\Xendlinerrangeseparator`
- `\Xendmoreethantwolines`
- `\Xendsublinesep`
- `\Xendtwolines`
- `\Xendtwolinesbutnotmore`
- `\Xendtwolinesonlyinsamepage`

For one specific command When calling `\appref` and `\Seref`, you can use as a first optional argument, in brackets (`[]`), any optional argument which can be used for critical footnotes (6.2.2 p. 23).

When calling `\apprefwithpage`, `\Serefwithpage` or `\Serefonlypage` you can use as a first optional argument, in brackets (`[]`), any optional argument which can be used for critical endnotes (6.2.3 p. 24).

11.7 Compatibility with *xr* package

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

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

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

12 Side notes

12.1 Basics

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

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

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

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

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

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

12.2 Setting

12.2.1 Width

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

12.2.2 Vertical position

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

12.2.3 Distance to the main text

`\ledlsnotesep` The texts are put a distance `\ledlsnotesep` (or `\ledrsnotesep`) into the left (or right)
`\ledrsnotesep` margin. These lengths are initially set to the value of `\linenumsep`.
`\ledlsnotefontsetup` These macros specify how the sidenote texts are to be typeset. The initial definitions
`\ledrsnotefontsetup` are:

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

These can of course be changed to suit.

12.2.4 Separator between notes

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

13 Indexing

13.1 Basics

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

Note that the file `.idx` will contain the right reference only after the third run, because of the internal indexing mechanism of `reledmac`. That means you must first run

(Xe/Lua) \LaTeX three times, then run `makeindex`, and then finally run (Xe/Lua) \LaTeX again, in order to get an index with the right page numbers.

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

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

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

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

13.2 Referring to critical notes

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

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

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

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

13.3 Separator between page and line numbers

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

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

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.

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

13.4 Using xindy

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

`pagenumber-linenummer`

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

This file also provides, with an explanation, the settings that are needed to put `reledmac` lines numbers in parenthesis, in order to make a better distinction between line numbers and page ranges.

In any case, you must load `reledmac` with the `xindy` option, in order to generate a `.xdy` file which is specific to your document. This file is needed by the `.xdy` example file which is in the “examples” folder. Its default name is `reledmac-markup-attr.xdy`, but you can change it by using your own as an argument of the `xindy+hyperref` option.

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

1. Use `xindy+hyperref` option when loading the `reledmac` package. When you run (Xe/Lua)TeX with this option, a `.xdy` configuration file will be generated with all the settings needed to allow internal hyperlinking in each index entry which is created by `\edindex`.
2. Use `hyperindex=false` option when loading `hyperref`.
3. Uncomment — by removing the semicolons at the beginning of the relevant lines — some lines in the `<code>.xdy</code>` file provided in the “examples” folder in order to restore internal links in the index to be used by the standard `index` command.²¹

13.5 Advanced setting

`\edindexlab` The `\edindex` process uses a `\label` and `\ref` mechanism to get the correct line number. It automatically generates labels of the form `\label{\edindexlab N}`, where `N` is a number, and the default definition of `\edindexlab` is:

`\newcommand*{\edindexlab}{\&\&}`

in the hopes that this will not be used by any other labels (`\edindex`’s labels are like `\label{\&\&27}`). You can change `\edindexlab` to something else if you need to.

14 Glossary

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

²⁰Or, for people who read French, read <http://geekographie.maieul.net/174>.

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

14.1 Preamble setting

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

- Or set `glossaries`:
`\glsSetCompositor{-}`
- Or set `reledmac`:
`\renewcommand{\pagelinesep}{-}`
 In this case, the above will have consequences for your use of `\edindex` and you should set your `.ist` file (13.3 p. 53).

14.2 Commands

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

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

15 Tabular material

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

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

```

edarrayl
edarrayc
edarrayr
edtabularl
edtabularc
edtabularr

```

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

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

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

For example:

```

\beginnumbering
\pstart
\begin{edtabularl}

```

```

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

```

produces the following parallel pair of verses.

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

`\edtabcolsep` The distance between the columns is controlled by the length `\edtabcolsep`.
`\spreadmath` `\spreadmath{\langle math \rangle}` typesets `\langle math \rangle` but the `\langle math \rangle` has no effect on the
`\spreadtext` calculation of column widths. `\spreadtext{\langle text \rangle}` is the analagous command for use
in edtabular environments.

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

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

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

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

```

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

```


1	2	3	4	5
Q		fd	h	qwertziohg
v	wptz	x	y	vb
g	nnn	$\overbrace{\hspace{10em}}$		
k	$\overbrace{\hspace{10em}}$		pq	dgh
1	2	3	co	ghweropjklmnbvcxys

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

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

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

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

1	2	3	4
a	\lrcorner	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, `\edaftertab` 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}
```

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

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

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

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

16 Sectioning commands

16.1 Sectioning commands without line numbers or critical notes

The standard sectioning commands (`\chapter`, `\section` etc.) can be used inside numbered text. In this case, you must call them as an optional argument of `\pstart` (5.2.3 p. 17):

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

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

16.2 Sectioning commands with line numbering and critical notes

You have to use the following commands:

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

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

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

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

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

16.3 Optimization

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

17 Quotation environments

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

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

18 Page breaks

18.1 Control page breaking

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

`\ledpb`
`\lednopb`

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

These commands have effect only at the second run.

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. $n + 1$.

`\ledpbsetting`

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

18.2 Prevent page break in a long verses

`\lednopbinversettrue`

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 `\lednopbinversettrue` in the beginning of your file (better: in your preamble).

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

19 Miscellaneous

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

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

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

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

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

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

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

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

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

19.1 Known and suspected limitations

19.1.1 Non-standard geometry

If you use classes other than `article` or `book`, or if you use the `geometry` package, you should use `maxnotesX` and/or `\Xmaxnotes` as explained in 7.13.5 p. 41 in order to prevent footnotes from overlapping the bottom margin.

19.1.2 floatrow package compatibility

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

19.1.3 ‘No room for a new’

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

In order to prevent such problems, the first thing is to use the options to optimize `reledmac`. For example, if you need only two series of notes, use the `series={A,B}` option. Read 16.3 p. 59 in order to know which are the available options.

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

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

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

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

19.1.4 Marginal notes

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

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

19.1.5 Paragraph shape

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

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

19.1.6 Paragraphed footnotes

The restriction on explicit line-breaking in paragraphed footnotes, mentioned on 7.1 p. 31, and described in more detail on XII.6.3 p. 156, 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` (7.13.5 p. 41), in order to define the maximum height relative to `\textheight` and not to `\vsize`, because the `\vsize` value is not the same inside and outside of the preamble.

19.1.7 Use with other packages

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

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

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

If you actually try this²² you will find \TeX whinging 'Missing { inserted', and then things start to fall apart. The trick in this case is to specify either:

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

or

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

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

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

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

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

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

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

²²Reported by Dirk-Jan Dekker in the CTT thread 'Incompatibility of "color" package' on 2003/08/28.

19.1.8 Parallel typesetting

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

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

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}[2016/05/08 v2.10.0 typeset critical editions]%
4 %

```

II.3 Package options

<pre> \ifledfinal \ifnocritical@ \if@noeled@sec \ifnoend@ \ifnofamiliar@ \ifnoledgroup@ \ifparapparatus@ \ifnoquotation@ \iflednopbinverse \ifparledgroup \ifwidthliketwocolumns \ifxindy@ \ifxindyhyperref@ \ifeledmaccompat@ </pre>	<pre> 5 \RequirePackage{xkeyval} 6 % 7 \newif\ifparledgroup 8 % 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 \newif\ifledfinal 36 \DeclareOptionX{final}{\ledfinaltrue} 37 \DeclareOptionX{draft}{\ledfinalfalse} 38 \ExecuteOptionsX{final} </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> <p>The parledgroup option is for reledpar. However, it has consequence on reledmac internal command. So we need to define the boolean now.</p> <p>And now, the options of reledmac.</p>
---	---	---

```

39 \newif\ifparapparatus@
40 \DeclareOptionX{parapparatus}{\parapparatus@true}
41
42 \newif\iflednopbinverse
43 \DeclareOptionX{nopbinverse}{\lednopbinversetrue}
44
45 \newif\ifwidthliketwocolumns%
46 \DeclareOptionX{widthliketwocolumns}{\widthliketwocolumnstrue}%
47
48 \newif\ifcontinuousnumberingwithcolumns
49 \DeclareOptionX{continuousnumberingwithcolumns}{\
50 continuousnumberingwithcolumnstrue}%
51
52 \newif\ifxindy@
53 \DeclareOptionX{xindy}[eledmac-markup-attr.xdy]{%
54   \AtBeginDocument{\immediate\openout\eledmac@xindy@out=#1}%
55   \newwrite\eledmac@xindy@out%
56   \xindy@true%
57   \gdef\eledmacmarkuplocdepth{:depth 1}%
58   \AtEndDocument{\immediate\closeout\eledmac@xindy@out}%
59 }%
60
61 \newif\ifxindyhyperref@
62 \DeclareOptionX{xindy+hyperref}{%
63   \xindyhyperref@true%
64 }%
65
66 \newif\ifeledmaccompat%
67 \DeclareOptionX{eledmac-compat}{%
68   \eledmaccompat@true%
69 }%
70 %
71 \ProcessOptionsX*\relax
72
73 %

```

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

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 Lua_{TeX} or Xe_{TeX} is running, and `ragged2e` to manage ragged justification for paragraphed notes.

```

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

```

II.5 Compatibility with Lua_T_EX

Here, we enable some primitives for Lua_T_EX.

```

87 \ifx\directlua\undefined\else%
88   \directlua{tex.enableprimitives("",{"texdir","pardir","bodydir"})}
89 \fi
90 %

```

II.6 Boolean flags

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

```

91 \newif\ifl@dmemoir
92 \@ifclassloaded{memoir}{\l@dmemoirtrue}{\l@dmemoirfalse}
93
94 %

```

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

```

95 \newif\if@ledgroup%
96 %

```

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

```

97 \newif\ifl@imakeidx
98 \@ifpackageloaded{imakeidx}{\l@imakeidxtrue}{}%False is the default value
99 %

```

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

```

100 \newif\ifl@indextools%
101 \@ifpackageloaded{indextools}{%
102   \l@indextoolstrue%
103   \l@imakeidxtrue%

```

```

104 \let\imki@wrindexentry\indtl@wrindexentry%
105 }{}%
106 %

```

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.

```

107 \ifdef{\if@RTL}{-}{\newif\if@RTL}
108 %

```

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.

```

109 \newcommand{\reledmac@warning}[1]{\PackageWarning{reledmac}{#1}}
110 %

```

`\reledmac@error` Write an error message.

```

111 \newcommand{\reledmac@error}[2]{\PackageError{reledmac}{#1}{#2}}
112 %

```

```

\led@err@NumberingStarted13 \newcommand*{\led@err@NumberingStarted}{%
\led@err@NumberingNotStarted14 \reledmac@error{Numbering has already been started}{\@ehc}}
NumberingShouldHaveStarted15 \newcommand*{\led@err@NumberingNotStarted}{%
116 \reledmac@error{Numbering was not started}{\@ehc}}
117 \newcommand*{\led@err@NumberingShouldHaveStarted}{%
118 \reledmac@error{Numbering should already have been started}{\@ehc}}
119 %

```

```

\led@err@edtextoutsidestart20 \newcommand*{\led@err@edtextoutsidestart}{%
121 \reledmac@error{\string\edtext\space outside numbered paragraph (\pstart\
ldots\pend)}{\@ehc}}%
122 %

```

```

\led@mess@NotesChanged23 \newcommand*{\led@mess@NotesChanged}{%
124 \typeout{reledmac reminder: }%
125 \typeout{ The number of the footnotes in this section
126 has changed since the last run.}%
127 \typeout{ You will need to run LaTeX two more times

```

```

128         before the footnote placement}%
129     \typeout{ and line numbering in this section are
130         correct.}}
131 %

```

```

\led@mess@SectionContinued32 \newcommand*\led@mess@SectionContinued}[1]{%
133     \message{Section #1 (continuing the previous section)}}
134 %

```

```

\led@err@LineationInNumbered35 \newcommand*\led@err@LineationInNumbered{%
136     \reledmac@error{You can't use \string\lineation\space within
137         a numbered section}{\@ehc}}
138 %

```

```

\led@warn@BadLineation39 \newcommand*\led@warn@BadLineation{%
\led@warn@BadLinenummargin40 \reledmac@warning{Bad \string\lineation\space argument}}
\led@warn@BadLockdisp41 \newcommand*\led@warn@BadLinenummargin{%
\led@warn@BadSublockdisp42 \reledmac@warning{Bad \string\linenummargin\space argument}}
143 \newcommand*\led@warn@BadLockdisp{%
144     \reledmac@warning{Bad \string\lockdisp\space argument}}
145 \newcommand*\led@warn@BadSublockdisp{%
146     \reledmac@warning{Bad \string\sublockdisp\space argument}}
147 %

```

```

\led@warn@NoLineFile48 \newcommand*\led@warn@NoLineFile}[1]{%
149     \reledmac@warning{Can't find line-list file #1}}
150 %

```

```

\led@warn@LineFileObsolete51 \newcommand*\led@warn@Obsolete}[1]{%
152     \reledmac@warning{Line-list file #1 was obsolete. We have not read it.
Please run LaTeX again.}}
153 %

```

```

\led@warn@BadAdvancelineSubline54 \newcommand*\led@warn@BadAdvancelineSubline{%
\led@warn@BadAdvancelineLine55 \reledmac@warning{\string\advanceline\space produced a sub-line
156     number less than zero.}}
157 \newcommand*\led@warn@BadAdvancelineLine{%
158     \reledmac@warning{\string\advanceline\space produced a line
159     number less than zero.}}
160 %

```

```

\led@warn@BadSetline61 \newcommand*{\led@warn@BadSetline}{%
\led@warn@BadSetlinenum62 \reledmac@warning{Bad \string\setline\space argument}}
163 \newcommand*{\led@warn@BadSetlinenum}{%
164 \reledmac@warning{Bad \string\setlinenum\space argument}}
165 %

\led@err@PstartNotNumbered66 \newcommand*{\led@err@PstartNotNumbered}{%
\led@err@PstartInPstart67 \reledmac@error{\string\pstart\space must be used within a
\led@err@PendNotNumbered68 numbered section %
\led@err@PendNoPstart69 (\string\...beginnumbering\string\endnumbering)}{\@ehc}}%
\led@err@AutoparNotNumbered70 \newcommand*{\led@err@PstartInPstart}{%
\led@err@NumberingWithoutPstart71 \reledmac@error{\string\pstart\space encountered while another
\string\pstart\space was in effect}{\@ehc}}
172 \newcommand*{\led@err@PendNotNumbered}{%
173 \reledmac@error{\string\pend\space must be used within a
174 numbered section}{\@ehc}}
175 \newcommand*{\led@err@PendNoPstart}{%
176 \reledmac@error{\string\pend\space must follow a \string\pstart}{\@ehc}}
177 \newcommand*{\led@err@AutoparNotNumbered}{%
178 \reledmac@error{\string\autopar\space must be used within a
179 numbered section}{\@ehc}}
180 \newcommand*{\led@err@NumberingWithoutPstart}{%
181 \reledmac@error{\string\beginnumbering...\string\endnumbering\space
182 without \string\pstart}{\@ehc}}%
183 %

\led@warn@BadAction84 \newcommand*{\led@warn@BadAction}{%
185 \reledmac@warning{Bad action code, value \next@action.}}
186 %

\led@warn@DuplicateLabel87 \newcommand*{\led@warn@DuplicateLabel}[1]{%
\led@warn@AppLabelOutEdtext88 \reledmac@warning{Duplicate definition of label `#1'\@gobble}%
\led@warn@RefUndefined89 \@latex@warning@no@line{Label `#1' multiply defined}%
\led@warn@RefUndefined90 }%
191 \newcommand*{\led@warn@AppLabelOutEdtext}[1]{%
192 \reledmac@warning{\string\applabel\space outside of \string\edtext\space
`#1' on page \the\pageno.}}%
193 \newcommand*{\led@warn@RefUndefined}[1]{%
194 \G@refundefinedtrue%
195 \reledmac@warning{Reference `#1' on page \the\pageno\space undefined.%
196 Using `000'.}}%
197 \@latex@warning{Reference `#1' undefined\on@line}%
198 }%
199 \newcommand*{\led@warn@pairRefUndefined}[1]{%
200 \G@refundefinedtrue%
201 \reledmac@warning{Reference `#1:start' and/or `#1:end' on page \the\
pageno\space undefined.}

```

```

202         Using `??'.}%
203     \@latex@warning{Reference `#1:start' and/or `#1:end' undefined\on@line}%
204 }
205 %

```

```

\led@warn@NoMarginpars\newcommand*\led@warn@NoMarginpars}{%
206
207     \reledmac@warning{You can't use \string\marginpar\space in numbered text
208     }}
209 %

```

```

\led@warn@BadSidenotemargin\newcommand*\led@warn@BadSidenotemargin}{%
210
211     \reledmac@warning{Bad \string\sidenotemmargin\space argument}}
212 %

```

```

\led@warn@NoIndexFile\newcommand*\led@warn@NoIndexFile}[1]{%
213
214     \reledmac@warning{Undefined index file #1}}
215 %

```

```

\led@warn@SeriesStillExist\newcommand*\led@warn@SeriesStillExist}[1]{%
216
217     \reledmac@warning{Series #1 is still existing !}%
218 }%
219 %

```

```

\led@err@ManySidenotes\newcommand*\led@err@ManySidenotes}{%
220
221     \ifledRcol%
222     \reledmac@warning{\itemcount@\space sidenotes on line \the\line@numR\
223     space p. \the\page@numR}%
224     \else%
225     \reledmac@warning{\itemcount@\space sidenotes on line \the\line@num\
226     space p. \the\page@num}%
227     \fi%
228 }%
229 \newcommand*\led@err@ManyLeftnotes}{%
230
231     \ifledRcol%
232     \reledmac@warning{\itemcount@\space leftnotes on line \the\line@numR\
233     space p. \the\page@numR}%
234     \else%
235     \reledmac@warning{\itemcount@\space leftnotes on line \the\line@num\
236     space p. \the\page@num}%
237     \fi%
238 }%
239 \newcommand*\led@err@ManyRightnotes}{%
240
241     \ifledRcol%
242     \reledmac@warning{\itemcount@\space rightnotes on line \the\line@numR\
243     space p. \the\page@numR}%
244     \else%
245     \reledmac@warning{\itemcount@\space rightnotes on line \the\line@num\
246     space p. \the\page@num}%
247     \fi%
248 }%

```



```

236 \else%
237 \reledmac@warning{\itemcount@\space rightnotes on line \the\line@num\
space p. \the\page@num}%
238 \fi%
239 }%
240 %

```

```

\led@err@TooManyColumns41 \newcommand*\led@err@TooManyColumns}{%
\led@err@UnequalColumns42 \reledmac@error{Too many columns}{\@ehc}}
\led@err@LowStartColumn43 \newcommand*\led@err@UnequalColumns}{%
\led@err@HighEndColumn44 \reledmac@error{Number of columns is not equal to the number
\led@err@ReverseColumns45 in the previous row (or \protect\ \space forgotten?)}{\
@ehc}}
246 \newcommand*\led@err@LowStartColumn}{%
247 \reledmac@error{Start column is too low}{\@ehc}}
248 \newcommand*\led@err@HighEndColumn}{%
249 \reledmac@error{End column is too high}{\@ehc}}
250 \newcommand*\led@err@ReverseColumns}{%
251 \reledmac@error{Start column is greater than end column}{\@ehc}}
252 %

```

```

err@EdtextWithoutFootnote53 \newcommand{\led@err@EdtextWithoutFootnote}{%
254 \reledmac@error{edtext without Xfootnote. Check syntax.}{\@ehc}%
255 }%
256 %

```

```

err@FootnoteWithoutEdtext57 \newcommand{\led@err@FootnoteWithoutEdtext}{%
258 \reledmac@error{Xfootnote without edtext. Check syntax.}{\@ehc}%
259 }%
260 %

```

```

rror@ImakeidxAfterEledmac61 \newcommand{\led@error@ImakeidxAfterEledmac}{%
262 \reledmac@error{Imakeidx must be loaded before reledmac.}{\@ehc}%
263 }%
264 %

```

```

or@IndextoolsAfterEledmac65 \newcommand{\led@error@IndextoolsAfterEledmac}{%
266 \reledmac@error{Indextools must be loaded before reledmac.}{\@ehc}%
267 }%
268 %

```

```

error@fail@patch@@makecol69 \newcommand{\led@error@fail@patch@@makecol}{%
270 \reledmac@error{Fail to patch \string\@makecol\space command.}{\@ehc}%
271 }%
272 %

```

```

\led@error@fail@patch@@reinserts 273 \newcommand{\led@error@fail@patch@@reinserts}{%
274   \reledmac@error{Fail to patch \string\@reinserts\space command.}\@ehc}%
275 }%
276 %

\led@error@fail@patch@@doclearpage 277 \newcommand{\led@error@fail@patch@@doclearpage}{%
278   \reledmac@error{Fail to patch \string\@doclearpage\space command.}\@ehc}%
279 }%
280 %

\led@error@fail@patch@@iiiminipage 281 \newcommand{\led@error@fail@patch@@iiiminipage}{%
282   \reledmac@error{Fail to patch \string\@iiiminipage\space command.}\@ehc}%
283 }%
284 %

\led@error@fail@patch@endminipage 285 \newcommand{\led@error@fail@patch@endminipage}{%
286   \reledmac@error{Fail to patch \string\endminipage\space command.}\@ehc}%
287 }%
288 %

\led@warning@hsizeX@deprecated 289 \newcommand{\led@warning@hsizeX@deprecated}{%
290   \reledmac@warning{\string\hsizeX\space command deprecated, use \string\
widthX\space instead.}%
291 }%
292 %

\led@warning@Xhsize@deprecated 293 \newcommand{\led@warning@Xhsize@deprecated}{%
294   \reledmac@warning{\string\Xhsize\space command deprecated, use \string\
Xwidth\space instead.}%
295 }%
296 %

```

II.8 Gobbling

Here, we define some commands which gobble their arguments.

```

\@gobblethree 297 \providecommand*\@gobblethree}[3]{}
\@gobblefour 298 \providecommand*\@gobblefour}[4]{}
\@gobblefive 299 \providecommand*\@gobblefive}[5]{}
300 %

```

II.9 Miscellaneous commands

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

```

301 \ifl@dfinal
302   \newcommand*{\showlemma}[1]{#1}
303 \else
304   \newcommand*{\showlemma}[1]{\underline{#1}}
305 \fi
306
307 %

```

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

Initialize it as `\empty`.

```

308 \let\linenumberlist=\empty
309
310 %

```

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

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

```

311 \newcount\@l@tempcnta \newcount\@l@tempcntb
312 %

```

II.10 Prepare reledpar

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

`\ifl@dpaging`

`\ifl@dprintingpages`

`\ifl@dprintingcolumns`

```

313 \newif\ifl@dpairing
314 \newif\ifl@dpaging%
315 \newif\ifl@dprintingpages%
316 \newif\ifl@dprintingcolumns%
317 \newif\ifpst@rtedL
318 \newcount\l@dnumpstartsL
319 %

```

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

```

320 \newif\ifledRcol
321 \newif\ifledRcol@
322 %

```

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

```
323 \newif\ifnumberingR
324 %
```

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

```
325 \newif\ifXnote@%
326 %
```

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.

```
327 \providebool{indtl@innote}%
328 \providebool{indtl@notenumber}%
329 %
```

III Sectioning commands

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

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

```
330 \newcount\section@num
331 \section@num=0
332 \let\extensionchars=\empty
333 %
```

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

`\numberingtrue`
`\numberingfalse`

```

334 \newif\ifnumbering
335 %

```

`\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.

```

336 \newcommand*{\beginnumbering}{%
337   \ifnumbering
338     \led@err@NumberingStarted
339   \endnumbering
340   \fi
341   \global\numberingtrue
342   \global\advance\section@num \@ne
343   \initnumbering@reg
344   \message{Section \the\section@num }%
345   \line@list@stuff{\jobname.\extensionchars\the\section@num}%
346   \l@dend@stuff
347   \setcounter{pstart}{1}
348   \ifl@dpairing
349     \global\l@dnumpstartsL \z@
350     \global\pst@rtedLfalse
351 %

```

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.

```

352 \else
353   \begingroup
354   \global\@afterindenttrue%In order to reestablish normal feature if the \
begingroup was not here
355   \initnumbering@quote

```

```

356 \ifwidthliketwocolumns%
357 \csuse{setwidthliketwocolumns@\columns@position}%
358 \csuse{setpositionliketwocolumns@\columns@position}%
359 \fi%
360 \fi
361 \gdef\eled@sections@{}%
362 \if@noeled@sec\else%
363 \makeatletter\input{IfFileExists{\jobname.eledsec\the\section@num}{-}{-}\
makeatother%
364 \immediate\openout\eled@sectioning@out=\jobname.eledsec\the\section@num
\relax%
365 \fi%
366 }
367 \newcommand*\initnumbering@reg{%
368 \global\pst@rtedLfalse
369 \global\l@dnumstartsL \z@
370 \global\absline@num \z@
371 \gdef\normal@page@break{}
372 \gdef\l@prev@pb{}
373 \gdef\l@prev@nopb{}
374 \global\line@num \z@
375 \global\subline@num \z@
376 \global\@lock \z@
377 \global\sub@lock \z@
378 \global\sublines@false
379 \global\let\next@page@num=\relax
380 \global\let\sub@change=\relax
381 \resetprevline@
382 \resetprevpage@num
383 }
384
385 %

```

\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.

```

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

```

```

399     \ifx\line@list\empty\else
400         \global\noteschanged@true
401     \fi
402 \fi
403 \ifnoteschanged@
404     \led@mess@NotesChanged
405 \fi
406 \else
407     \led@err@NumberingNotStarted
408 \fi
409 \autoparfalse
410 \if@noeled@sec\else%
411     \immediate\closeout\eled@sectioning@out%
412 \fi%
413 \ifl@dpairing\else
414     \global\l@dnumpstartsL=\z@%
415 \endgroup
416 \fi
417 }
418 %

```

\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

```

419 \newcommand{\pausenumbering}{%
420     \ifautopar\global\autopar@pausetrue\fi%
421     \endnumbering\global\numberingtrue}
422 %

```

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.

```

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

```

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

```

438 \fi%
439 \ifcontinuousnumberingwithcolumns%
440 \ifdefined\line@numR%
441 \ifnum\line@numR>\line@num%
442 \expandafter\setlinenum\expandafter{\the\line@numR}%
443 \fi%
444 \ifnum\last@page@numR>\last@page@num%
445 \global\last@page@num=\last@page@numR%
446 \fi%
447 \fi%
448 \fi%
449 \else
450 \led@err@NumberingShouldHaveStarted
451 \endnumbering
452 \beginnumbering
453 \fi}
454
455
456 %

```

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.

```

457 \newcommand*{\list@create}[1]{%
458 \global\let#1=\empty%
459 }%
460 %

```

`\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 .


```

461 \newcommand*{\list@clear}[1]{%
462   \global\let#1=\empty%
463 }
464 %

```

\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.

```

465 \newtoks\led@toksa \newtoks\led@toksb
466 \global\led@toksa={\}
467 \long\def\xright@appenditem#1\to#2{%
468   \global\led@toksb=\expandafter{#2}%
469   \xdef#2{\the\led@toksb\the\led@toksa\expandafter{#1}}%
470   \global\led@toksb={}}
471 %

```

\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.

```

472 \long\def\xleft@appenditem#1\to#2{%
473   \global\led@toksb=\expandafter{#2}%
474   \xdef#2{\the\led@toksa\expandafter{#1}\the\led@toksb}%
475   \global\led@toksb={}}
476 %

```

\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.

```

477 \def\gl@p#1\to#2{\expandafter\gl@poff#1\gl@poff#1#2}
478 \long\def\gl@poff\#1#2\gl@poff#3#4{\gdef#4{#1}\gdef#3{#2}}
479
480 %

```

V Line counting

V.1 Choosing the system of lineation

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

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

```

\ifbypstart@
\ifbypstart@true
\ifbypstart@false
  \ifbypage@
  \ifbypage@true
  \ifbypage@false

```

- 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.

```
481 \newif\ifbypage@
482 \newif\ifbypstart@
483 %
```

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@R84 \newif\ifbypage@R
\ifbypstart@R85 \newif\ifbypstart@R
486 %
```

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

```
487 \newcommand*{\lineation}[1]{%
488 %
```

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

```
489 \ifnumbering
490 \led@err@LineationInNumbered
491 \else
492 %
```

If the argument is page.

```
493 \def\@tempa{#1}\def\@tempb{page}%
494 \ifx\@tempa\@tempb
495 \global\bypage@true
496 \global\bypstart@false
497 \unless\ifnocritical@%
498 \Xpstart[] [false]%
499 \fi%
500 %
```

If the argument is pstart.

```
501 \else
502 \def\@tempb{pstart}%
503 \ifx\@tempa\@tempb
504 \global\bypage@false
505 \global\bypstart@true
506 \unless\ifnocritical@%
507 \Xpstart%
508 \fi%
509 %
```

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

```

510     \else
511         \def\@tempb{section}
512         \ifx\@tempa\@tempb
513             \global\bypage@false
514             \global\bystart@false
515             \unless\ifnocritical@%
516                 \Xpstart[] [false]%
517             \fi%
518 %

```

In other case, it is an error.

```

519     \else
520         \led@warn@BadLineation
521     \fi
522 \fi
523 \fi
524 \fi}}
525 %

```

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`

```

526 \newcount\line@margin
527
528 \newcommand*{\linenummargin}[1]{%
529     \l@dgetline@margin{#1}%
530     \ifnum\@l@dttempcntb>\m@ne
531         \ifledRcol
532             \global\line@marginR=\@l@dttempcntb
533             \led@warn@setting@in@rightside{\linenummargin}%
534         \else
535             \global\line@margin=\@l@dttempcntb
536         \fi
537     \fi}}
538
539 \newcommand*{\l@dgetline@margin}[1]{%
540     \def\@tempa{#1}\def\@tempb{left}%
541     \ifx\@tempa\@tempb
542         \@l@dttempcntb \z@
543     \else
544         \def\@tempb{right}%
545         \ifx\@tempa\@tempb
546             \@l@dttempcntb \@ne
547         \else
548             \def\@tempb{outer}%

```

```

549     \ifx\@tempa\@tempb
550       \l@dttempcntb \tw@
551     \else
552       \def\@tempb{inner}%
553       \ifx\@tempa\@tempb
554         \l@dttempcntb \thr@@
555       \else
556         \led@warn@BadLinenummargin
557         \l@dttempcntb \m@ne
558       \fi
559     \fi
560   \fi
561 \fi}
562
563 %

```

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.

```

564 \newcounter{firstlinenum}
565   \setcounter{firstlinenum}{5}
566 \newcounter{linenumincrement}
567   \setcounter{linenumincrement}{5}
568 %

```

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

```

569 \newcounter{firstsublinenum}
570   \setcounter{firstsublinenum}{5}
571 \newcounter{sublinenumincrement}
572   \setcounter{sublinenumincrement}{5}
573
574 %

```

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

```

575 \newcommand*{\firstlinenum}[1]{%
576   \ifledRcol%
577     \setcounter{firstlinenumR}{#1}%
578     \led@warn@setting@in@rightside{\firstlinenum}%
579   \else%
580     \setcounter{firstlinenum}{#1}%
581   \fi

```

```

582 \fi%
583 }
584 \newcommand*{\linenumincrement}[1]{%
585 \ifledRcol%
586 \setcounter{linenumincrementR}{#1}%
587 \led@warn@setting@in@rightside{\linenumincrement}%
588 \else%
589 \setcounter{linenumincrement}{#1}%
590 \fi%
591 }
592 \newcommand*{\firstsublinenum}[1]{%
593 \ifledRcol%
594 \setcounter{firstsublinenumR}{#1}%
595 \led@warn@setting@in@rightside{\firstsublinenum}%
596 \else%
597 \setcounter{firstsublinenum}{#1}%
598 \fi%
599 }
600 \newcommand*{\sublinenumincrement}[1]{%
601 \ifledRcol%
602 \setcounter{sublinenumincrementR}{#1}%
603 \led@warn@setting@in@rightside{\sublinenumincrement}%
604 \else%
605 \setcounter{sublinenumincrement}{#1}%
606 \fi%
607 }
608 %
609 %

```

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.

```

610 \newcount\lock@disp
611 \newcommand{\lockdisp}[1]{%
612 \l@dgetlock@disp{#1}%
613 \ifnum\@l@dttempcntb>\m@ne
614 \global\lock@disp=\@l@dttempcntb
615 \else
616 \led@warn@BadLockdisp
617 \fi}}
618 \newcommand*{\l@dgetlock@disp}[1]{
619 \def\@tempa{#1}\def\@tempb{first}%
620 \ifx\@tempa\@tempb
621 \@l@dttempcntb \z@

```

```

622 \else
623   \def\@tempb{last}%
624   \ifx\@tempa\@tempb
625     \l@dttempcntb \@ne
626   \else
627     \def\@tempb{all}%
628     \ifx\@tempa\@tempb
629       \l@dttempcntb \tw@
630     \else
631       \l@dttempcntb \m@ne
632     \fi
633   \fi
634 \fi}
635
636 %

```

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

```

637 \newcount\sublock@disp
638 \newcommand{\sublockdisp}[1]{\%
639   \l@dtgetlock@disp{#1}%
640   \ifnum\@l@dttempcntb>\m@ne
641     \global\sublock@disp=\@l@dttempcntb
642   \else
643     \led@warn@BadSublockdisp
644   \fi}}
645
646 %

```

V.5 Line number style

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

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

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

`\sublinenumrep`

`\sublinenumr@p`

```

647 \newcommand*{\linenumberstyle}[1]{\%
648   \def\linenumrep##1{\@nameuse{@#1}{##1}}}
649 \newcommand*{\sublinenumberstyle}[1]{\%
650   \def\sublinenumrep##1{\@nameuse{@#1}{##1}}}
651 %

```

Initialise the number styles to arabic.

```

652 \linenumberstyle{arabic}
653 \let\linenumr@p\linenumrep

```

```

654 \sublinenumberstyle{arabic}
655 \let\sublinenumr@p\sublinenumrep
656
657 %

```

V.6 Line number printing

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

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

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

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

```

658 \newlength{\linenumsep}
659 \setlength{\linenumsep}{1pc}
660 \newcommand*{\numlabfont}{\normalfont\scriptsize}
661 \newcommand*{\ledlinenum}{%
662   \bgroup%
663   \ifluatex%
664     \texdir TL%
665   \fi%
666   \numlabfont\linenumrep{\line@num}%
667   \ifsublines@
668     \ifnum\subline@num>0\relax
669       \unskip%
670       \Xsublinesep@side%
671       \sublinenumrep{\subline@num}%
672   \fi
673   \fi%
674   \egroup%
675 }%
676
677 \newcommand*{\leftlinenum}{%
678   \ledlinenum
679   \kern\linenumsep}
680 \newcommand*{\rightlinenum}{%
681   \kern\linenumsep
682   \ledlinenum}
683

```

```
684 %
```

V.7 Line number counters and lists

Footnote references using line numbers rather than symbols can't be generated in one pass, because we do not know the line numbers till we ship out the pages. It would be possible if footnotes were never keyed to more than one line; but some footnotes gloss passages that may run for several lines, and they must be tied to the first line of the passage glossed. And even one-line passages require two passes if we want line-per-page numbering rather than line-per-section numbering.

So we run \LaTeX over the text several times, and each time save information about page and line numbers in a 'line-list file' to be used during the next pass. At the start of each section—whenever `\beginnumbering` is executed—the line-list file for that section is read, and the information from it is encoded into a few list macros.

We need first to define the different line numbers that are involved in these macros, and the associated counters.

`\line@num` The count `\line@num` stores the line number that is used in marginal line numbering and in notes: counting either by section, page or pstart, depending on your choice for this section. This may be qualified by `\subline@num`.

```
685 \newcount\line@num
686 %
```

`\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.

```
687 \newcount\subline@num
688 %
```

`\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-lined 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.

```
689 \newif\ifsublines@
690 %
```

`\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.

```
691 \newcount\absline@num
692 %
```

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.

```
693 \newcount\@lock
694 \newcount\sub@lock
695 %
```

V.9 Line number associated to lemma

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

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

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

```
23|35|0|24|3|0|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 `\advance line` and `\set line` 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:

```
696 \list@create{\line@list}
697 \list@create{\insertlines@list}
698 \list@create{\actionlines@list}
699 \list@create{\actions@list}
700
701 %
```

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

```
702 \newcount\page@num
703 \newcount\endpage@num
704 \newcount\endline@num
705 \newcount\endsubline@num
706 %
```

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

```

707 \newif\ifnoteschanged@
708 %

```

\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@09 \newcommand*{\resetprevline@}{%
710 \def\do##1{\global\csundef{prevline##1}}%
711 \dolistloop{\@series}%
712 }
713 %

```

\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@14 \newcommand*{\resetprevpage@num}{%
715 \def\do##1{\ifcsdef{prevpage##1@num}{\global\csname prevpage##1@num\
endcsname=0}{}}%
716 \dolistloop{\@series}%
717 }
718 %

```

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.

```

719 \newread\@inputcheck
720 \newcommand*{\read@linelist}[1]{%
721 \ifledRcol%
722 \list@clearing@regR%
723 \else%
724 \list@clearing@reg%
725 \fi%
726 %

```

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).

```

727 \list@clear{\maxlinesinpar@list}
728 %

```

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

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

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

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

```

729 \get@linelistfile{#1}%
730 \ifcontinuousnumberingwithcolumns
731     \global\page@numR=\page@numR\relax
732     \global\last@page@numR=\last@page@numR\relax
733     \global\page@num=\page@num\relax
734     \global\last@page@num=\last@page@num\relax
735 \fi
736 \endgroup
737 %

```

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.

```

738 \ifledRcol
739     \global\page@numR=\m@ne
740     \ifx\actionlines@listR\empty
741         \gdef\next@actionlineR{1000000}%
742     \else
743         \glp\actionlines@listR\to\next@actionlineR
744         \glp\actions@listR\to\next@actionR
745     \fi
746 \else
747     \global\page@num=\m@ne
748     \ifx\actionlines@list\empty
749         \gdef\next@actionline{1000000}%
750     \else
751         \glp\actionlines@list\to\next@actionline
752         \glp\actions@list\to\next@action
753     \fi
754 \fi
755 }
756 %

```

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

```

757 \newcommand*\list@clearing@reg{%
758   \list@clear{\line@list}%
759   \list@clear{\insertlines@list}%
760   \list@clear{\actionlines@list}%
761   \list@clear{\actions@list}%
762   \list@clear{\linesinpar@listL}%
763   \list@clear{\linesonpage@listL}%
764 }%
765 %

```

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

```

766 \newcommand*\get@linelistfile}[1]{%
767   \InputIfFileExists{#1}{%
768     \global\noteschanged@false
769     \begingroup
770       \catcode`\[=1 \catcode`\]=2
771       \makeatletter \catcode`\^M=9}{%
772     \led@warn@NoLineFile{#1}%
773     \global\noteschanged@true
774     \begingroup}%
775 }
776
777 %

```

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

```

778 \newcommand{\line@list@version}[1]{%
779   \IfStrEq{#1}{\this@line@list@version}%
780   {}%
781   {\ifledRcol%
782     \led@warn@Obsolete{\jobname.\extensionchars\the\section@num}%
783     \else%
784     \led@warn@Obsolete{\jobname.\extensionchars\the\section@num}%
785     \fi%
786     \endinput%
787   }%
788 }%
789 %

```

`\@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`.

```

790
791 \newcommand*{\@nl}[2]{%
792   \fix@page{#1}%
793   \ifledRcol%
794     \@nl@regR%
795   \else%
796     \@nl@reg%
797   \fi%
798 }
799 \newcommand*{\@nl@reg}{%
800   \ifx\l@dchset@num\relax \else
801     \advance\absline@num \@ne

```

```

802 \set@line@action
803 \let\l@dchset@num=\relax
804 \advance\absline@num \m@ne
805 \advance\line@num \m@ne
806 \fi
807 %

```

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

```

808 \advance\absline@num \@ne
809 \ifx\next@page@num\relax \else
810 \page@action
811 \let\next@page@num=\relax
812 \fi
813 \ifx\sub@change\relax \else
814 \ifnum\sub@change>\z@
815 \sublines@true
816 \else
817 \sublines@false
818 \fi
819 \sub@action
820 \let\sub@change=\relax
821 \fi
822 %

```

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

```

823 \ifcase\@lock
824 \or
825 \@lock \tw@
826 \or \or
827 \@lock \z@
828 \fi
829 \ifcase\sub@lock
830 \or
831 \sub@lock \tw@
832 \or \or
833 \sub@lock \z@
834 \fi
835 %

```

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

```

836 \ifsublines@
837 \ifnum\sub@lock<\tw@
838 \advance\subline@num \@ne
839 \fi
840 \else
841 \ifnum\@lock<\tw@
842 \advance\line@num \@ne \subline@num \z@

```



```

843         \fi
844     \fi}
845
846 %

```

`\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`.

```

847 \newcount\last@page@num
848 \last@page@num=-10000
849
850 \newcommand*{\fix@page}[1]{%
851     \ifledRcol
852         \ifnum #1=\last@page@numR
853         \else
854             \ifbypage@R
855                 \line@numR \z@ \subline@numR \z@
856             \fi
857             \page@numR=#1\relax
858             \last@page@numR=#1\relax
859             \def\next@page@numR{#1}%
860         \fi
861     \else
862         \ifnum #1=\last@page@num
863         \else
864             \ifbypage@
865                 \line@num \z@ \subline@num \z@
866             \fi
867             \page@num=#1\relax
868             \last@page@num=#1\relax
869             \def\next@page@num{#1}%
870             \listxadd{\normal@page@break}{\the\absline@num}
871         \fi
872     \fi}
873 %

```

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

```

874 \@lopR \newcommand*{\@pend}[1]{}
875 \@pendR \newcommand*{\@pendR}[1]{}
876 \@lopL \newcommand*{\@lopL}[1]{}
877 \@lopR \newcommand*{\@lopR}[1]{}
878
879 %

```

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

```

880 \newcommand*{\sub@on}{\ifsublines@
881   \let\sub@change=\relax
882   \else
883     \def\sub@change{1}%
884   \fi}
885 \newcommand*{\sub@off}{\ifsublines@
886   \def\sub@change{-1}%
887   \else
888     \let\sub@change=\relax
889   \fi}
890
891 %

```

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

```

892
893 \newcommand*{\@adv}[1]{%
894   \ifsublines@
895     \ifledRcol
896       \advance\subline@numR by #1\relax
897       \ifnum\subline@numR<\z@
898         \led@warn@BadAdvancelineSubline
899         \subline@numR \z@
900       \fi
901     \else
902       \advance\subline@num by #1\relax
903       \ifnum\subline@num<\z@
904         \led@warn@BadAdvancelineSubline
905         \subline@num \z@
906       \fi
907     \fi
908   \else
909     \ifledRcol
910       \advance\line@numR by #1\relax
911       \ifnum\line@numR<\z@
912         \led@warn@BadAdvancelineLine
913         \line@numR \z@
914       \fi
915     \else
916       \advance\line@num by #1\relax
917       \ifnum\line@num<\z@
918         \led@warn@BadAdvancelineLine
919         \line@num \z@
920       \fi
921     \fi
922   \fi
923   \set@line@action}
924
925 %

```

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

```

926
927 \newcommand*{\@set}[1]{%
928   \ifledRcol
929     \ifsublines@
930       \subline@numR=#1\relax
931     \else
932       \line@numR=#1\relax
933     \fi
934     \set@line@action
935   \else
936     \ifsublines@
937       \subline@num=#1\relax
938     \else
939       \line@num=#1\relax
940     \fi
941     \set@line@action
942   \fi}
943
944 %

```

\l@d@set The `\l@d@set{<num>}` macro sets the line number for the next `\pstart` to the value specified as its argument. This is used to implement `\setlinenum`.
\l@dchset@num `\l@dchset@num` is a flag to the `\@nl?` macro. If it is not `\relax` then a linenum change is to be done.

```

945
946 \newcommand*{\l@d@set}[1]{%
947   \ifledRcol
948     \line@numR=#1\relax
949     \advance\line@numR \@ne
950     \def\l@dchset@num{#1}
951   \else
952     \line@num=#1\relax
953     \advance\line@num \@ne
954     \def\l@dchset@num{#1}
955   \fi}
956 \let\l@dchset@num\relax
957
958 %

```

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

```

959
960 \newcommand*{\page@action}{%
961   \ifledRcol
962     \xright@appenditem{\the\absline@numR}\to\actionlines@listR
963     \xright@appenditem{\next@page@numR}\to\actions@listR

```

```

964 \else
965 \xright@appenditem{\the\absline@num}\to\actionlines@list
966 \xright@appenditem{\next@page@num}\to\actions@list
967 \fi}
968 %

```

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

```

969
970 \newcommand*{\set@line@action}{%
971 \ifledRcol
972 \xright@appenditem{\the\absline@numR}\to\actionlines@listR
973 \ifsublines@
974 \@l@tempcnta=-\subline@numR
975 \else
976 \@l@tempcnta=-\line@numR
977 \fi
978 \advance\@l@tempcnta by -5000\relax
979 \xright@appenditem{\the\@l@tempcnta}\to\actions@listR
980 \else
981 \xright@appenditem{\the\absline@num}\to\actionlines@list
982 \ifsublines@
983 \@l@tempcnta=-\subline@num
984 \else
985 \@l@tempcnta=-\line@num
986 \fi
987 \advance\@l@tempcnta by -5000\relax
988 \xright@appenditem{\the\@l@tempcnta}\to\actions@list
989 \fi}
990 %

```

\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.

```

991
992 \newcommand*{\sub@action}{%
993 \ifledRcol
994 \xright@appenditem{\the\absline@numR}\to\actionlines@listR
995 \ifsublines@
996 \xright@appenditem{-1001}\to\actions@listR
997 \else
998 \xright@appenditem{-1002}\to\actions@listR
999 \fi
1000 \else
1001 \xright@appenditem{\the\absline@num}\to\actionlines@list
1002 \ifsublines@
1003 \xright@appenditem{-1001}\to\actions@list
1004 \else
1005 \xright@appenditem{-1002}\to\actions@list

```

```

1006 \fi
1007 \fi}
1008 %

```

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

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

```

1009 \newcommand*{\lock@on}{\futurelet\next\do@lockon}
1010
1011 \newcommand*{\do@lockon}{%
1012 \ifx\next\lock@off
1013 \global\let\lock@off=\skip@lockoff
1014 \else
1015 \ifledRcol
1016 \do@lockonR
1017 \else
1018 \do@lockonL
1019 \fi
1020 \fi}
1021
1022
1023 \newcommand*{\do@lockonL}{%
1024 \xright@appenditem{\the\absline@num}\to\actionlines@list
1025 \ifsublines@
1026 \xright@appenditem{-1005}\to\actions@list
1027 \ifnum\sub@lock=\z@
1028 \sub@lock \@ne
1029 \else
1030 \ifnum\sub@lock=\thr@@
1031 \sub@lock \@ne
1032 \fi
1033 \fi
1034 \else
1035 \xright@appenditem{-1003}\to\actions@list
1036 \ifnum\@lock=\z@
1037 \@lock \@ne
1038 \else
1039 \ifnum\@lock=\thr@@
1040 \@lock \@ne
1041 \fi
1042 \fi
1043 \fi}
1044
1045 %

```

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

```

1046 \do@lockoff \newcommand*{\do@lockoffL}{%
1047 \xright@appenditem{\the\absline@num}\to\actionlines@list
1048 \ifsublines@
1049   \xright@appenditem{-1006}\to\actions@list
1050   \ifnum\sub@lock=\tw@
1051     \sub@lock \thr@@
1052   \else
1053     \sub@lock \z@
1054   \fi
1055 \else
1056   \xright@appenditem{-1004}\to\actions@list
1057   \ifnum\@lock=\tw@
1058     \@lock \thr@@
1059   \else
1060     \@lock \z@
1061   \fi
1062 \fi}
1063
1064 \newcommand*{\do@lockoff}{%
1065   \ifledRcol
1066     \do@lockoffR
1067   \else
1068     \do@lockoffL
1069   \fi}
1070 \newcommand*{\skip@lockoff}{\global\let\lock@off=\do@lockoff}
1071 \global\let\lock@off=\do@lockoff
1072
1073 %

```

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

```

1074 \newcommand*{\n@num}{%
1075   \ifledRcol%
1076     \xright@appenditem{\the\absline@numR}\to\actionlines@listR
1077     \xright@appenditem{-1007}\to\actions@listR
1078   \else%
1079     \xright@appenditem{\the\absline@num}\to\actionlines@list%
1080     \xright@appenditem{-1007}\to\actions@list%
1081   \fi%
1082 }%
1083
1084 %

```

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

```

1085 \newcommand*{\n@num@stanza}{%

```

```

1086 \ifledRcol%
1087 \xright@appenditem{\the\absline@numR}\to\actionlines@listR%
1088 \xright@appenditem{-1008}\to\actions@listR%
1089 \else%
1090 \xright@appenditem{\the\absline@num}\to\actionlines@list%%
1091 \xright@appenditem{-1008}\to\actions@list%
1092 \fi%
1093 }
1094 %

```

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

`\hidenum`

```

\h@num%5 \newif\ifl@dhidenumber
1096 \newcommand*{\hidenum}{
1097 \ifledRcol%
1098 \write\linenum@outR{\string\hide@num}%
1099 \else%
1100 \write\linenum@out{\string\hide@num}%
1101 \fi%
1102 }%
1103 \newcommand*{\hide@num}{%
1104 \ifledRcol%
1105 \xright@appenditem{\the\absline@numR}\to\actionlines@listR%
1106 \xright@appenditem{-1009}\to\actions@listR%
1107 \else%
1108 \xright@appenditem{\the\absline@num}\to\actionlines@list%%
1109 \xright@appenditem{-1009}\to\actions@list%
1110 \fi%
1111 }
1112 %

```

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

`\insert@count` 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`.

```

1113 \newcount\insert@count
1114 %

```

- #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.

```

1115 \newcommand*{\dummy@ref}[2]{#2}
1116 %

```

`\@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.

```

1117 \newcommand*{\@ref}[2]{%
1118   \ifledRcol%
1119     \@ref@regR{#1}{#2}%
1120   \else%
1121     \@ref@reg{#1}{#2}%
1122   \fi%
1123 }%
1124 \newcommand*{\@ref@reg}[2]{%
1125   \global\insert@count=#1\relax
1126   \global\advance\@edtext@level by 1%
1127   \loop\ifnum\insert@count>\z@
1128     \xright@appenditem{\the\absline@num}\to\insertlines@list
1129     \global\advance\insert@count \m@ne
1130   \repeat
1131 %

```

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.

```

1132 \begingroup
1133   \let\@ref=\dummy@ref
1134   \let\@lopL\@gobble
1135   \let\page@action=\relax
1136   \let\sub@action=\relax
1137   \let\set@line@action=\relax
1138   \let\@lab=\relax
1139   \let\@lemma=\relax%
1140   \let\@sw\@gobblethree%
1141   #2
1142   \global\endpage@num=\page@num
1143   \global\endline@num=\line@num
1144   \global\endsubline@num=\subline@num
1145 \endgroup
1146 %

```

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

```

1147 \xright@appenditem%
1148   {\the\page@num|\the\line@num|%
1149     \ifsublines@ \the\subline@num \else 0\fi}%
1150   {\the\endpage@num|\the\endline@num|%
1151     \ifsublines@ \the\endsubline@num \else 0\fi}}\to\line@list
1152 %

```

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.


```

1153 \expandafter\list@create\expandafter{\csname sw@list@edtext@tmp@the\
@edtext@level\endcsname}%
1154 \providebool{lemmacommand@the\@edtext@level}%
1155 \boolfalse{lemmacommand@the\@edtext@level}%
1156 %

```

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

```

1157 #2%
1158 %

```

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.

```

1159 \ifnum\@edtext@level>0%
1160 \def\create@this@edtext@level{\expandafter\list@create\expandafter{\
csname sw@list@edtext@the\@edtext@level\endcsname}}%
1161 \ifcsundef{sw@list@edtext@the\@edtext@level}{\create@this@edtext@level
}{}%
1162 \letcs{\@tmp}{sw@list@edtext@the\@edtext@level}%
1163 \letcs{\@tmpp}{sw@list@edtext@tmp@the\@edtext@level}%
1164 \xright@appenditem{\expandonce\@tmpp}\to\@tmp%
1165 \global\cslet{sw@list@edtext@the\@edtext@level}{\@tmp}%
1166 \fi%
1167 %

```

Decrease edtext level counter.

```

1168 \global\advance\@edtext@level by -1%
1169 %
1170 }
1171 %
1172 %

```

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.

```

1173 \newwrite\linenum@out
1174 %

```

`\iffirst@linenum@out@` Once any file is opened on this stream, we keep it open forever, or else switch to another
`\first@linenum@out@true` file that we keep open. The reason is that we want the output routine to write the page
`\first@linenum@out@false` number for every page to this file; otherwise we would have to write it at the start of every line. But it is not very easy for the output routine to tell whether an output stream

is open or not. There is no way to test the status of a particular output stream directly, and the asynchronous nature of output routines makes the status hard to determine by other means.

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

```
1175 \newif\iffirst@linenum@out@
1176 \first@linenum@out@true
1177 %
```

`\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.

```
1178 \newcommand{\this@line@list@version}{5}%
1179 %
```

`\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.

```
1180 \newcommand*{\line@list@stuff}[1]{%
1181 %
```

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

```
1182 \read@linelist{#1}%
1183 %
```

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.

```
1184 \iffirst@linenum@out@
1185 \immediate\closeout\linenum@out%
1186 \global\first@linenum@out@false%
1187 \immediate\openout\linenum@out=#1\relax%
1188 \immediate\write\linenum@out{\string\line@list@version{\
this@line@list@version}}%
1189 \ifl@dpaging%
1190 \immediate\write\linenum@out{\string\@par@sync@option{\
@par@this@sync@option}}%
1191 \fi%
1192 \else
1193 %
```

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.

```

1194     \if@minipage%
1195         \leavevmode%
1196     \fi%
1197     \closeout\linenum@out%
1198     \openout\linenum@out=#1\relax%
1199     \write\linenum@out{\string\line@list@version{\this@line@list@version}}
1200 %
1201     \ifl@dpaging%
1202         \write\linenum@out{\string\@par@sync@option{\@par@this@sync@option}}
1203 %
1204     \fi%
1205 \fi}

```

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

```

1206 \newcommand*{\new@line}{%
1207     \IfStrEq{\led@pb@setting}{after}%
1208     {\xifinlist{\the\absline@num}{\l@prev@nopb}%
1209     {\xifinlist{\the\absline@num}{\normal@page@break}%
1210     {\numgdef{\@next@page}{\c@page+1}%
1211     \write\linenum@out{\string\@nl[\@next@page][\@next@page]}}%
1212     }%
1213     {\write\linenum@out{\string\@nl[\the\c@page][\thepage]}}}%
1214     }%
1215     {\write\linenum@out{\string\@nl[\the\c@page][\thepage]}}}%
1216     }%
1217     \IfStrEq{\led@pb@setting}{before}%
1218     {\numgdef{\next@absline}{\the\absline@num+1}%
1219     \xifinlist{\next@absline}{\l@prev@nopb}%
1220     {\xifinlist{\the\absline@num}{\normal@page@break}%
1221     {\numgdef{\nc@page}{\c@page+1}%
1222     \write\linenum@out{\string\@nl[\nc@page][\nc@page]}}%
1223     }%
1224     {\write\linenum@out{\string\@nl[\the\c@page][\thepage]}}}%
1225     }%
1226     {\write\linenum@out{\string\@nl[\the\c@page][\thepage]}}}%
1227     }%
1228     {}%
1229     \IfStrEqCase{\led@pb@setting}{\{before\}{\relax\}{after\}{\relax\}}{\write\linenum@out{\string\@nl[\the\c@page][\thepage]}}%
1230 }
1231
1232 %

```

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

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

```

1233 \newif\if@noneed@Footnote%
1234
1235 \newcommand*{\flag@start}{%
1236   \ifledRcol%
1237     \edef\next{\write\linenum@outR{%
1238       \string\@ref[\the\insert@countR] []}%
1239     \next%
1240     \ifnum\insert@countR<1%
1241       \if@noneed@Footnote\else%
1242         \led@err@EdtextWithoutFootnote%
1243       \fi%
1244     \fi%
1245   \else%
1246     \edef\next{\write\linenum@out{%
1247       \string\@ref[\the\insert@count] []}%
1248     \next%
1249     \ifnum\insert@count<1%
1250       \if@noneed@Footnote\else%
1251         \led@err@EdtextWithoutFootnote%
1252       \fi%
1253     \fi%
1254   \fi}%
1255 %
1256 %

```

`\startsub` `\startsub` and `\endsub` turn sub-lineation on and off, by writing appropriate instructions to the line-list file. When sub-lineation is in effect, the line number counter is frozen and the sub-line counter advances instead. If one of these commands appears in the middle of a line, it does not take effect until the next line; in other words, a line is counted as a line or sub-line depending on what it started out as, even if that changes in the middle.

We tinker with `\lastskip` because a command of either sort really needs to be attached to the last word preceding the change, not the first word that follows the change. This is because sub-lineation will often turn on and off in mid-line—stage directions, for example, often are mixed with dialogue in that way—and when a line is mixed we want to label it using the system that was in effect at its start. But when sub-lineation begins at the very start of a line we have a problem, if we don't put in this code.

```

1257
1258
1259 \newcommand*{\startsub}{\dimen0\lastskip
1260   \ifdim\dimen0>0pt \unskip \fi

```

```

1261 \ifledRcol \write\linenum@outR{\string\sub@on}%
1262 \else      \write\linenum@out{\string\sub@on}%
1263 \fi
1264 \ifdim\dimen0>0pt \hskip\dimen0 \fi}
1265 \def\endsub{\dimen0\lastskip
1266 \ifdim\dimen0>0pt \unskip \fi
1267 \ifledRcol \write\linenum@outR{\string\sub@off}%
1268 \else      \write\linenum@out{\string\sub@off}%
1269 \fi
1270 \ifdim\dimen0>0pt \hskip\dimen0 \fi}
1271
1272 %

```

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

```

1273 \newcommand*{\advanceline}[1]{\leavevmode%
1274 \ifledRcol \write\linenum@outR{\string\@adv[#1]}%
1275 \else      \write\linenum@out{\string\@adv[#1]}%
1276 \fi%
1277 }
1278 %

```

\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.

```

1279
1280 \newcommand*{\setline}[1]{%
1281 \leavevmode%
1282 \ifnum#1<\z@
1283 \led@warn@BadSetline
1284 \else
1285 \ifledRcol \write\linenum@outR{\string\@set[#1]}%
1286 \else      \write\linenum@out{\string\@set[#1]}%
1287 \fi
1288 \fi}
1289
1290 %

```

\setlinenum You can use `\setlinenum{<num>}` before a `\pstart` to set the visible line-number to a specified positive value. It writes a `\led@set` command to the line-list file.

```

1291
1292 \newcommand*{\setlinenum}[1]{%
1293 \ifnum#1<\z@
1294 \led@warn@BadSetlinenum
1295 \else
1296 \ifledRcol \write\linenum@outR{\string\led@set[#1]}
1297 \else      \write\linenum@out{\string\led@set[#1]} \fi
1298 \fi}

```

```
1299
1300 %
```

`\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.

```
1301
1302 \newcommand*\startlock{%
1303   \ifledRcol \write\linenum@outR{\string\lock@on}%
1304   \else      \write\linenum@out{\string\lock@on}%
1305   \fi}
1306 \def\endlock{%
1307   \ifledRcol \write\linenum@outR{\string\lock@off}%
1308   \else      \write\linenum@out{\string\lock@off}%
1309   \fi}
1310 %
```

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

```
1311 \l@dskipnumbertrue \newif\ifl@dskipnumber
1312 \l@dskipnumberfalse \newif\ifl@dskipversenumber%
1313 \skipnumbering \newcommand*\skipnumbering{%
1314   \leavevmode%
1315   \ifledRcol%
1316     \ifinstanza%
1317       \write\linenum@outR{\string\n@num@stanza}%
1318     \else%
1319       \write\linenum@outR{\string\n@num}%
1320     \fi%
1321     \advanceline{-1}%
1322   \else%
1323     \ifinstanza%
1324       \write\linenum@out{\string\n@num@stanza}%
1325     \else%
1326       \write\linenum@out{\string\n@num}%
1327     \fi%
1328     \advanceline{-1}%
1329   \fi%
1330 }%
1331
1332 %
```

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

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

VI.1 `\edtext` itself

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

macro or token register, which in turn would be formatted and output when all of #2 for the lemma has been read.

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

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

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

```
1333 \list@create{\end@lemmas}
1334 %
```

`\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.

```
1335 \newcommand{\dummy@edtext}[2]{#1}
1336 %
```

`\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`.

```
1337 \newcommand{\dummy@edtext@showlemma}[2]{\showlemma{#1}}%
1338 %
```

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

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

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

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

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 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 \TeX .)

```

1339 \newcommand*{\no@expands}{%
1340   \let\select@lemmafnt=0%
1341   \let\startsub=\relax   \let\endsub=\relax
1342   \let\startlock=\relax  \let\endlock=\relax
1343   \let\edlabel=\@gobble
1344   \let\setline=\@gobble  \let\advanceline=\@gobble
1345   \let\sameword\sameword@inedtext%
1346   \let\edtext=\dummy@edtext
1347   \l@dtabnoexpands
1348   \morenoexpands}
1349 \let\morenoexpands=\relax
1350
1351 %
```

`\@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.

```

1352 \newcommand{\@tag}{}

```

```
1353 %
```

\@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.

```
1354 \newcount\@edtext@level%
1355 \@edtext@level=0%
1356 %
```

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

```
1357 \newcounter{edtext}
1358 \renewcommand{\theedtext}{\edtxt@arabic{edtext}}%
1359 %
```

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

```
1360 \newcommand{\edtext}[2]{\leavevmode%
1361 %
```

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

```
1362 \ifnumberedpar%
1363 %
```

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

```
1364 \global\advance\@edtext@level by 1%
1365 %
```

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

```
1366 \stepcounter{edtext}%
1367 %
```

By default, we do not use `\lemma`

```
1368 \global\@lemmacommand@false%
1369 %
```

```
1370 \begingroup%
1371 %
```

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`.

```

1372     \ifledRcol%
1373         \ifcsvoid{sw@list@edtextR@the\@edtext@level}%
1374             {\global\let\sw@inthisedtext\empty}%
1375             {\expandafter\gl@p\csname sw@list@edtextR@the\@edtext@level\
endcsname\to\sw@inthisedtext}%
1376         \else%
1377             \ifcsvoid{sw@list@edtext@the\@edtext@level}%
1378                 {\global\let\sw@inthisedtext\empty}%
1379                 {\expandafter\gl@p\csname sw@list@edtext@the\@edtext@level\
endcsname\to\sw@inthisedtext}%
1380         \fi%
1381     %

```

\@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.

```

1382     \global\renewcommand{\@tag}{%
1383         \no@expands #1%
1384     }%
1385 %

```

\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`.

```

1386     \set@line%
1387 %

```

`\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`.

```

1388     \ifledRcol \global\insert@countR \z@%
1389     \else      \global\insert@count \z@ \fi%
1390 %

```

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.

```

1391     \ignorespaces #2\relax%
1392 %

```

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

```

1393 \ifundefined{xpg@main@language}{%if not polyglossia
1394 \flag@start}%
1395 {\ifRTL\flag@end\else\flag@start\fi%
1396 }%
1397 %

```

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

```

1398 \if@lemmacommand{%
1399 \ifledRcol%
1400 \write\linenum@outR{\string\@lemma}%
1401 \else%
1402 \write\linenum@out{\string\@lemma}%
1403 \fi%
1404 \fi%
1405 %

```

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.

```

1406 \endgroup%
1407 \ifdef{\hypertarget}%
1408 {%
1409 \csedef{thisedtext@the\@edtext@level}{\theedtext}%
1410 \Hy@raisedlink@left{\hypertarget{\csuse{thisedtext@the\
@edtext@level}:start}}}%
1411 \showlemma{#1}%
1412 \Hy@raisedlink{\hypertarget{\csuse{thisedtext@the\@edtext@level}:
end}}}%
1413 }%
1414 {%
1415 \showlemma{#1}%
1416 }%
1417 %

```

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.

```

1418 \ifx\end@lemmas\empty \else%
1419 \glp\end@lemmas\to\x@lemma%
1420 \x@lemma%
1421 \global\let\x@lemma=\relax%
1422 \fi%
1423 \ifundefined{xpg@main@language}{%if not polyglossia
1424 \flag@end}%

```

```

1425     {\if@RTL\flag@start\else\flag@end\fi% With polyglossia, you must
      track whether the language reads left to right (English) or right to left
      (Arabic).
1426     }%
1427 %

```

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`.

```

1428     \global\@noneed@Footnotefalse%
1429     \global\advance\@edtext@level by -1%
1430     \global\@lemmacommand@false%
1431 %

```

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

```

1432     \else%
1433     \showlemma{#1} (\textbf{\textsc{Edtext outside numbered paragraph}})\
      led@err@edtextoutsidestart%
1434     \fi%
1435 }%
1436
1437 \newcommand*{\flag@end}{%
1438     \ifledRcol%
1439         \write\linenum@outR{]}%
1440     \else%
1441         \write\linenum@out{]}%
1442     \fi}%
1443
1444 %

```

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

```

1445 \newif\ifnumberline
1446 \numberlinetrue
1447 %

```

`\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.

```

1448 \newcommand*{\set@line}{%
1449   \ifledRcol
1450     \ifx\line@listR\empty
1451       \global\noteschanged@true
1452       \xdef\l@d@nums{000|000|000|000|000|000|000|\edfont@info}%
1453     \else
1454       \gl@p\line@listR\to\@tempb
1455       \xdef\l@d@nums{\@tempb|\edfont@info}%
1456       \global\let\@tempb=\undefined
1457     \fi
1458   \else
1459     \ifx\line@list\empty
1460       \global\noteschanged@true
1461       \xdef\l@d@nums{000|000|000|000|000|000|000|\edfont@info}%
1462     \else
1463       \gl@p\line@list\to\@tempb
1464       \xdef\l@d@nums{\@tempb|\edfont@info}%
1465       \global\let\@tempb=\undefined
1466     \fi
1467   \fi}
1468
1469 %

```

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

```

1470 \newcommand*{\edfont@info}{\f@encoding/\f@family/\f@series/\f@shape}
1471
1472 %

```

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. 115).

```

1473 \newcommand*{\lemma}[1]{%
1474   \global\@lemmacommand@true%
1475   \global\renewcommand{\@tag}{%
1476     \no@expands #1%
1477   }%
1478   \ignorespaces%
1479 }%
1480 %

```

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

`\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.

VI.3 Substitute line numbers

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. 89): the starting page, line, and sub-line numbers, followed by the ending page, line, and sub-line numbers, and then the font specifier for the lemma. However, you can omit any parameters you do not want to change, and you can omit a string of vertical bars at the end of the argument. Hence `\linenum{18|4|0|18|7|1|0}` is an invocation that changes all the parameters, but `\linenum{13}` only changes the starting line number, and leaves the rest unaltered.

`\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.

```

1492 \def\line@set#1|#2\\#3|#4\\{%
1493   \gdef\@tempb{#1}%
1494   \ifx\@tempb\empty
1495     \l@{@add{#3}%
1496   \else
1497     \l@{@add{#1}%
1498   \fi
1499   \gdef\@tempb{#4}%
1500   \ifx\@tempb\empty\else
1501     \l@{@add{}}\line@set#2\\#4\\}%
1502   \fi}
1503 %

```

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

```
1504 \newcommand{\l@d@add}[1]{\xdef\l@d@nums{\l@d@nums#1}}
1505
1506 %
```

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

```

1507 \newcommand{\get@sw@txt}[1]{%
1508   \ifxetex%
1509     \xdef\sw@txt{#1}%
1510   \else%
1511     \expandafter\xdef\expandafter\sw@txt\expandafter{\detokenize{#1}}%
1512   \fi%
1513 }%
1514 %

```

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

```

1515 \newcommandx{\sameword}[2][1,usedefault]{%
1516   \leavevmode%
1517   \get@sw@txt{#2}%
1518 %

```

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

```

1519   \unless\ifledRcol%
1520     \csnumgdef{sw@\sw@txt}{\csuse{sw@\sw@txt}+1}%
1521 %

```

Then, write its value to the numbered file.

```

1522   \protected@write\linenum@out{\string\@sw{\sw@txt}{\csuse{sw@\sw@txt}
1523   }}{#1}}%

```

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

```

1524   \else%
1525     \csnumgdef{sw@\sw@txt}{\csuse{sw@\sw@txt}+1}%

```

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

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

```

1526 \protected@write\linenum@outR{}\string\@sw{\sw@txt}{\csuse{sw@\sw@txt
}}{#1}}%
1527 \fi%
1528 %

```

And print the word.

```

1529 #2%
1530 }%
1531 %

```

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

```

\if@addsw32 \newif\if@addsw%
1533 %

```

\@sw The command printed in the auxiliary files.

```

1534 \newcommand{\@sw}[3]{%
1535 \get@sw@txt{#1}%
1536 \unless\ifledRcol%
1537 %

```

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

```

1538 \csxdef{sw@\sw@txt @\the\absline@num @\the\section@num}{#2}%
1539 %

```

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

```

1540 \numdef{\prev@line}{\the\absline@num-1}%
1541 \ifcsundef{sw@\sw@txt @\prev@line @\the\section@num}{%
1542 \csnumgdef{sw@\sw@txt @\prev@line @\the\section@num}{#2-1}%
1543 }{}%
1544 %

```

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

```

1545 \numdef{\the@sw}{#2-\csuse{sw@\sw@txt @\prev@line @\the\section@num}}%
1546 %

```

And do the same thing for the right side.

```

1547 \else%
1548 \csxdef{sw@\sw@txt @\the\absline@numR @\the\section@numR @R}{#2}%
1549 \numdef{\prev@line}{\the\absline@numR-1}%
1550 \ifcsundef{sw@\sw@txt @\prev@line @\the\section@numR @R}{%
1551 \csnumgdef{sw@\sw@txt @\prev@line @\the\section@numR @R}{#2-1}%
1552 }{}%
1553 \numdef{\the@sw}{#2-\csuse{sw@\sw@txt @\prev@line @\the\section@numR @R
}}%
1554 \fi%
1555 %

```

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

```

1556 \@tempcnta=\@edtext@level
1557 \@whilenum{\@tempcnta>0}\do{%
1558   \ifcsdef{sw@list@edtext@tmp@\the\@tempcnta}%
1559     {%
1560       \@addswfalse%
1561       \notbool{lemmacommand@\the\@tempcnta}%
1562       {\@addswtrue}%
1563       {\IfStrEq{#3}{inlemma}%
1564        {\@addswtrue}%
1565        {%
1566          \def\do##1{%
1567            \ifnumequal{##1}{\the\@tempcnta}%
1568              {\@addswtrue\listbreak}%
1569              {}}%
1570          }%
1571          \docsvlist{#3}%
1572        }%
1573      }%
1574      \if@addsw%
1575        \letcs{\@tmp}{sw@list@edtext@tmp@\the\@tempcnta}%
1576        \ifledRcol%
1577          \xright@appenditem{\the@sw}{\the\absline@numR}}\to\@tmp%
1578        \else%
1579          \xright@appenditem{\the@sw}{\the\absline@num}}\to\@tmp%
1580        \fi%
1581        \cslet{sw@list@edtext@tmp@\the\@tempcnta}{\@tmp}%
1582      \fi%
1583    }%
1584    {%
1585      \advance\@tempcnta by -1%
1586    }%
1587  }%
1588 %

```

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

```

1589 \newcommandx{\sameword@inedtext}[2][1,usedefault]{%
1590   \get@sw@txt{#2}%
1591   \unless\ifledRcol%
1592 %

```

Just a precaution.

```

1593   \ifx\sw@list@inedtext\empty%
1594     \def\the@sw{999}%
1595     \def\this@absline{-99}%
1596   \else%
1597 %

```

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

```

1598     \glp\sw@list@inedtext\to\@tmp%
1599     \edef\the@sw{\expandafter\@firstoftwo\@tmp}%
1600     \edef\this@absline{\expandafter\@secondoftwo\@tmp}%
1601     \fi%
1602 %

```

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

```

1603     \ifcsdef{sw@sw@txt @\this@absline @\the\section@num}{%
1604         \numdef{\prev@line}{\this@absline-1}%
1605         \numdef{\sw@atthisline}{\csuse{sw@sw@txt @\this@absline @\the\
section@num}-\csuse{sw@sw@txt @\prev@line @\the\section@num}}%
1606     }%
1607     {\numdef{\sw@atthisline}{0}}%
1608 %

```

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

```

1609     \ifnumgreater{\sw@atthisline}{1}%
1610         {\showwordrank{#2}{\the@sw}}%
1611         {#2}%
1612 %

```

And the same for right side.

```

1613 \else%
1614     \ifx\sw@list@inedtext\empty%
1615         \def\the@sw{999}%
1616         \def\this@absline{-99}%
1617     \else%
1618         \glp\sw@list@inedtext\to\@tmp%
1619         \edef\the@sw{\expandafter\@firstoftwo\@tmp}%
1620         \edef\this@absline{\expandafter\@secondoftwo\@tmp}%
1621         \fi%
1622         \ifcsdef{sw@sw@txt @\this@absline @\the\section@numR @R}{%
1623             \numdef{\prev@line}{\this@absline-1}%
1624             \numdef{\sw@atthisline}{\csuse{sw@sw@txt @\this@absline @\the\
section@numR @R}-\csuse{sw@sw@txt @\prev@line @\the\section@numR @R}}%
1625         }%
1626         {\numdef{\sw@atthisline}{0}}%
1627         \ifnumgreater{\sw@atthisline}{1}%
1628             {\showwordrank{#2}{\the@sw}}%
1629             {#2}%
1630     \fi%
1631 }%
1632 %

```

```

\showwordrank33 % Finally, the way the rank will be printed.
1634 \newcommand{\showwordrank}[2]{%
1635   #1\textsuperscript{#2}%
1636 }%
1637 %

```

VII Paragraph decomposition and reassembly

In order to be able to count the lines of text and affix line numbers, we add an extra stage of processing for each paragraph. We send the paragraph into a box register, rather than straight onto the vertical list, and when the paragraph ends we slice the paragraph into its component lines; to each line we add any notes or line numbers, add a command to write to the line-list, and then at last send the line to the vertical list. This section contains all the code for this processing.

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

Here are numbers and flags that are used internally in the course of the paragraph decomposition.

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

```

1638 \newbox\raw@text
1639 \newif\ifnumberedpar@
1640 \newcount\num@lines
1641 \newbox\one@line
1642 \newcount\par@line
1643 %

```

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

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.

```

1644
1645 \newcommand{\AtEveryPstart}[1]{%
1646   \ifstrempy{#1}%
1647   {\xdef\at@every@pstart{}}%
1648   {\gdef\at@every@pstart{\noindent#1}}%
1649 }%

```

```

1650 \xdef\at@every@pstart{}%
1651
1652 \newcounter{pstart}
1653 \renewcommand{\thepstart}{\bfseries\@arabic\c@pstart}. }
1654 \newif\ifnumberpstart
1655 \numberpstartfalse
1656 \newif\iflabelpstart
1657 \labelpstartfalse
1658 \newcommandx*{\pstart}[1][1]{%
1659   \normal@pars%
1660   \ifstrempy{#1}{\at@every@pstart}{\noindent#1}%
1661   \ifluatex%
1662     \edef\l@luatextextdir@L{\the\textdir}%
1663   \fi%
1664   \@nobraektrue%
1665   \ifnumbering \else%
1666     \led@err@PstartNotNumbered%
1667     \beginnumbering%
1668   \fi%
1669   \ifnumberedpar@%
1670     \led@err@PstartInPstart%
1671   \pend%
1672   \fi%
1673   \list@clear{\inserts@list}%
1674   \global\let\next@insert=\empty%
1675   \begingroup%
1676   \global\advance \l@dnumpstartsL\@ne
1677   \global\setbox\raw@text=\vbox\bgroup%
1678     \ifautopar\else%
1679     \ifnumberpstart%
1680       \ifinstanza\else%
1681       \ifsidepstartnum\else%
1682       \thepstart%
1683     \fi%
1684     \fi%
1685     \fi%
1686     \fi%
1687   \numberedpar@true%
1688   \iflabelpstart\protected@edef\@currentlabel%
1689     {\p@pstart\thepstart}
1690   \fi%
1691   \l@dzeropenalties%
1692   \ignorespaces%because not automatically ignored if an optional argument
    is used (classical TeX behavior for space after commands)
1693 }
1694 %

```

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

```

1695 \newcommandx*{\pend}[1][1]{\ifnumbering \else%

```

```

1696 \led@err@PendNotNumbered%
1697 \fi%
1698 \global\l@dskipversenumberfalse%
1699 \ifnumberedpar@ \else%
1700 \led@err@PendNoPstart%
1701 \fi%
1702 %

```

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.

```

1703 \l@dzeropenalties%
1704 \endgraf\global\num@lines=\prevgraf\egroup%
1705 \global\par@line=0%
1706 %

```

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`.

```

1707 \csnumdef{pstartline}{0}%
1708 \loop\ifvbox\raw@text%
1709 \csnumdef{pstartline}{\pstartline+1}%
1710 \do@line%
1711 \ifbypstart@%
1712 \ifnumequal{\pstartline}{1}{%
1713 \bgroup%
1714 \let\leavevmode\relax%
1715 \setline{1}%
1716 \egroup%
1717 \resetprevline@}{}%
1718 \fi%
1719 \repeat%
1720 %

```

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

```

1721 \flush@notes%
1722 \endgroup%
1723 \ignorespaces%
1724 %

```

Increase `pstart` counter.

```

1725 \ifnumberpstart%
1726 \pstartnumtrue%
1727 \fi%
1728 \addtocounter{pstart}{1}%

```



```
1729 %
```

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

```
1730 \normal@pars%
1731 \ifstrempy{#1}{\at@every@pend}{\noindent#1}%
1732 %
```

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

```
1733 \@nobreakfalse%
1734 \ifautopar%
1735   \autopar%
1736 \fi%
1737 }
1738
1739 %
```

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.

```
\AtEveryPend40
\at@every@pend41 \newcommand{\AtEveryPend}[1]{%
1742   \ifstrempy{#1}%
1743     {\xdef\at@every@pend{}}%
1744     {\gdef\at@every@pend{\noindent#1}}%
1745 }%
1746 \xdef\at@every@pend{}%
1747
1748 %
```

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

```
1749 \newcommand*\l@dzeropenalties{%
1750   \brokenpenalty \z@ \clubpenalty \z@
1751   \displaywidowpenalty \z@ \interlinepenalty \z@ \predisplaypenalty \z@
1752   \postdisplaypenalty \z@ \widowpenalty \z@}
1753
1754 %
```

\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.

```

1755 \newif\ifautopar
1756 \autoparfalse
1757 \newcommand*{\autopar}{
1758   \ifledRcol
1759     \ifnumberingR \else
1760       \led@err@AutoparNotNumbered
1761       \beginnumberingR
1762       \fi
1763     \else
1764       \ifnumbering \else
1765         \led@err@AutoparNotNumbered
1766         \beginnumbering
1767         \fi
1768       \fi
1769     \autopartrue
1770     \everypar{\setbox0=\lastbox
1771               \endgraf \vskip-\parskip
1772               \pstart \noindent \kern\wd0 \ifnumberpstart\ifinstanza\else\thepstart\fi\fi
1773               \let\par=\pend}%
1774   \ignorespaces}
1775 %

```

`\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.

```

1776 \newcommand*{\normal@pars}{\ifautopar\everypar{}\fi\let\par\endgraf}
1777
1778 %

```

`\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.

```
1779 \newif\ifautopar@pause
1780 %
```

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`

```
1781 \newcommand*{\l@dunhbox@line}[1]{\unhbox #1}
1782 \newcommand*{\do@line}{%
1783   {\vbadness=10000
1784     \splittopskip=\z@
1785     \do@linehook
1786     \l@emptyd@ta
1787     \global\setbox\one@line=\vsplit\raw@text to\baselineskip}%
1788     \unvbox\one@line \global\setbox\one@line=\lastbox
1789     \getline@num
1790     \IfStrEq{\led@pb@setting}{before}{\led@check@pb\led@check@nopb}{%}
1791     \ifnum\@lock>\@ne
1792       \inserthangingsymboltrue
1793     \else
1794       \inserthangingsymbolfalse
1795     \fi
1796     \check@pb@in@verse
1797     \ifl@dhiddenumber%
1798       \global\l@dhiddenumberfalse%
1799       \f@x@l@cks%
1800     \else%
1801       \affixline@num%
1802     \fi%
1803   }%
```

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

```
1804 \xifinlist{\the\l@dnumpstartsL}{\eled@sections@}%
1805   {\print@eledsection}%
1806   {\print@line}%
1807   \IfStrEq{\led@pb@setting}{after}{\led@check@pb\led@check@nopb}{%}
1808   }%
1809 %
```

VII.2.2 Process for “normal” line

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

```
1810 \def\print@line{
1811 %
```

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

```
1812 \affixpstart@num%
1813 %
```

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

```
1814 \hb@xt@ \linewidth{%
1815 %
```

User hook.

```
1816 \do@insidelinehook%
1817 %
```

Left line number

```
1818 \l@dld@ta%
1819 %
```

Restore marginal and footnotes.

```
1820 \add@inserts\affixside@note%
1821 %
```

Print left notes.

```
1822 \l@dlsn@te
1823 %
```

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

```
1824 {\ledllfill\hb@xt@ \wd\one@line{\new@line%
1825 %
```

If we use Lua_{TEX} then restore the direction.

```
1826 \ifluatex%
1827 \textdir\l@luatextextdir@L%
1828 \fi%
1829 %
```

Insert, if needed, the hanging symbol.

```
1830 \inserthangingsymbol %Space keep for backward compatibility
1831 %
```

And so, print the line.

```
1832 \l@dunhbox@line{\one@line}}%
1833 %
```

Right line number

```
1834 \ledrlfill\l@drd@ta%
1835 %
```

Print right notes.

```
1836 \l@drsn@te
1837 }}%
1838 %
```

And reinsert penalties (for page breaking)...

```
1839 \add@penalties%
1840 }
1841 %
```

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.

```
1842 \def\print@eledsection{%
1843   \add@inserts\affixside@note%
1844   \numdef{\temp@}{\l@dnumstartL-1}%
1845   \xifinlist{\temp@}{\eled@sections@}{\@nobreaktrue}{\@nobreakfalse}%
1846   \@eled@sectioningtrue%
1847   \csuse{eled@sectioning@the\l@dnumstartL}%
1848   \@eled@sectioningfalse%
1849   \global\csundef{eled@sectioning@the\l@dnumstartL}%
1850   \if@RTL%
1851     \hspace{-3\paperwidth}%
1852     {\hbox{\l@dunhbox@line{one@line}} \new@line}%
1853   \else%
1854     \hspace{3\paperwidth}%
1855     {\new@line \hbox{\l@dunhbox@line{one@line}}}%
1856   \fi%
1857   \vskip-\baselineskip%
1858 }
1859 %
```

VII.2.4 Hooks

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

```
1860 \newcommand*{\do@linehook}{}
1861 \newcommand*{\do@insidelinehook}{}
1862 %
```

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

```

1863 \newcommand*\dolinehook}[1]{\gdef\do@linehook{#1}}%
1864 \newcommand*\doinsidelinehook}[1]{\gdef\do@insidelinehook{#1}}%
1865
1866 %

```

VII.2.5 Sidenotes and marginal line number initialization

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

```

1867 \l@dcsnotetext \newcommand*\l@emptyd@ta{%
1868 \l@dcsnotetext@l \gdef\l@dld@ta{%
1869 \l@dcsnotetext@r \gdef\l@drd@ta{%
1870 \gdef\l@dcsnotetext@l{}%
1871 \gdef\l@dcsnotetext@r{}%
1872 \gdef\l@dcsnotetext{}%
1873
1874 %

```

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

```

1875 \l@drsn@te \newcommand*\l@dlsn@te{%
1876 \hb@xt@ \z@{\hss\box\l@dlp@rbox\kern\ledlsnotesep}}
1877 \newcommand*\l@drsn@te{%
1878 \hb@xt@ \z@{\kern\ledrsnotesep\box\l@drp@rbox\hss}}
1879
1880 %

```

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

```

1881 \newcommand*\ledllfill{\hfil}
1882 \newcommand*\ledrlfill{}
1883
1884 %

```

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.

```

1885 \newcommand*\getline@num{%
1886 \global\advance\absline@num \@ne%
1887 \do@actions
1888 \do@ballast
1889 \ifnumberline
1890 \ifsublines@

```

```

1891     \ifnum\sub@lock<\tw@
1892         \global\advance\subline@num \@ne
1893     \fi
1894 \else
1895     \ifnum\@lock<\tw@
1896         \global\advance\line@num \@ne
1897         \global\subline@num \z@
1898     \fi
1899 \fi
1900 \fi
1901 }
1902 %

```

\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. 144).

\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.

```

1903 \newcount\ballast@count
1904 \newcounter{ballast}
1905 \setcounter{ballast}{0}
1906 %

```

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.

```

1907 \newcommand*{\do@ballast}{\global\ballast@count \z@
1908 \begingroup
1909     \advance\absline@num \@ne
1910     \ifnum\next@actionline=\absline@num
1911         \ifnum\next@action>-1001\relax
1912         \global\advance\ballast@count by -\c@ballast
1913     \fi
1914 \fi
1915 \endgroup}
1916 %

```

\do@actions The `\do@actions` macro looks at the list of actions to take at particular absolute line
\do@actions@next 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`.

```

1917 \newcommand*{\do@actions}{%
1918   \global\let\do@actions@next=\relax
1919   \ifnum\absline@num<\next@actionline\else
1920 %

```

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.

```

1921   \ifnum\next@action>-1001
1922     \global\page@num=\next@action
1923     \ifbypage@
1924       \global\line@num=\z@ \global\subline@num=\z@
1925       \resetprevline@
1926     \fi
1927 %

```

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`.)

```

1928   \else
1929     \ifnum\next@action<-4999
1930       \@l@dttempcnta=-\next@action
1931       \advance\@l@dttempcnta by -5001
1932       \ifsublines@
1933         \global\subline@num=\@l@dttempcnta
1934       \else
1935         \global\line@num=\@l@dttempcnta
1936       \fi
1937 %

```

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

```

1938   \else
1939     \@l@dttempcnta=-\next@action
1940     \advance\@l@dttempcnta by -1000
1941     \do@actions@fixedcode
1942   \fi
1943 \fi
1944 %

```

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

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

```

1945   \ifx\actionlines@list\empty
1946     \gdef\next@actionline{1000000}%
1947   \else
1948     \gl@p\actionlines@list\to\next@actionline
1949     \gl@p\actions@list\to\next@action
1950     \global\let\do@actions@next=\do@actions
1951   \fi

```



```

1952 \fi
1953 %
      Make the recursive call, if necessary.
1954 \do@actions@next}
1955 %
1956 %

```

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

```

1957 \newcommand*{\do@actions@fixedcode}{%
1958 \ifcase\@l@dttempcnta
1959 \or% % 1001
1960 \global\sublines@true
1961 \or% % 1002
1962 \global\sublines@false
1963 \or% % 1003
1964 \global\@lock=\@ne
1965 \or% % 1004
1966 \ifnum\@lock=\tw@
1967 \global\@lock=\thr@@
1968 \else
1969 \global\@lock=\z@
1970 \fi
1971 \or% % 1005
1972 \global\sub@lock=\@ne
1973 \or% % 1006
1974 \ifnum\sub@lock=\tw@
1975 \global\sub@lock=\thr@@
1976 \else
1977 \global\sub@lock=\z@
1978 \fi
1979 \or% % 1007
1980 \l@dskipnumbertrue
1981 \or% % 1008
1982 \l@dskipversenumbertrue%
1983 \or% % 1009
1984 \l@dhiddenumbertrue
1985 \else
1986 \led@warn@BadAction
1987 \fi}
1988 %
1989 %
1990 %

```

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.

```
1991 \newcommand*{\affixline@num}{%
1992 %
```

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).

```
1993 \ifl@groupnotesL\else
1994 \ifnumberline
1995 \ifl@dskipnumber
1996 \global\l@dskipnumberfalse
1997 \else
1998 \ifsublines@
1999 \l@tempcntb=\subline@num
2000 \ifnum\subline@num>\c@firstsublinenum
2001 \l@tempcnta=\subline@num
2002 \advance\l@tempcnta by-\c@firstsublinenum
2003 \divide\l@tempcnta by\c@sublinenumincrement
2004 \multiply\l@tempcnta by\c@sublinenumincrement
2005 \advance\l@tempcnta by\c@firstsublinenum
2006 \else
2007 \l@tempcnta=\c@firstsublinenum
2008 \fi
2009 %
```

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.

```
2010 \ch@cksub@l@ck
2011 %
```

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

```
2012 \else
2013 \l@tempcntb=\line@num
2014 %
```

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

```

2015     \ifx\linenumberlist\empty
2016       \ifnum\line@num>\c@firstlinenum
2017         \@l@tempcnta=\line@num
2018         \advance\@l@tempcnta by-\c@firstlinenum
2019         \divide\@l@tempcnta by\c@linenumincrement
2020         \multiply\@l@tempcnta by\c@linenumincrement
2021         \advance\@l@tempcnta by\c@firstlinenum
2022       \else
2023         \@l@tempcnta=\c@firstlinenum
2024     \fi
2025 \else
2026 %

```

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

```

2027     \@l@tempcnta=\line@num
2028     \edef\rem@inder{\,\linenumberlist,\number\line@num,}%
2029     \edef\sc@n@list{\def\noexpand\sc@n@list
2030       ###1,\number\@l@tempcnta,###2|{\def\noexpand\rem@inder
2031 {####2}}}%
2032     \sc@n@list\expandafter\sc@n@list\rem@inder|
2033     \ifx\rem@inder\empty%
2034       \advance\@l@tempcnta\@ne
2035     \fi
2036 %

```

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

```

2037     \ch@ck@l@ck
2038 \fi
2039 %

```

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

```

2040     \ifnum\@l@tempcnta=\@l@tempcntb
2041     \ifl@dskipversenumber\else
2042 %

```

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`.

```

2043 \l@drd@ta
2044 \if@twocolumn
2045 \if@firstcolumn
2046 \gdef\l@dld@ta{\llap{\leftlinenum}}}%
2047 \else
2048 \gdef\l@drd@ta{\rlap{\rightlinenum}}}%
2049 \fi
2050 \else
2051 \@l@tempcntb=\line@margin
2052 \ifnum\@l@tempcntb>\@ne
2053 \advance\@l@tempcntb \page@num
2054 \fi
2055 \ifodd\@l@tempcntb
2056 \gdef\l@drd@ta{\rlap{\rightlinenum}}}%
2057 \else
2058 \gdef\l@dld@ta{\llap{\leftlinenum}}}%
2059 \fi
2060 \fi
2061 \fi
2062 %

```

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

```

2063 \f@x@l@cks
2064 \fi
2065 \fi
2066 \fi
2067 }
2068 %
2069 %

```

`\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.

```

2070 \newcommand*\ch@cksub@l@ck{%
2071 \ifcase\sub@lock
2072 \or
2073 \ifnum\sublock@disp=\@ne
2074 \@l@tempcntb=\z@ \@l@tempcnta=\@ne
2075 \fi
2076 \or
2077 \ifnum\sublock@disp=\tw@ \else
2078 \@l@tempcntb=\z@ \@l@tempcnta=\@ne
2079 \fi
2080 \or
2081 \ifnum\sublock@disp=\z@

```

```

2082         \@l@dttempcntb=\z@ \@l@dttempcnta=\@ne
2083         \fi
2084     \fi}
2085 %

```

Similarly for line numbers.

```

2086 \newcommand*{\ch@ck@l@ck}{%
2087     \ifcase\@lock
2088     \or
2089         \ifnum\lock@disp=\@ne
2090             \@l@dttempcntb=\z@ \@l@dttempcnta=\@ne
2091             \fi
2092     \or
2093         \ifnum\lock@disp=\tw@ \else
2094             \@l@dttempcntb=\z@ \@l@dttempcnta=\@ne
2095             \fi
2096     \or
2097         \ifnum\lock@disp=\z@
2098             \@l@dttempcntb=\z@ \@l@dttempcnta=\@ne
2099             \fi
2100     \fi}
2101 %

```

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

```

2102 \newcommand*{\f@x@l@cks}{%
2103     \ifcase\@lock
2104     \or
2105         \global\@lock=\tw@
2106     \or \or
2107         \global\@lock=\z@
2108     \fi
2109     \ifcase\sub@lock
2110     \or
2111         \global\sub@lock=\tw@
2112     \or \or
2113         \global\sub@lock=\z@
2114     \fi}
2115 %
2116 %

```

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:

`\affixpstart@num`
`\pstartnum`

- The pstarts counter is upgrade in the `\pend` command. Consequently, the `\affixpstart@num` command has not to upgrade it, unlike the `\affixline@num` which upgrades the lines counter.

- To print the pstart number only at the beginning of a pstart, and not in every line, a boolean test is made. The `\pstartnum` boolean is set to TRUE at every `\pend`. It is tried in the `\leftpstartnum` and `\rightpstartnum` commands. After the try, it is set to FALSE.

```

\leftpstartnum17
\rightpstartnum18 \newif\ifsidepstartnum
\ifsidepstartnum19 \newcommand*{\affixpstart@num}{%
2120   \ifsidepstartnum
2121     \if@twocolumn
2122       \if@firstcolumn
2123         \gdef\l@dld@ta{\llap{\leftpstartnum}}}%
2124       \else
2125         \gdef\l@drd@ta{\rlap{\rightpstartnum}}}%
2126       \fi
2127     \else
2128       \l@dttempcntb=\line@margin
2129       \ifnum\l@dttempcntb>\@ne
2130         \advance\l@dttempcntb \page@num
2131       \fi
2132       \ifodd\l@dttempcntb
2133         \gdef\l@drd@ta{\rlap{\rightpstartnum}}}%
2134       \else
2135         \gdef\l@dld@ta{\llap{\leftpstartnum}}}%
2136       \fi
2137     \fi
2138   \fi
2139 }
2140 %
2141
2142 \newif\ifpstartnum
2143 \pstartnumtrue
2144 \newcommand*{\leftpstartnum}{
2145   \ifpstartnum\thepstart
2146   \kern\linenumsep\fi
2147   \global\pstartnumfalse
2148 }
2149
2150 \newcommand*{\rightpstartnum}{
2151   \ifpstartnum
2152   \kern\linenumsep
2153   \thepstart
2154   \fi
2155   \global\pstartnumfalse
2156 }
2157 %

```

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.

```
2158 \list@create{\inserts@list}
2159 %
```

`\add@inserts` `\add@inserts` is the penultimate macro used by `\do@line`; it takes insertions saved in a list macro and sends them onto the vertical list.

It may call itself recursively, and to do this efficiently (using \TeX 's optimization for tail recursion), we define a control-sequence called `\add@inserts@next` that is always the last thing that `\add@inserts` does. If there could be more inserts to process for this line, `\add@inserts@next` is set equal to `\add@inserts`; otherwise it is just `\relax`.

```
2160 \newcommand*{\add@inserts}{%
2161   \global\let\add@inserts@next=\relax
2162   %
```

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

```
2163   \ifx\inserts@list\empty \else
2164   %
```

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.

```
2165   \ifx\next@insert\empty
2166     \ifx\insertlines@list\empty
2167       \global\noteschanged@true
2168       \gdef\next@insert{100000}%
2169     \else
2170       \gl@p\insertlines@list\to\next@insert
2171     \fi
2172   \fi
2173   %
```

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.

```

2174 \ifnum\next@insert=\absline@num
2175   \gl@p\inserts@list\to\@insert
2176   \@insert
2177   \global\let\@insert=\undefined
2178   \global\let\next@insert=\empty
2179   \global\let\add@inserts@next=\add@inserts
2180 \fi
2181 \fi
2182 %

```

Make the recursive call, if necessary.

```

2183 \add@inserts@next}
2184
2185 %

```

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. 135). Finally, the penalty is checked to see that it does not go below -10000 .

```

2186 \newcommand*{\add@penalties}{\@l@tempcnta=\ballast@count
2187   \ifnum\num@lines>\@ne
2188     \global\advance\par@line \@ne
2189     \ifnum\par@line=\@ne
2190       \advance\@l@tempcnta \clubpenalty
2191     \fi
2192     \@l@tempcntb=\par@line \advance\@l@tempcntb \@ne
2193     \ifnum\@l@tempcntb=\num@lines
2194       \advance\@l@tempcnta \widowpenalty
2195     \fi
2196     \ifnum\par@line<\num@lines
2197       \advance\@l@tempcnta \interlinepenalty
2198     \fi
2199   \fi
2200   \ifnum\@l@tempcnta=\z@
2201     \relax
2202   \else

```



```

2203 \ifnum\@l@dttempcnta>-10000
2204 \penalty\@l@dttempcnta
2205 \else
2206 \penalty -10000
2207 \fi
2208 \fi}
2209
2210 %

```

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.

```

2211 \newcommand*{\flush@notes}{%
2212 \xloop
2213 \ifx\inserts@list\empty \else
2214 \gl@p\inserts@list\to\@insert
2215 \insert
2216 \global\let\@insert=\undefined
2217 \repeat}
2218
2219 %

```

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

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

```

2220 \def\xloop#1\repeat{%
2221 \def\body{#1\expandafter\body\fi}%
2222 \body}
2223
2224 %

```

XII Critical footnotes

The footnote macros are adapted from those in PLAIN T_EX, but they differ in these respects: the outer-level commands must add other commands to a list macro rather than doing insertions immediately; there are many separate levels of the footnotes, not just one; and there are options to reformat footnotes into paragraphs or into multiple columns.

XII.1 Fonts

Before getting into the details of formatting the notes, we set up some font macros. It is the notes that present the greatest challenge for our font-handling mechanism, because we need to be able to take fragments of our main text and print them in different forms: it is common to reduce the size, for example, without otherwise changing the fonts used.

`\select@lemmafont` `\select@lemmafont` is provided to set the right font for the lemma in a note. This macro extracts the font specifier from the line and page number cluster, and issues the associated font-changing command, so that the lemma is printed in its original font.

```

2225 \def\select@lemmafont#1|#2|#3|#4|#5|#6|#7|{\select@@lemmafont#7|}
2226 \def\select@@lemmafont#1/#2/#3/#4|{%
2227     {\fontencoding{#1}\fontfamily{#2}\fontseries{#3}\fontshape{#4}%
2228     \selectfont}
2229
2230 %

```

XII.2 Individual note options

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

```

2231 \newcommand*\footnoteoptions@[3]{%
2232     \def\do##1{%
2233         \ifstrequal{#1}{L}{% In Leftside
2234             \xright@appenditem{\noexpand\setkeys[mac]{#3footnoteoption}{\
2235             unexpanded{##1}}}{\to\inserts@list%
2236             \global\advance\insert@count \@ne% Increment the left insert
2237             counter.
2238             }%
2239             \xright@appenditem{\noexpand\setkeys[mac]{#3footnoteoption}{\
2240             unexpanded{##1}}}{\to\inserts@listR%
2241             \global\advance\insert@countR \@ne% Increment the right insert
2242             counter insert.
2243             }%
2244             \notblank{#2}{\docsvlist{#2}}{}}% Parsing all options
2245     }
2246 %

```

XII.3 Notes language

`\footnotelang@lua` `\footnotelang@lua` is called to remember the information about the direction of a lemma when Lua¹TeX is used.

```

2245 \newcommand*{\footnotelang@lua}[1][1=L,usedefault]{%
2246   \ifstrequal{#1}{L}{%
2247     \xright@appenditem{\csxdef{footnote@luatextextdir}{\the\textdir}}{\to\
inserts@list%Know the dir of lemma
2248     \global\advance\insert@count \@ne%
2249     \xright@appenditem{\csxdef{footnote@luatexpardir}{\the\pardir}}{\to\
inserts@list%Know the dir of lemma
2250     \global\advance\insert@count \@ne%
2251   }%
2252   {%
2253     \xright@appenditem{\csxdef{footnote@luatextextdir}{\the\textdir}}{\to\
inserts@listR%Know the dir of lemma
2254     \global\advance\insert@countR \@ne%
2255     \xright@appenditem{\csxdef{footnote@luatexpardir}{\the\pardir}}{\to\
inserts@listR%Know the dir of lemma
2256     \global\advance\insert@countR \@ne%
2257   }%
2258 }
2259 %

```

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

```

2260 \newcommand*{\footnotelang@poly}[1][1=L,usedefault]{%
2261   \ifstrequal{#1}{L}{%
2262     \if@RTL%
2263       \xright@appenditem{\csxdef{footnote@dir}{@RTLtrue}}{\to\
inserts@list%Know the language used in the lemma
2264       \global\advance\insert@count \@ne%
2265     }%
2266     \else
2267       \xright@appenditem{\csxdef{footnote@dir}{@RTLfalse}}{\to\
inserts@list%Know the language of lemma
2268       \global\advance\insert@count \@ne%
2269     }%
2270     \fi%
2271     \xright@appenditem{\csxdef{footnote@lang}{\expandonce\language}}{\to\
inserts@list%Know the language of lemma
2272     \global\advance\insert@count \@ne%
2273   }%
2274   {%
2275     \if@RTL
2276       \xright@appenditem{\csxdef{footnote@dir}{@RTLtrue}}{\to\
inserts@listR%Know the language of lemma
2277       \global\advance\insert@countR \@ne%
2278     }%
2279     \else
2280       \xright@appenditem{\csxdef{footnote@dir}{@RTLfalse}}{\to\
inserts@listR%Know the language of lemma

```

```

2278 \global\advance\insert@countR \@ne%
2279 \fi
2280 \xright@appenditem{{\csxdef{footnote@lang}{\expandonce\language\language}}}{\
to\inserts@listR%Know the language of lemma
2281 \global\advance\insert@countR \@ne%
2282 }%
2283 }
2284 %

```

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 \LaTeX `\floatingpenalty`.

```

2285 \newcommand*{\footsplitskips}{%
2286 \interlinepenalty=\interfootnotelinepenalty
2287 \unless\ifl@dprintingpages%
2288 \floatingpenalty=\@MM%
2289 \fi%

```

```

2290 \splittopskip=\ht\strutbox \splitmaxdepth=\dp\strutbox
2291 \leftskip=\z@skip \rightskip=\z@skip
2292
2293 %

```

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

```

2294 \let\normalfootnoterule=\footnoterule
2295 %

```

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.

```

2296 \newcommand{\Xarrangement}[2][1,usedefault]{%
2297   \def\do##1{%
2298     \csname Xarrangement@#2\endcsname{##1}%
2299   }%
2300   \ifstrempy{#1}%
2301   {%
2302     \dolistloop{\@series}%
2303   }%
2304   {
2305     \docsvlist{#1}%
2306   }%
2307 }%
2308 %

```

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\vAfootnote=\normalvfootnote \let\Afootfmt=\normalfootfmt
\let\Afootstart=\normalfootstart \let\Afootgroup=\normalfootgroup
\let\Afootnoterule=\normalfootnoterule

```

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

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

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

```

2309 \newcommand*{\Xarrangement@normal}[1]{%
2310   \csgdef{series@display#1}{normal}
2311   \expandafter\let\csname #1footstart\endcsname=\normalfootstart
2312   \expandafter\let\csname v#1footnote\endcsname=\normalvfootnote
2313   \expandafter\let\csname #1footfmt\endcsname=\normalfootfmt
2314   \expandafter\let\csname #1footgroup\endcsname=\normalfootgroup
2315   \expandafter\let\csname #1footnoterule\endcsname=%
2316                                     \normalfootnoterule
2317   \count\csname #1footins\endcsname=1000
2318   \dimen\csname #1footins\endcsname=\csuse{Xmaxhnotes@#1}
2319   \skip\csname #1footins\endcsname=\csuse{Xbeforenotes@#1}%
2320   \advance\skip\csname #1footins\endcsname by\csuse{Xafterrule@#1}%
2321   %

```

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.

```

2322   \csxdef{default@#1footins}{1000}%Use this to confine the notes to one
2323   side only
2324   %

```

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).

```

2324   \ifnoledgroup@else%
2325     \expandafter\let\csname mpv#1footnote\endcsname=\mpnormalvfootnote
2326     \expandafter\let\csname mp#1footgroup\endcsname=\mpnormalfootgroup
2327     \count\csname mp#1footins\endcsname=1000
2328     \dimen\csname mp#1footins\endcsname=\csuse{Xmaxhnotes@#1}
2329     \skip\csname mp#1footins\endcsname=\csuse{Xbeforenotes@#1}%
2330     \advance\skip\csname mp#1footins\endcsname by\csuse{Xafterrule@#1}%
2331   \fi
2332 }
2333
2334 %

```

`\normalvfootnote` We now begin a series of commands that do 'normal' footnote formatting: a format much like that implemented in PLAIN \TeX , 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.

```

2335 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\normalvfootnote}[2]{%
2336   \csuse{Xbeforeinserting@#1}%
2337   \insert\csname #1footins\endcsname\bgroup
2338   \hsize=\expandafter\dimexpr\csuse{Xwidth@#1}\relax%
2339   \noindent\csuse{Xbhooknote@#1}%
2340   \csuse{Xnotefontsize@#1}%
2341   \footsplitskips
2342   \ifl@dpairing\ifl@dpaging\else%
2343     \setXnoteswidthliketwocolumns@{#1}%
2344   \fi\fi%
2345   \setXnotespositionliketwocolumns@{#1}%
2346   \spaceskip=\z@skip \xspaceskip=\z@skip
2347   \csname #1footfmt\endcsname #2{#1}\egroup}
2348 %

```

\mpnormalvfootnote And a somewhat different version for minipages.

```

2349 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\mpnormalvfootnote}[2]{%
2350   \global\setbox\@nameuse{mp#1footins}\vbox{%
2351     \unvbox\@nameuse{mp#1footins}
2352     \noindent\csuse{Xbhooknote@#1}%
2353     \csuse{Xnotefontsize@#1}%
2354     \hsize\columnwidth
2355     \@parboxrestore
2356     \color@begingroup
2357     \csname #1footfmt\endcsname #2{#1}\color@endgroup}}
2358 %
2359 %

```

\normalfootfmt `\normalfootfmt` is a ‘normal’ macro to take the footnote line and page number information (see V.9 p. 89), 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.

```

2360 %
2361 %
2362 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\normalfootfmt}[4]{%
2363   \Xledsetnormalparstuff{#4}%
2364   \hangindent=\csuse{Xhangindent@#4}
2365   \everypar{\hangindent=\csuse{Xhangindent@#4}}%
2366   \strut{\printlinefootnote{#1}{#4}}%
2367   \print@lemma{#1}{#2}{#4}%
2368   \csuse{Xwrapcontent@#4}{#3}%
2369   \strut\par}
2370 %

```

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

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

But if the skip `\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`.

```
2371 \newcommand*{\normalfootstart}[1]{%
2372 %
```

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

```
2373 \ifdimequal{0pt}{\preXnotes@}{}%
2374 {%
2375 \iftoggle{preXnotes@}{%
2376 \togglefalse{preXnotes@}%
2377 \skip\csname #1footins\endcsname=%
2378 \dimexpr\csuse{preXnotes@}+\csuse{Xaftererrule@#1}\relax%
2379 }%
2380 }%
2381 }%
2382 \vskip\skip\csname #1footins\endcsname%
2383 %
```

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. 197 for the general description of the problem.

```
2384 \leftskip0pt \rightskip0pt
2385 \ifl@dpairing\else%
2386 \hsize=\old@hsize%
2387 \fi%
2388 \setXnoteswidthliketwocolumns@{#1}%
2389 \setXnotespositionliketwocolumns@{#1}%
2390 %
```

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


```

2391 \print@Xfootnoterule{#1}%
2392 \noindent\leavevmode}
2393 %

```

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

```

2394 \newcommand*{\normalfootgroup}[1]{%
2395   {\csuse{Xnotefontsize@#1}\noindent\csuse{Xtxtbeforenotes@#1}}%
2396   \csuse{Xbhookgroup@#1}%
2397   \unvbox\csname #1footins\endcsname%
2398   \hspace=\old@hspace%
2399   }%
2400
2401 %

```

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

```

2402 \unless\ifnoledgroup@
2403 \newcommand*{\mpnormalfootgroup}[1]{%
2404   \vskip\skip\@nameuse{mp#1footins}
2405   \ifl@dpairing\ifparledgroup%
2406     \leavevmode\marks\parledgroup@{begin}%
2407     \marks\parledgroup@series{#1}%
2408     \marks\parledgroup@type{Xfootnote}%
2409     \fi\fi\normalcolor%
2410     \ifparledgroup%
2411       \ifl@dpairing%
2412       \else%
2413         \setXnoteswidthliketwocolumns@{#1}%
2414         \setXnotespositionliketwocolumns@{#1}%
2415         \print@Xfootnoterule{#1}%%
2416       \fi%
2417     \else%
2418       \setXnoteswidthliketwocolumns@{#1}%
2419       \setXnotespositionliketwocolumns@{#1}%
2420       \print@Xfootnoterule{#1}%%
2421     \fi%
2422     \setlength{\parindent}{0pt}
2423     {\csuse{Xnotefontsize@#1}\csuse{Xtxtbeforenotes@#1}}
2424     \csuse{Xbhookgroup@#1}%
2425     \unvbox\csname mp#1footins\endcsname}}
2426 \fi
2427 %

```

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.

```

2428 \newcommand*{\Xarrangement@paragraph}[1]{%
2429   \csgdef{series@display#1}{paragraph}
2430   \expandafter\newcount\csname #1prevpage@num\endcsname
2431   \expandafter\let\csname #1footstart\endcsname=\parafootstart
2432   \expandafter\let\csname v#1footnote\endcsname=\paravfootnote
2433   \expandafter\let\csname #1footfmt\endcsname=\parafootfmt
2434   \expandafter\let\csname #1footgroup\endcsname=\parafootgroup
2435   \count\csname #1footins\endcsname=1000
2436   \csxdef{default@#1footins}{1000}%Use this to confine the notes to one
side only
2437   \dimen\csname #1footins\endcsname=\csuse{Xmaxhnotes@#1}
2438   \skip\csname #1footins\endcsname=\csuse{Xbeforenotes@#1}%
2439   \advance\skip\csname #1footins\endcsname by\csuse{Xafterrule@#1}%
2440   \para@footsetup{#1}
2441   %

```

And the extra setup for minipages.

```

2442   \ifnoledgroup@else
2443     \expandafter\let\csname mpv#1footnote\endcsname=\mpparavfootnote
2444     \expandafter\let\csname mp#1footgroup\endcsname=\mpparafootgroup
2445     \count\csname mp#1footins\endcsname=1000
2446     \dimen\csname mp#1footins\endcsname=\csuse{Xmaxhnotes@#1}
2447     \skip\csname mp#1footins\endcsname=\csuse{Xbeforenotes@#1}%
2448     \advance\skip\csname mp#1footins\endcsname by\csuse{Xafterrule@#1}%
2449   \fi
2450 }
2451 %

```

`\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.

```

2452 \providecommand{\footfudgefiddle}{64}
2453 %

```

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

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

```

2454 \newcommand*{\para@footsetup}[1]{\csuse{Xbhookgroup@#1}\csuse{
Xnotefontsize@#1}
2455 \setXnoteswidthliketwocolumns@{#1}%
2456 \ifcempty{Xwidth@#1}%
2457 {}%
2458 {\columnwidth=\expandafter\dimexpr\csuse{Xwidth@#1}\relax}%
2459 \dimen0=\baselineskip
2460 \multiply\dimen0 by 1024
2461 \divide \dimen0 by \columnwidth \multiply\dimen0 by \footfudgefiddle\
relax
2462 \csxdef{#1footfudgefactor}{%
2463 \expandafter\strip@pt\dimen0 }}
2464
2465 %

```

`\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.

```

2466 \newcommand*{\parafootstart}[1]{%
2467 \rightskip=0pt \leftskip=0pt%
2468 \nottoggle{Xparindent@#1}{\parindent=\z@}{}%
2469 \ifdimequal{0pt}{\preXnotes@}{}%
2470 {%
2471 \iftoggle{preXnotes@}{%
2472 \togglefalse{preXnotes@}%
2473 \skip\csname #1footins\endcsname=%
2474 \dimexpr\csuse{preXnotes@}+\csuse{Xafterrule@#1}\relax%
2475 }%
2476 }%
2477 }%
2478 \vskip\skip\csname #1footins\endcsname%
2479 \setXnoteswidthliketwocolumns@{#1}%
2480 \setXnotespositionliketwocolumns@{#1}%
2481 \print@Xfootnoterule{#1}%
2482 \let\@bidiRTL@everypar\@empty%
2483 \noindent\leavevmode}
2484 %

```

`\paravfootnote` `\paravfootnote` is a version of the `\vfootnote` command that is used for paragraphed notes. It gets appended to the `\inserts@list` list by an outer-level footnote command like `\Afootnote`. The first argument is the note series letter; the second is the full text of the printed note itself, including line numbers, lemmata, and footnote text.

The initial model for this insertion is, of course, the `\insert\footins` definition in *The TeXbook*, p. 398. There, the footnotes are first collected up in `hboxes`, and these `hboxes` are later unpacked and stuck together into a paragraph.

However, Michael Downes has pointed out that because text in `hboxes` gets typeset in restricted horizontal mode, there are some undesirable side-effects if you later want to break such text across lines. In restricted horizontal mode, where \TeX does not expect to have to break lines, it does not insert certain items like `\discretionary`s. If you later unbox these `hboxes` and stick them together, as the *TeXbook* macros do to make these footnotes, you lose the ability to hyphenate after an explicit hyphen. This can lead to overfull `hboxes` when you would not expect to find them, and to the uninitiated it might be very hard to see why the problem had arisen.²⁷

Wayne Sullivan pointed out to us another subtle problem that arises from the same cause: \TeX also leaves the `\language` whatsit nodes out of the horizontal list.²⁸ So changes from one language to another will not invoke the proper hyphenation rules in such footnotes. Since critical editions often do deal with several languages, especially in a footnotes, we really ought to get this bit of code right.

To get around these problems, Wayne suggested emendations to the *TeXbook* versions of these macros which are broadly the same as those described by Michael: the central idea (also suggested by Donald Knuth in a letter to Michael) is to avoid collecting the text in an `hbox` in the first place, but instead to collect it in a `vbox` whose width is (virtually) infinite. The text is therefore typeset in unrestricted horizontal mode, as a paragraph consisting of a single long line. Later, there is an extra level of unboxing to be done: we have to unpack the `vbox`, as well as the `hboxes` inside it, but that is not too hard. For details, we refer you to Michael's article, where the issues are clearly explained.²⁹ Michael's unboxing macro is called `\Xunvxh`: `unvbox`, extract the last line, and `unhbox` it.

Doing things this way has an important consequence: as Michael pointed out, you really can't put an explicit line-break into a note built in a `vbox` the way we are doing.³⁰ In other words, be very careful not to use `\break`, or `\penalty-10000`, or any equivalent inside your para-footnote. If you do, most of the note will probably disappear. You *are* allowed to make strong suggestions; in fact `\penalty-9999` will be quite okay. Just do not make the break mandatory. We have not applied any of Michael's solutions 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. 152 above). We need to do this, since `\footfudgefactor` is calculated on the assumption that the

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

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

²⁹Wayne supplied his own macros to do this, but since they were almost identical to Michael's, Peter Wilson have used the latter's `\Xunvxh` macro since it is publicly documented.

³⁰'Line Breaking', p. 610.

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.

```

2485 \newcommand*{\paravfootnote}[2]{%
2486   \csuse{Xbeforeinserting@#1}%
2487   \insert\csname #1footins\endcsname
2488   \bgroup
2489     \csuse{Xnotefontsize@#1}
2490     \footsplitskips
2491     \setbox0=\vbox{\hsize=\maxdimen%
2492       \let\bid@RTL@everypar\@empty%
2493       \noindent\csuse{Xhooknote@#1}%
2494       \csname #1footfmt\endcsname #2{#1}}%
2495     \setbox0=\hbox{\Xunvxh{0}{#1}}%
2496     \dp0=0pt
2497     \ht0=\csname #1footfudgefactor\endcsname\wd0
2498   %

```

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

```

2499   \if@RTL\noindent \leavevmode\fi\box0%
2500   \penalty0
2501   \egroup}
2502 %
2503 %

```

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

`\mpparavfootnote` This version is for minipages.

```

2504 \newcommand*{\mpparavfootnote}[2]{%
2505   \global\setbox\@nameuse{mp#1footins}\vbox{%
2506     \unvbox\@nameuse{mp#1footins}%
2507     \csuse{Xnotefontsize@#1}
2508     \footsplitskips
2509     \setbox0=\vbox{\hsize=\maxdimen%
2510       \let\bid@RTL@everypar\@empty%
2511       \noindent\color@begingroup%
2512       \csuse{Xhooknote@#1}%
2513       \csname #1footfmt\endcsname #2{#1}\color@endgroup}%

```

```

2514 \setbox0=\hbox{\Xunvxh{0}{#1}}%
2515 \dp0=\z@
2516 \ht0=\csname #1footfudgefactor\endcsname\wd0
2517 \box0
2518 \penalty0
2519 }}
2520
2521 %

```

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

```

2522 \newcommand*{\Xunvxh}[2]{%
2523   \setbox0=\vbox{\unvbox#1%
2524     \global\setbox1=\lastbox}%
2525   \unhbox1
2526   \unskip           % remove \rightskip,
2527   \unskip           % remove \parfillskip,
2528   \unpenalty        % remove \penalty of 10000,
2529   \hskip\csuse{Xafternote@#2}\relax}% but add the glue to go between the
notes
2530
2531 %

```

\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).

```

2532 \newcommand*{\parafootfmt}[4]{%
2533   \Xinsertparafootsep{#4}%
2534   \ledsetnormalparstuff@common%
2535   \printlinefootnote{#1}{#4}%
2536   \print@lemma{#1}{#2}{#4}%
2537   \csuse{Xwrapcontent@#4}{#3}%
2538   \penalty-10 }
2539 %

```

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.

```

2540 \newcommand*{\parafootgroup}[1]{%
2541   \hsize=\expandafter\dimexpr\csuse{Xwidth@#1}\relax%
2542   \unvbox\csname #1footins\endcsname
2543   \ifcsstring{Xragged@#1}{L}{\RaggedLeft}{}%
2544   \ifcsstring{Xragged@#1}{R}{\RaggedRight}{}%
2545   \makehboxofhboxes
2546   \setbox0=\hbox{\csuse{Xnotefontsize@#1}\csuse{Xtxtbeforenotes@#1}}\
unhbox0 \removehboxes}%
2547   \csuse{Xbhookgroup@#1}%
2548   \csuse{Xnotefontsize@#1}%
2549   \unhbox0\par%
2550   \global\hsize=\old@hsize%
2551 }%
2552
2553 %

```

`\mpparafootgroup` The minipage version.

```

2554 \newcommand*{\mpparafootgroup}[1]{%
2555   \setXnoteswidthliketwocolumns@{#1}%
2556   \vskip\skip\@nameuse{mp#1footins}
2557   \ifl@dpairing\ifparledgroup%
2558     \leavevmode\marks\parledgroup@{begin}%
2559     \marks\parledgroup@series{#1}%
2560     \marks\parledgroup@type{Xfootnote}%
2561   \fi\fi\normalcolor
2562   \ifparledgroup%
2563     \ifl@dpairing%
2564     \else%
2565       \setXnoteswidthliketwocolumns@{#1}%
2566       \setXnotespositionliketwocolumns@{#1}%
2567       \print@Xfootnoterule{#1}%%
2568     \fi%
2569   \else%
2570     \setXnoteswidthliketwocolumns@{#1}%
2571     \setXnotespositionliketwocolumns@{#1}%
2572     \print@Xfootnoterule{#1}%
2573   \fi%
2574   \unvbox\csname mp#1footins\endcsname
2575   \ifcsstring{Xragged@#1}{L}{\RaggedLeft}{}%
2576   \ifcsstring{Xragged@#1}{R}{\RaggedRight}{}%
2577   \makehboxofhboxes
2578   \setbox0=\hbox{\csuse{Xnotefontsize@#1}\csuse{Xtxtbeforenotes@#1}}\
unhbox0 \removehboxes}%
2579   \csuse{Xbhookgroup@#1}%
2580   \csuse{Xnotefontsize@#1}%
2581   \nottoggle{Xparindent@#1}{\parindent=\z@}{}%

```

```

2582 \unhbox0\par}}
2583
2584 %

```

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
2587 \unpenalty
2588 \setbox2=\lastbox
2589 \ifhbox2
2590 \setbox0=\hbox{\box2\unhbox0}%
2591 \repeat}
2592
2593 \newcommand*{\removehboxes}{\setbox0=\lastbox
2594 \ifhbox0{\removehboxes}\unhbox0 \fi}
2595
2596 %

```

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

```

\prevpage@num\newcommand{\Xinsertparafootsep}[1]{%
\Xinsertparafootsep
2599 \ifnumequal{\csuse{#1prevpage@num}}{\page@num}%
2600 {\ifcscdef{prevline#1}% Be sur \prevline#1 exists.
2601 {\ifnumequal{\csuse{prevline#1}}{\line@num}%
2602 {\ifcsempty{Xsymlinenum@#1}{\csuse{Xparafootsep@#1}}{}}}%
2603 {\csuse{Xparafootsep@#1}}}%
2604 }%
2605 {\csuse{Xparafootsep@#1}}}%
2606 }%
2607 }%
2608 \global\csname #1prevpage@num\endcsname=\page@num%
2609 }
2609 %

```

XII.6.4 Columnar footnotes

Common tools

`\rigidbalance` We will now define macros for three-column notes and two-column notes. Both sets of macros will use `\rigidbalance`, which splits a box (#1) into into a number (#2) of columns, each with a space (#3) between the top baseline and the top of the `\vbox`. The `\Xrigidbalance` 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 `\dosplits` `\splitoff` `\@h` `\@k`

unboxing in `\splitoff`, which allows the new `\vbox` to have its natural height as it goes into the alignment.

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

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

```

2610 \newcount\@k \newdimen\@h
2611 \newcommand*\Xrigidbalance}[3]{%
2612   \hsize=\expandafter\dimexpr\csuse{Xwidth@\@currentseries}\relax%
2613   \rigidbalance{#1}{#2}{#3}%
2614 }%
2615
2616 \newcommand*\rigidbalanceX}[3]{%
2617   \hsize=\expandafter\dimexpr\csuse{widthX@\@currentseries}\relax%
2618   \rigidbalance{#1}{#2}{#3}%
2619 }%
2620
2621 \newcommand*\rigidbalance}[3]{%
2622   \setbox0=\box#1 \@k=#2 \@h=#3%
2623   \@@line{\splittopskip=\@h \vbadness=\@M \hfilneg
2624     \valign{##\vfil\cr\dosplits}}}%
2625
2626 \newcommand*\dosplits{\ifnum\@k>0 \noalign{\hfil}\splitoff
2627   \global\advance\@k-1\cr\dosplits\fi}
2628
2629 \newcommand*\splitoff{\dimen0=\ht0
2630   \divide\dimen0 by\@k \advance\dimen0 by\@h
2631   \setbox2 \vsplit0 to \dimen0
2632   \unvbox2 }
2633
2634 %

```

Three columns

```

\Xarrangement@threecol35 \newcommand*\Xarrangement@threecol}[1]{%
2636   \csgdef{series@display#1}{threecol}
2637   \expandafter\let\csname v#1footnote\endcsname=\threecolvfootnote
2638   \expandafter\let\csname #1footfmt\endcsname=\threecolfootfmt
2639   \expandafter\let\csname #1footgroup\endcsname=\threecolfootgroup
2640   \dimen\csname #1footins\endcsname=\csuse{Xmaxhnotes@#1}%
2641   \skip\csname #1footins\endcsname=\csuse{Xbeforenotes@#1}%
2642   \advance\skip\csname #1footins\endcsname by\csuse{Xafterrule@#1}%
2643   \threecolfootsetup{#1}
2644 %

```

The additional setup for minipages.

```

2645   \ifnoledgroup@else
2646     \expandafter\let\csname mpv#1footnote\endcsname=\mpnormalvfootnote

```

```

2647 \expandafter\let\csname mp#1footgroup\endcsname=\mpthreecolfootgroup
2648 \skip\csname mp#1footins\endcsname=\csuse{Xbeforenotes@#1}%
2649 \advance\skip\csname mp#1footins\endcsname by\csuse{Xafterrule@#1}%
2650 \mpthreecolfootsetup{#1}
2651 \fi
2652 }
2653
2654 %

```

The `\footstart` and `\footnoterule` macros for these notes assume the normal values (XII.6.2 p. 152 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.

```

2655 \newcommand*{\threecolfootsetup}[1]{%
2656   \count\csname #1footins\endcsname 333
2657   \csxdef{default@#1footins}{333}%Use this to confine the notes to one
side only
2658   \multiply\dimen\csname #1footins\endcsname \thr@@}
2659 %

```

`\mpthreecolfootsetup` The setup for minipages.

```

2660 \newcommand*{\mpthreecolfootsetup}[1]{%
2661   \count\csname mp#1footins\endcsname 333
2662   \multiply\dimen\csname mp#1footins\endcsname \thr@@}
2663
2664 %

```

`\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).

```

2665 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\threecolvfootnote}[2]{%
2666 \csuse{Xbeforeinserting@#1}%
2667 \insert\csname #1footins\endcsname\bgroup%
2668 \hsize=\expandafter\dimexpr\csuse{Xwidth@#1}\relax%
2669 \noindent\csuse{Xbhooknote@#1}%
2670 \csuse{Xnotefontsize@#1}%
2671 \footsplitskips%
2672 \csname #1footfmt\endcsname #2{#1}\egroup}
2673 %

```

\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.

```

2674 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\threecolfootfmt}[4]{%
2675 \normal@pars%
2676 \hsize \csuse{Xhsizethreecol@#4}%
2677 \nottoggle{Xparindent@#4}{\parindent=\z@}{}%
2678 \tolerance=5000%
2679 \hangindent=\csuse{Xhangindent@#4}%
2680 \everypar{\hangindent=\csuse{Xhangindent@#4}}%
2681 \@tempdima=\parindent%
2682 \csuse{Xcolalign@#4}%
2683 \parindent=\@tempdima%
2684 \strut{%
2685 \hspace{\parindent}%
2686 \printlinefootnote{#1}{#4}%
2687 }%
2688 \print@lemma{#1}{#2}{#4}%
2689 \csuse{Xwrapcontent@#4}{#3}%
2690 \strut\par\allowbreak}
2691 %

```

\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.

```

2692 \newcommand*{\threecolfootgroup}[1]{\csuse{Xnotefontsize@#1}%
2693 \noindent\csuse{Xtxtbeforenotes@#1}}%
2694 \csuse{Xbhookgroup@#1}\par%
2695 \splittopskip=\ht\strutbox

```

```

2696 \expandafter
2697 \Xrigidbalance\csname #1footins\endcsname \thr@@ \splittopskip}
2698 %

```

\mpthreecolfootgroup The setup for minipages.

```

2699 \newcommand*{\mpthreecolfootgroup}[1]{%
2700 \vskip\skip\@nameuse{mp#1footins}
2701 \ifl@dpairing\ifparledgroup%
2702 \leavevmode\marks\parledgroup@{begin}%
2703 \marks\parledgroup@series{#1}%
2704 \marks\parledgroup@type{Xfootnote}%
2705 \fi\fi\normalcolor
2706 \ifparledgroup%
2707 \ifl@dpairing%
2708 \else%
2709 \setXnoteswidthliketwocolumns@{#1}%
2710 \setXnotespositionliketwocolumns@{#1}%
2711 \print@Xfootnoterule{#1}%
2712 \fi%
2713 \else%
2714 \setXnoteswidthliketwocolumns@{#1}%
2715 \setXnotespositionliketwocolumns@{#1}%
2716 \print@Xfootnoterule{#1}%
2717 \fi%
2718 {\csuse{Xnotefontsize@#1}\noindent\csuse{Xtxtbeforenotes@#1}}%
2719 \csuse{Xbhookgroup@#1}\par%
2720 \splittopskip=\ht\strutbox
2721 \expandafter
2722 \Xrigidbalance\csname mp#1footins\endcsname \thr@@ \splittopskip}}
2723 %
2724 %

```

Two columns

```

\Xarrangement@twocol25 \newcommand*{\Xarrangement@twocol}[1]{%
2726 \csgdef{series@display#1}{twocol}
2727 \expandafter\let\csname v#1footnote\endcsname=\twocolvfootnote
2728 \expandafter\let\csname #1footfmt\endcsname=\twocolfootfmt
2729 \expandafter\let\csname #1footgroup\endcsname=\twocolfootgroup
2730 \dimen\csname #1footins\endcsname=\csuse{Xmaxhnotes@#1}%
2731 \skip\csname #1footins\endcsname=\csuse{Xbeforenotes@#1}%
2732 \advance\skip\csname #1footins\endcsname by\csuse{Xafterrule@#1}%
2733 \twocolfootsetup{#1}
2734 %

```

The additional setup for minipages.

```

2735 \ifnoledgroup\else
2736 \expandafter\let\csname mpv#1footnote\endcsname=\mpnormalvfootnote

```

```

2737 \expandafter\let\csname mp#1footgroup\endcsname=\mptwocolfootgroup
2738 \skip\csname mp#1footins\endcsname=\csuse{Xbeforenotes@#1}%
2739 \advance\skip\csname mp#1footins\endcsname by\csuse{Xafterrule@#1}%
2740 \mptwocolfootsetup{#1}
2741 \fi
2742 }
2743
2744 %

```

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

```

\twocolvfootnote
\twocolfootfmt
\twocolfootgroup
2745 \newcommand*{\twocolfootsetup}[1]{%
2746 \count\csname #1footins\endcsname 500
2747 \csxdef{default@#1footins}{500}%
2748 \multiply\dimen\csname #1footins\endcsname \tw@}
2749 %

2750 \notbool{parapparatus}{\newcommand*{\twocolvfootnote}[2]{%
2751 \csuse{Xbeforeinserting@#1}%
2752 \insert\csname #1footins\endcsname\bgroup%
2753 \hsz=\expandafter\dimexpr\csuse{Xwidth@#1}\relax%
2754 \noindent\csuse{Xbhooknote@#1}%
2755 \csuse{Xnotefontsize@#1}%
2756 \footssplitsskip%
2757 \csname #1footfmt\endcsname #2{#1}\egroup}
2758 %

2759 \notbool{parapparatus}{\newcommand*{\twocolfootfmt}[4]{% 4th
arg is optional, for backward compatibility
2760 \normal@pars%
2761 \hsz \csuse{Xhsizetwocol@#4}%
2762 \nottoggle{Xparindent@#4}{\parindent=\z@}{}%
2763 \tolerance=5000%
2764 \hangindent=\csuse{Xhangindent@#4}%
2765 \everypar{\hangindent=\csuse{Xhangindent@#4}}%
2766 \@tempdima=\parindent%
2767 \csuse{Xcolalign@#4}%
2768 \parindent=\@tempdima%
2769 \strut{%
2770 \hspace{\parindent}%
2771 \printlinefootnote{#1}{#4}%
2772 }%
2773 \print@lemma{#1}{#2}{#4}%
2774 \csuse{Xwrapcontent@#4}{#3}%
2775 \strut\par\allowbreak}
2776 %

```

```

2777 \newcommand*{\twocolfootgroup}[1]{\csuse{Xnotefontsize@#1}
2778 \noindent\csuse{Xtxtbeforenotes@#1}}%
2779 \csuse{Xbhookgroup@#1}\par%
2780 \splittopskip=\ht\strutbox
2781 \expandafter
2782 \Xrigidbalance\csname #1footins\endcsname \tw@ \splittopskip}
2783
2784 %

```

`\mptwocolfootsetup` The versions for minipages.

`\mptwocolfootgroup`

```

2785 \newcommand*{\mptwocolfootsetup}[1]{%
2786 \count\csname mp#1footins\endcsname 500
2787 \multiply\dimen\csname mp#1footins\endcsname \tw@}
2788 %

2789 \newcommand*{\mptwocolfootgroup}[1]{%
2790 \vskip\skip\@nameuse{mp#1footins}
2791 \ifl@dpairing\ifparledgroup%
2792 \leavevmode\marks\parledgroup@{begin}%
2793 \marks\parledgroup@series{#1}%
2794 \marks\parledgroup@type{Xfootnote}%
2795 \fi\fi\normalcolor
2796 \ifparledgroup%
2797 \ifl@dpairing%
2798 \else%
2799 \setXnoteswidthliketwocolumns@{#1}%
2800 \setXnotespositionliketwocolumns@{#1}%
2801 \print@Xfootnoterule{#1}%
2802 \fi%
2803 \else%
2804 \setXnoteswidthliketwocolumns@{#1}%
2805 \setXnotespositionliketwocolumns@{#1}%
2806 \print@Xfootnoterule{#1}%
2807 \fi%
2808 {\csuse{Xnotefontsize@#1}\noindent\csuse{Xtxtbeforenotes@#1}}%
2809 \csuse{Xbhookgroup@#1}\par%
2810 \splittopskip=\ht\strutbox
2811 \expandafter
2812 \Xrigidbalance\csname mp#1footins\endcsname \tw@ \splittopskip}}
2813
2814 %

```

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` `\fullstop` `\rbracket` The fonts that are used for printing notes might not have the character mapping we expect: for example, the Computer Modern font that contains old-style numerals does not contain an en-dash or square brackets, and its period and comma are in odd locations. To allow use of the standard footnote macros with such fonts, we use the following macros for certain characters.

The `\endashchar` macro is simply an en-dash from the normal font and is immune to changes in the surrounding font. The same goes for the full stop. These two are used in `\printlines`. The right bracket macro is the same again; it crops up in `\normalfootfmt` and the other footnote macros for controlling the format of the footnotes.

With `polyglossia`, each critical note has a `\footnote@lang` which shows the language of the lemma, and which can be used to switch the bracket from right to left.

```

2815 \def\endashchar{\textnormal{--}}
2816
2817 \newcommand*{\fullstop}{\textnormal{.}}
2818 \def\Xsublinesep@side{\fullstop}
2819
2820 \newcommand*{\rbracket}{\textnormal{%
2821   \csuse{text\csuse{footnote@lang}}{%
2822     \ifluatex%
2823       \ifdefstring{\footnote@luatextextdir}{TRT}{\thinspace[]\thinspace
2824     }}%
2825     \else%
2826     \thinspace}%
2827     \fi}%
2828   }}%
2829
2830 %

```

XII.7.2 Pstart number in footnote

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

```

2831 \newcommand{\printpstart}[0]{%
2832   \ifboolexpr{bool{!@dpairing} or bool{!@dprintingpages} or bool{
2833     !@dprintingcolumns}}{%
2834     \ifledRcol%
2835       \thepstartR%
2836     \else%
2837       \thepstartL%
2838     \fi%
2839   }{%
2840     \thepstart%
2841   }}%
2842 %

```

XII.7.3 Lemma printing

`\print@lemma` `\print@lemma` is called inside critical footnotes to print the lemma and the lemma separator (#1: line number and font information, #2: lemma, #3: series).

```

2843 %
2844 \newcommand{\print@lemma}[3]{%
2845   \bgroup%
2846   \nottoggle{Xlemmadisablefontselection@#3}%
2847   {\select@lemmafонт#1|}%
2848   }%
2849   \bgroup%
2850   \csuse{Xlemmafонт@#3}%Deprecated
2851   \csuse{Xwraplemma@#3}{#2}%
2852   \egroup%
2853   \egroup%
2854   \iftoggle{nosep@}{%
2855     \hskip\csuse{Xinplaceoflemmaseparator@#3}%
2856     \relax%
2857   }%
2858   {\ifcsemtty{Xlemmaseparator@#3}%
2859     {%
2860       \hskip\csuse{Xinplaceoflemmaseparator@#3}%
2861       \relax%
2862     }%
2863     {%
2864       \nobreak%
2865       \hskip\csuse{Xbeforelemmaseparator@#3}%
2866       \csuse{Xlemmaseparator@#3}%
2867       \hskip\csuse{Xafterlemmaseparator@#3}%
2868       \relax%
2869     }%
2870   }%
2871 }%
2872 %

```

XII.7.4 Line number printing

`\printlinefootnote` The `\printlinefootnote` macro is called in each `\<type>footfmt` command. It controls whether the line number is printed or not, according to the series options. Its first argument is the information about lines; its second is the series of the footnote. The printing of the line number is shared in `\printlinefootnotenumbers`.

```

2873 \newcommand{\printlinefootnote}[2]{%
2874   \l@dp@rsefootspec#1|}%
2875   \iftoggle{Xnumberonlyfirstintwolines@#2}{%
2876     \edef\lineinfo@{\l@dparsedstartline - \l@dparsedstartsub - \
2877     \l@dparsedendline - \l@dparsedendsub}%
2878   }%

```



```

2879 \edef\lineinfo@{\l@dparsedstartline - \l@dparsedstartsub}%
2880 }%
2881 \iftoggle{nonum@}{%Try if the line number must printed for this specific
not (by default, yes)
2882 \hspace{\csuse{Xinplaceofnumber@#2}}%
2883 }%
2884 {%
2885 {%
2886 \iftoggle{Xnonumber@#2}%Try if the line number must printed (by
default, yes)
2887 {%
2888 \hspace{\csuse{Xinplaceofnumber@#2}}%
2889 }%
2890 {%
2891 {\iftoggle{Xnumberonlyfirstinline@#2}% If for this series the
line number must be printed only in the first time.
2892 {%
2893 \ifcsdef{prevline#2}%
2894 {%Be sure the \prevline exists.
2895 \ifcsequal{prevline#2}{\lineinfo@}%Try it
2896 {%
2897 \ifcseempty{Xsymlinenum@#2}%
2898 {%
2899 \hspace{\csuse{Xinplaceofnumber@#2}}%
2900 }%
2901 {\printsymlinefootnotearea{#2}}%
2902 }%
2903 {%
2904 \printlinefootnotearea{#1}{#2}%
2905 }%
2906 }%
2907 {%
2908 \printlinefootnotearea{#1}{#2}%
2909 }%
2910 }%
2911 {%
2912 \printlinefootnotearea{#1}{#2}%
2913 }%
2914 \csxdef{prevline#2}{\lineinfo@}%
2915 }%
2916 }%
2917 }%
2918 }%
2919 }
2920 %

```

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

```

2921 \newcommand{\printsymlinefootnotearea}[1]{%

```

```

2922 \hspace{\csuse{Xbeforesymlinewidth@#1}}%
2923 \csuse{Xnotenumfont@#1}%
2924 \ifdimequal{\csuse{Xboxsymlinewidth@#1}}{\z@}%
2925   {\csuse{Xsymlinewidth@#1}}%
2926   {\hbox to \csuse{Xboxsymlinewidth@#1}%
2927     {\csuse{Xsymlinewidth@#1}\hfill}%
2928   }%
2929 \hspace{\csuse{Xaftersymlinewidth@#1}}%
2930 }%
2931 %

```

\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.)

```

2932 \newcommand{\printlinefootnotearea}[2]{%
2933   \printXbeforenumber{#2}%
2934   \csuse{Xnotenumfont@#2}%
2935   \boxfootnotenumbers{#1}{#2}%
2936   \printXafternumber{#2}%
2937 }%
2938 %

```

\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.

```

2939 \newcommand{\boxfootnotenumbers}[2]{%
2940   \ifdimequal{\csuse{Xboxlinewidth@#2}}{\z@}{%
2941     \printlinefootnotenumbers{#1}{#2}%
2942   }%
2943   {%
2944     \hbox to \csuse{Xboxlinewidth@#2}%
2945     {%
2946       \IfSubStr{RC}{\csuse{Xboxlinenumalign@#2}}{\hfill}{}%
2947       \printlinefootnotenumbers{#1}{#2}%
2948       \IfSubStr{LC}{\csuse{Xboxlinenumalign@#2}}{\hfill}{}%
2949     }%
2950   }%
2951 }%
2952 %

```

\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.

```

2953 \newcommand{\printlinefootnotenumbers}[2]{%
2954   \xdef\@currentseries{#2}%

```

```

2955 \ifboolexpr{%
2956   (togl{Xpstart@#2} and bool{numberpstart}})%
2957   or togl{Xpstarteverytime@#2}})%
2958 {\printpstart}{}%
2959 \iftoggle{Xstanza@#2}{%
2960   \ifnumberstanza%
2961     \printstanza%
2962     \csuse{Xstanzaseparator@#2}%
2963   \fi}%
2964 }{%}%
2965 \iftoggle{Xonlypstart@#2}{%}%
2966   \csuse{Xtxtbeforenumber@#2}%
2967   \printlines#1||\ifledRcol@{\@Rlineflag\fi}%
2968 }%
2969 %

```

\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.)

```

2970 \newcommand{\printXbeforenumber}[1]{%
2971   \hspace{\csuse{Xbeforenumber@#1}}%
2972 }%
2973 %

```

\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

```

2974 \newcommand{\printXafternumber}[1]{%
2975   \iftoggle{Xnonbreakableafternumber@#1}{\nobreak}{}%
2976   \hspace{\csuse{Xafternumber@#1}}%
2977 }%
2978 %

```

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. 89: 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.

Maïeul Rouquette has added `\ifl@d@Xtwolines` and `\ifl@d@Xmorethantwolines` to print a symbol which stands for “and subsequent” when there are two, three or more lines.

```

\ifl@d@pnum79 \newif\ifl@d@pnum
\ifl@d@ssub80 \newif\ifl@d@ssub
\ifl@d@elin81 \newif\ifl@d@elin
\ifl@d@esl82 \newif\ifl@d@esl
\ifl@d@dash83 \newif\ifl@d@dash
\ifl@d@Xtwolines84 \newif\ifl@d@Xtwolines%
\ifl@d@Xmorethantwolines85 \newif\ifl@d@Xmorethantwolines%
86 %

\l@dp@rsefootsspec \l@dp@rsefootsspec parses lines specification and defines macros which hold the nu-
\l@dparsedstartpage meric values. Just a reminder of the arguments:
\l@dparsedstartline \printlines #1 | #2 | #3 | #4 | #5 | #6 | #7
\l@dparsedstartsub \printlines start-page | line | subtitle | end-page | line | subtitle | fontflag
\l@dparsedendpage2987 \def\l@dp@rsefootsspec#1|#2|#3|#4|#5|#6|#7|{%
\l@dparsedendline2988 \gdef\l@dparsedstartpage{#1}%
\l@dparsedendsub2989 \gdef\l@dparsedstartline{#2}%
2990 \gdef\l@dparsedstartsub{#3}%
2991 \gdef\l@dparsedendpage{#4}%
2992 \gdef\l@dparsedendline{#5}%
2993 \gdef\l@dparsedendsub{#6}%
2994 }
2995 %

Initialise the several number value macros.

2996 \def\l@dparsedstartpage{0}%
2997 \def\l@dparsedstartline{0}%
2998 \def\l@dparsedstartsub{0}%
2999 \def\l@dparsedendpage{0}%
3000 \def\l@dparsedendline{0}%
3001 \def\l@dparsedendsub{0}%
3002
3003 %

```

`\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`.

```

3004 \newcommand*{\setprintlines}[6]{%
3005   \l@d@pnumfalse \l@d@dashfalse
3006 %

```

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

```

3007 \ifbypage@
3008   \ifnum#4=#1 \else
3009     \l@dpnumtrue
3010     \l@ddashtrue
3011   \fi
3012 \fi
3013 %

```

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.

```

3014 \ifl@dpnum \l@delintrue \else \l@delinfalse \fi
3015 \ifnum#2=#5 \else
3016   \l@delintrue
3017   \l@ddashtrue
3018 \fi
3019 %

```

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

```

3020 \l@ssubfalse
3021 \ifnum#3=0 \else
3022   \l@ssubtrue
3023 \fi
3024 %

```

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.

```

3025 \l@eslfalse
3026 \ifnum#6=0 \else
3027   \ifnum#6=#3
3028     \ifl@delin \l@esltrue \else \l@eslfalse \fi
3029   \else
3030     \l@esltrue
3031     \l@ddashtrue
3032   \fi
3033 \fi%
3034 %

```

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

```

3035 \ifl@ddash%
3036   \ifboolexpr{togl{fulllines@} or test{\ifcsemt{Xtwolines@}\
@currentseries}}}%
3037   {%
3038   {%
3039     \setistwofollowinglines{#1}{#2}{#4}{#5}%
3040     \ifboolexpr{%
3041       (%
3042         togl {Xtwolinesbutnotmore@\@currentseries}%
3043         and not%

```

```

3044      (%)
3045      bool {istwofollowinglines@}%
3046      )%
3047  )%
3048  or%
3049  (%)
3050      (not test{\ifnumequal{#1}{#4}})%
3051      and tog1{Xtwolinesonlyinsamepage@\@currentseries}%
3052      )%
3053  }%
3054  {}%
3055  {%
3056      \l@d@dashfalse%
3057      \l@d@Xtwolinestrue%
3058      \l@d@elinfalse%
3059      \l@d@eslfalse%
3060      \ifcempty{Xmorethantwolines@\@currentseries}%
3061      {}%
3062      {\ifistwofollowinglines@\else%
3063          \l@d@Xmorethantwolinestrue%
3064          \fi%
3065      }%
3066  }%
3067  }%
3068  \fi%
3069  %

End of \setprintlines.

3070 }%
3071 %

```

`\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.

```

3072 \newif\ifistwofollowinglines@%
3073 \newcommand{\setistwofollowinglines}[4]{%
3074   \ifcsdef{lastlinenumberon@#1}%
3075     {\numdef{\tmp}{\csuse{lastlinenumberon@#1}}}%
3076     {\numdef{\tmp}{0}}%
3077   \istwofollowinglines@false%
3078   \ifnumequal{#4-#2}{1}%
3079     {\istwofollowinglines@true}%
3080     {\ifbypage@%
3081       \ifnumequal{#3-#1}{1}%
3082       {%
3083         \ifnumequal{#2}{\tmp}%
3084         {\ifnumequal{#4}{1}{\istwofollowinglines@true}{}}%
3085         {}%
3086       }%
3087       {}%
3088     \fi%
3089   }%
3090 }%
3091 %

```

`\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

```

3092 \def\printlines#1|#2|#3|#4|#5|#6|#7|#8|{%
3093   \begingroup%
3094   %

```

If we use Lua_{TeX}, ensure we use good text's direction.

```

3095 \ifluatex%
3096   \edef\@tmp{\the\textdir}%
3097   \ifdefstring{\@tmp}{TLT}{\textdir TLT}%Test in order to prevent
    spurious space (bug #397)
3098   \fi%
3099 %

```

Decide which part of line number components we will print.

```

3100 \setprintlines{#1}{#2}{#3}{#4}{#5}{#6}%
3101 %

```

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.

```

3102 \ifdimequal{\csuse{Xboxstartlinenum@\@currentseries}}{0pt}%
3103 {\bgroup}%
3104 {\leavevmode\hbox to \csuse{Xboxstartlinenum@\@currentseries}\bgroup\
hfill}%
3105 \ifl@d@pnum%
3106 \wrap@edcrossref{\@this@crossref@start}{#1}%
3107 \csuse{Xsublinesep@\@currentseries}%
3108 \fi%
3109 \wrap@edcrossref{\@this@crossref@start}{%
3110 \linenumrep{#2}%
3111 \iftoggle{Xlineflag@\@currentseries}{#8}{}}%
3112 }%
3113 \ifl@d@ssub%
3114 \csuse{Xsublinesep@\@currentseries}%
3115 \wrap@edcrossref{\@this@crossref@start}{\sublinenumrep{#3}}%
3116 \fi
3117 \egroup%
3118 %

```

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

```

3119 \ifdimequal{\csuse{Xboxendlinenum@\@currentseries}}{0pt}%
3120 {\bgroup}%
3121 {\hbox to \csuse{Xboxendlinenum@\@currentseries}\bgroup}%
3122 \ifl@d@Xtwolines%
3123 \ifl@d@Xmorethantwolines%
3124 \csuse{Xmorethantwolines@\@currentseries}%
3125 \else%
3126 \csuse{Xtwolines@\@currentseries}%
3127 \fi%
3128 \else%
3129 \ifl@d@dash%
3130 \ifdefined\linrangesep%
3131 \linrangesep%
3132 \else%
3133 \csuse{Xlinrangeseparator@\@currentseries}%
3134 \fi%
3135 \fi%
3136 \ifl@d@pnum%
3137 \wrap@edcrossref{\@this@crossref@end}{#4}%
3138 \csuse{Xsublinesep@\@currentseries}%
3139 \fi%
3140 \ifl@d@elin%
3141 \wrap@edcrossref{\@this@crossref@end}{%
3142 \linenumrep{#5}%
3143 \iftoggle{Xlineflag@\@currentseries}{#8}{}}%
3144 }%
3145 \fi%
3146 \ifl@d@esl%
3147 \ifl@d@elin%
3148 \csuse{Xsublinesep@\@currentseries}%

```



```

3149     \fi%
3150     \wrap@edcrossref{\@this@crossref@end}{\sublinenumrep{#6}}%
3151     \fi%
3152 \fi%
3153 \ifdimequal{\csuse{Xboxendlinenum@\@currentseries}}{0pt}%
3154 {}%
3155 {\hfill}%Prevent underfull hbox
3156 \egroup%
3157 \endgroup%
3158 }%
3159 %

```

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`

```

3160 \providecommand*\multiplefootnotemarker{3sp}
3161 \providecommand*\multfootsep{\textsuperscript{\normalfont,}}
3162
3163 %

```

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

```

3164 \providecommand*\m@mmf@prepare{%
3165   \kern-\multiplefootnotemarker
3166   \kern\multiplefootnotemarker\relax}
3167 %

```

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

```

3168 \providecommand*\m@mmf@check{%
3169   \ifdim\lastkern=\multiplefootnotemarker\relax
3170   \edef\x@sf{\the\spacefactor}%

```

```

3171 \unkern
3172 \multfootsep
3173 \spacefactor\@x@sf\relax
3174 \fi}
3175
3176 %

```

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

```

3177 \ifclassloaded{memoir}{\}%
3178 %

```

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

```

3179 \apptocmd{\@footnotetext}{\m@mmf@prepare}{\}{\}
3180 %

```

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

```

3181
3182 \patchcmd{\@footnotemark}
3183 {\nobreak}
3184 {\m@mmf@check
3185 \nobreak
3186 }
3187 {}{}
3188 \patchcmd{\@footnotemark}
3189 {\@makefnmark}
3190 {\@makefnmark
3191 \m@mmf@prepare
3192 }
3193 {}{}
3194 %

```

Finished the modifications for the non-memoir case.

```

3195 }
3196
3197 %

```

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 with its `\@footnotetext`, using different forms for when in numbered or regular text.

```

3198 \pretocmd{\@footnotetext}{%
3199   \ifnumberedpar@
3200   \edtext{}\{\l@dbfnote{#1}}}%
3201   \else
3202   }{}{}
3203 \apptocmd{\@footnotetext}{\fi}{}{}%
3204 %

```

`\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`

```

3205 \get@thisfootnote
3206 \patchcmd%
3207   {\footnote}%
3208   {\stepcounter\@mpfn}%
3209   {%
3210   \ifl@dpairing%
3211   \global\advance\footnote@reading by \@ne%
3212     \get@thisfootnote%
3213     \get@fnmark{\thisfootnote}%
3214     \ifcsdef{footnotereading\the\footnote@reading=typeset}%
3215     {\setcounter{\@mpfn}{\csuse{footnotereading\the\footnote@reading=
typeset}}}%
3216     {\setcounter{\@mpfn}{\footnote@reading}}%
3217     \else%
3218     \stepcounter\@mpfn%
3219     \fi%
3220   }%
3221   {}
3222   {}
3223
3224 \newcommand{\get@thisfootnote}{%
3225   \ifl@dpairing
3226     \protected@xdef\thisfootnote{\the\footnote@reading}%
3227   \else%
3228     \protected@xdef\thisfootnote{\the\footnote}%
3229   \fi%
3230 }%
3231
3232 \newcommand{\l@dbfnote}[1]{%
3233   \get@thisfootnote%
3234   \gdef\@tag{#1\relax}%
3235   \ifledRcol%
3236     \xright@appenditem{%
3237       \ifdefined\Hy@footnote@currentHref%
3238       \noexpand\def\noexpand\Hy@footnote@currentHref{\
Hy@footnote@currentHref}%
3239       \fi%

```

```

3240 \noexpand\vl@dbfnote{\expandonce\@tag}}{\thisfootnote}%
3241 }%
3242 \to\inserts@listR
3243 \global\advance\insert@countR \@ne%
3244 \else%
3245 \xright@appenditem{%
3246 \ifdefined\Hy@footnote@currentHref%
3247 \noexpand\def\noexpand\Hy@footnote@currentHref{\
Hy@footnote@currentHref}%
3248 \fi%
3249 \noexpand\vl@dbfnote{\expandonce\@tag}}{\thisfootnote}%
3250 }%
3251 \to\inserts@list
3252 \global\advance\insert@count \@ne%
3253 \fi
3254 \ignorespaces%
3255 }%
3256
3257 \newcommand{\get@fnmark}[1]{%
3258 \ifboolexpr{bool{l@dpairing} or bool{l@dprintingpages} or bool{
l@dprintingcolumns}}%
3259 {%
3260 \stepcounter{footnote@typeset}%
3261 \setcounter{footnote}{\c@footnote@typeset}%
3262 \immediate\write\@mainaux{%
3263 \global\csdef{footnotereading#1=typeset}{\the\c@footnote@typeset}
%
3264 }%
3265 \def\@thefnmark{\thefootnote}%
3266 }%
3267 {%
3268 \@namedef{\@thefnmark}{#1}%
3269 }%
3270 }%
3271
3272 \newcommand{\vl@dbfnote}[2]{%
3273 \get@fnmark{#2}%
3274 \@footnotetext{#1}%
3275 }%
3276 %

```

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`

```

3277 \newcommand*{\prebodyfootmark}{%

```

```

3278 \leavevmode
3279 \ifhmode
3280   \edef\@x@sf{\the\spacefactor}%
3281   \m@mmf@check
3282   \nobreak
3283   \fi}
3284 \newcommand{\postbodyfootmark}{%
3285   \m@mmf@prepare
3286   \ifhmode\spacefactor\@x@sf\fi\relax}
3287
3288 %

```

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.

```

3289 \newcommandx{\arrangementX}[2][1,usedefault]{%
3290   \def\do##1{%
3291     \csname arrangementX@#2\endcsname{##1}%
3292   }%
3293   \ifstrempy{#1}%
3294     {%
3295       \dolistloop{\@series}%
3296     }%
3297     {
3298       \docsvlist{#1}%
3299     }%
3300 }%
3301 %

```

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.

```

3302 \newcommand*{\normal@footnotemarkX}[1]{%
3303   \prebodyfootmark
3304   \wrapped@bodyfootmarkX{#1}%
3305   \postbodyfootmark}
3306
3307 %

```

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

```

3308 \newcommand*{\normalbodyfootmarkX}[1]{%
3309   \hbox{\textsuperscript{\normalfont\@nameuse{@thefnmark#1}}}}

```

```

3310 %

\normalvfootnoteX \normalvfootnoteX{<series>}{<text>} does the \insert for the <series> and calls the
series' \footfmt... to format the <text>.

3311 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\normalvfootnoteX}[2]{%
3312   \csuse{beforeinsertingX@#1}%
3313   \insert\@nameuse{footins#1}\bgroup
3314     \hsize=\expandafter\dimexpr\csuse{widthX@#1}\relax%
3315     \noindent\csuse{bhooknoteX@#1}%
3316     \csuse{notefontsizeX@#1}%
3317     \footsplitskips
3318     \ifl@dpairing\ifl@dpaging\else%
3319       \setnoteswidthliketwocolumnsX@{#1}%
3320     \fi\fi%
3321     \setnotesXpositionliketwocolumns@{#1}%
3322     \spaceskip=\z@skip \xspaceskip=\z@skip
3323     \csuse{\csuse{footnote@dir}}\@nameuse{footfmt#1}{#1}{#2}\egroup}
3324
3325 %

```

\mpnormalvfootnoteX The minipage version.

```

3326 \newcommand*{\mpnormalvfootnoteX}[2]{%
3327   \get@thisfootnoteX{#1}%
3328   \get@fnmarkX{#1}{\thisfootnote}%
3329   \edef\this@footnoteX@reading{\the\csname footnote#1@reading\endcsname}%
3330   \global\setbox\@nameuse{mpfootins#1}\vbox{%
3331     \unvbox\@nameuse{mpfootins#1}
3332     \noindent\csuse{bhooknoteX@#1}%
3333     \csuse{notefontsizeX@#1}%
3334     \hsize\columnwidth
3335     \@parboxrestore
3336     \color@begingroup
3337     \@nameuse{footfmt#1}{#1}{#2}\color@endgroup}}
3338
3339 %

```

\normalfootfmtX \normalfootfmtX{<series>}{<text>} typesets the footnote text, prepended by the marker.

```

3340 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\normalfootfmtX}[2]{%
3341   \ifluatex%
3342     \texdir\footnote@luatextextdir%
3343     \pardir\footnote@luatexpardir%
3344     \par%
3345   \fi%
3346   \protected@edef\@currentlabel{%
3347     \@nameuse{@thefnmark#1}%
3348   }%
3349   \ledsetnormalparstuffX{#1}%

```

```

3350 \hangindent=\csuse{hangindentX@#1}%
3351 \everypar{\hangindent=\csuse{hangindentX@#1}}%
3352 {\csuse{notenumfontX@#1}\wrapped@footfootmarkX{#1}}\strut%
3353 \csuse{wrapcontentX@#1}{#2}%
3354 \strut\par}}
3355
3356 %

```

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

```

3357 \newcommand*{\normalfootfootmarkX}[1]{%
3358 \textsuperscript{\@nameuse{@thefnmark#1}}}%
3359
3360 %

```

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

```

3361 \newcommand*{\normalfootstartX}[1]{%
3362 \ifdimequal{Opt}{\prenotesX@}{}%
3363 {%
3364 \iftoggle{prenotesX@}{%
3365 \togglefalse{prenotesX@}%
3366 \skip\csname footins#1\endcsname=%
3367 \dimexpr\csuse{prenotesX@}+\csuse{afterruleX@#1}\relax%
3368 }%
3369 }%
3370 }%
3371 \vskip\skip\csname footins#1\endcsname%
3372 \leftskip=\z@
3373 \rightskip=\z@
3374 \ifl@dpairing\else%
3375 \hspace=\old@hspace%
3376 \fi%
3377 \setnoteswidthliketwocolumnsX@{#1}%
3378 \setnotesXpositionliketwocolumns@{#1}%
3379 \print@footnoteXrule{#1}%
3380 }%
3381
3382 %

```

\normalfootnoteruleX The rule drawn before the footnote series group.

```

3383 \let\normalfootnoteruleX=\footnoterule
3384
3385 %

```

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

```

3386 \newcommand*{\normalfootgroupX}[1]{%
3387   \csuse{bhookgroupX@#1}%
3388   \unvbox\@nameuse{footins#1}%
3389   \hsize=\old@hsize%
3390 }%
3391
3392 %

```

`\mpnormalfootgroupX` The minipage version.

```

3393 \newcommand*{\mpnormalfootgroupX}[1]{%
3394   \vskip\skip\@nameuse{mpfootins#1}
3395   \ifl@dpairing\ifparledgroup%
3396     \leavevmode\marks\parledgroup@{begin}%
3397     \marks\parledgroup@series{#1}%
3398     \marks\parledgroup@type{footnoteX}%
3399     \fi\fi\normalcolor
3400   \ifparledgroup%
3401     \ifl@dpairing%
3402     \else%
3403       \setnoteswidthliketwocolumnsX@{#1}%
3404       \setnotesXpositionliketwocolumns@{#1}%
3405       \print@footnoteXrule{#1}%
3406     \fi%
3407   \else%
3408     \setnoteswidthliketwocolumnsX@{#1}%
3409     \setnotesXpositionliketwocolumns@{#1}%
3410     \print@footnoteXrule{#1}%
3411   \fi%
3412   \csuse{bhookgroupX@#1}%
3413   \unvbox\@nameuse{mpfootins#1}}
3414
3415 %

```

`\normalbfnoteX`₁₆

```

3417 \newcommand{\normalbfnoteX}[2]{%
3418   \get@thisfootnoteX{#1}%
3419   \ifledRcol%
3420     \ifluatex
3421       \footnotelang@lua[R]%
3422     \fi
3423     \@ifundefined{xpg@main@language}%if polyglossia
3424       {}%
3425       {\footnotelang@poly[R]}%
3426     \xright@appenditem{%
3427       \noexpand\led@set@index@fornote{#1}%
3428       \unexpanded{\def\this@footnoteX@reading}{\the\csname footnote#1
3429 @reading\endcsname}%
       \noexpand\vbfnoteX{#1}{#2}{\expandonce\thisfootnote}%

```



```

3430 \noexpand\led@reinit@index@fornote%
3431 }%
3432 \to\inserts@listR
3433 \global\advance\insert@countR \@ne%
3434 \else%
3435 \ifluatex
3436 \footnotelang@lua%
3437 \fi
3438 \@ifundefined{xpg@main@language}%if polyglossia
3439 {}%
3440 {\footnotelang@poly}%
3441 \xright@appenditem{%
3442 \noexpand\led@set@index@fornote{#1}%
3443 \unexpanded{\def\this@footnoteX@reading}{\the\csname footnote#1
@reading\endcsname}%
3444 \noexpand\vbfnoteX{#1}{#2}{\expandonce\thisfootnote}%
3445 \noexpand\led@reinit@index@fornote%
3446 }%
3447 \to\inserts@list
3448 \global\advance\insert@count \@ne%
3449 \fi
3450 \ignorespaces}
3451
3452 %

```

\get@thisfootnoteX The macro `\get@thisfootnote` command just saves the footnote number in the `\thisfootnote` macro, depending on the use of pairing environments.

```

3453 \newcommand{\get@thisfootnoteX}[1]{%
3454 \ifl@pairing%
3455 \protected@xdef\thisfootnote{\the\csname footnote#1@reading\endcsname
}%
3456 \else%
3457 \protected@xdef\thisfootnote{\csuse{thefootnote#1}}%
3458 \fi%
3459 }%
3460 %

```

\vbfnoteX This command calls the correct footnote-inserting commands.

```

3461 \newcommand{\vbfnoteX}[3]{%
3462 \get@fnmarkX{#1}{#3}%
3463 \@nameuse{regvfootnote#1}{#1}{#2}%
3464 }%
3465
3466 %

```

\get@fnmarkX This command gets the correct footnote number when typesetting parallel texts.

```

3467 \newcommand{\get@fnmarkX}[2]{%
3468   \ifboolexpr{bool{l@dpairing} or bool{l@dprintingpages} or bool{
l@dprintingcolumns}}{%
3469     {%
3470       \stepcounter{footnote#1@typeset}%
3471       \setcounter{footnote#1}{\value{footnote#1@typeset}}%
3472       \@namedef{@thefnmark#1}{\csuse{thefootnote#1}}%
3473       \immediate\write\@mainaux{%
3474         \global\csdef{footnote#1reading#2=typeset}{\the\csname c@footnote
#1@typeset\endcsname}%
3475       }%
3476     }%
3477     {%
3478       \@namedef{@thefnmark#1}{#2}%
3479     }%
3480   }
3481   %
3482   %

```

```

\vnunfootnoteX83 \newcommand{\vnunfootnoteX}[2]{%
3484   \ifnumberedpar@
3485     \edtext{}{\normalbfnoteX{#1}{#2}}%
3486   \else
3487     \def\this@footnoteX@reading{\the\csname footnote#1@reading\endcsname}%
3488     \@nameuse{regvfootnote#1}{#1}{#2}%
3489   \fi}
3490
3491   %

```

`arrangementX@normal` `\arrangementX@normal{⟨series⟩}` initialises the settings for the `⟨series⟩` footnotes. This should always be called for each series.

```

3492 \newcommand*{\arrangementX@normal}[1]{%
3493   \csgdef{series@displayX#1}{normal}
3494   \expandafter\let\csname footstart#1\endcsname=\normalfootstartX
3495   \expandafter\newcount\csname prevpage#1@num\endcsname
3496   \@namedef{@footnotemark#1}{\normal@footnotemarkX{#1}}
3497   \@namedef{bodyfootmark#1}{\normalbodyfootmarkX{#1}}
3498   \expandafter\let\csname regvfootnote#1\endcsname=\normalvfootnoteX
3499   \expandafter\let\csname vfootnote#1\endcsname=\vnunfootnoteX
3500   \expandafter\let\csname footfmt#1\endcsname=\normalfootfmtX
3501   \@namedef{footfootmark#1}{\normalfootfootmarkX{#1}}
3502   \expandafter\let\csname footgroup#1\endcsname=\normalfootgroupX
3503   \expandafter\let\csname footnoterule#1\endcsname=\normalfootnoteruleX
3504   \count\csname footins#1\endcsname=1000
3505   \csxdef{default@footins#1}{1000}%Use to have note only for one side
3506   \dimen\csname footins#1\endcsname=\csuse{maxhnotesX@#1}
3507   \skip\csname footins#1\endcsname=\csuse{beforenotesX@#1}%
3508   \advance\skip\csname footins#1\endcsname by\csuse{afterruleX@#1}%

```

```

3509 %
Additions for minipages.
3510 \ifnoledgroup@else%
3511 \expandafter\let\csname mpvfootnote#1\endcsname=\mpnormalvfootnoteX
3512 \expandafter\let\csname mpfootgroup#1\endcsname=\mpnormalfootgroupX
3513 \count\csname mpfootins#1\endcsname=1000
3514 \dimen\csname mpfootins#1\endcsname=\csuse{maxhnotesX@#1}%
3515 \skip\csname mpfootins#1\endcsname=\csuse{beforenotesX@#1}%
3516 \advance\skip\csname mpfootins#1\endcsname by\csuse{afterruleX@#1}%
3517 \fi
3518 }
3519
3520 %

```

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 21 \newcommand*{\arrangementX@twocol}[1]{%
3522 \csgdef{series@displayX#1}{twocol}
3523 \expandafter\let\csname regvfootnote#1\endcsname=\twocolvfootnoteX
3524 \expandafter\let\csname footfmt#1\endcsname=\twocolfootfmtX
3525 \expandafter\let\csname footgroup#1\endcsname=\twocolfootgroupX
3526 \dimen\csname footins#1\endcsname=\csuse{maxhnotesX@#1}%
3527 \skip\csname footins#1\endcsname=\csuse{beforenotesX@#1}%
3528 \advance\skip\csname footins#1\endcsname by \csuse{afterruleX@#1}\relax%
3529 \twocolfootsetupX{#1}
3530 \ifnoledgroup@else%
3531 \expandafter\let\csname mpvfootnote#1\endcsname=\mpnormalvfootnoteX
3532 \expandafter\let\csname mpfootgroup#1\endcsname=\mptwocolfootgroupX
3533 \skip\csname mpfootins#1\endcsname=\csuse{beforenotesX@#1}%
3534 \advance\skip\csname mpfootins#1\endcsname by\csuse{afterruleX@#1}
3535 \mptwocolfootsetupX{#1}
3536 \fi%
3537 }
3538
3539 %

```

```

\twocolfootsetupX \twocolfootsetupX{\series}
\mptwocolfootsetupX
3540 \newcommand*{\twocolfootsetupX}[1]{%
3541 \count\csname footins#1\endcsname 500
3542 \csxdef{default@footins#1}{500}%Use this to confine the notes to one
side only
3543 \multiply\dimen\csname footins#1\endcsname by \tw@}
3544 \newcommand*{\mptwocolfootsetupX}[1]{%
3545 \count\csname mpfootins#1\endcsname 500

```

```

3546 \multiply\dimen\csname mpfootins#1\endcsname by \tw@
3547
3548 %

```

`\twocolvfootnoteX` `\twocolvfootnoteX{<series>}`

```

3549 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\twocolvfootnoteX}[2]{%
3550 \csuse{beforeinsertingX@#1}%
3551 \insert\csname footins#1\endcsname\bgroup%
3552 \hsize=\expandafter\dimexpr\csuse{widthX@#1}\relax%
3553 \noindent\csuse{bhooknoteX@#1}%
3554 \csuse{notefontsizeX@#1}%
3555 \footsplittopskip%
3556 \spaceskip=\z@skip \xspaceskip=\z@skip%
3557 \@nameuse{footfmt#1}{#1}{#2}\egroup}
3558
3559 %

```

`\twocolfootfmtX` `\twocolfootfmtX{<series>}`

```

3560 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\twocolfootfmtX}[2]{%
3561 \protected@edef\@currentlabel{%
3562 \@nameuse{@thefnmark#1}%
3563 }%
3564 \normal@pars%
3565 \hangindent=\csuse{hangindentX@#1}%
3566 \everypar{\hangindent=\csuse{hangindentX@#1}}%
3567 \hsize \csuse{hsizetwocolX@#1}%
3568 \nottoggle{parindentX@#1}{\parindent=\z@}{}%
3569 \tolerance=5000\relax%
3570 \par%
3571 \@tempdima=\parindent%
3572 \csuse{colalignX@#1}%
3573 \parindent=\@tempdima%
3574 {\hspace{\parindent}%
3575 \csuse{notenumfontX@#1}\wrapped@footfootmarkX{#1}\strut%
3576 \csuse{wrapcontentX@#1}{#2}%
3577 \strut\par}%
3578 \allowbreak%
3579 }%
3580
3581 %

```

`\twocolfootgroupX` `\twocolfootgroupX{<series>}`

`\mptwocolfootgroupX`

```

3582 \newcommand*{\twocolfootgroupX}[1]{\csuse{bhookgroupX@#1}\csuse{
notefontsizeX@#1}
3583 \splittopskip=\ht\strutbox
3584 \expandafter
3585 \rigidbalanceX\csname footins#1\endcsname \tw@ \splittopskip}}

```

```

3586 \newcommand*{\mptwocolfootgroupX}[1]{%
3587   \vskip\skip\@nameuse{mpfootins#1}
3588   \ifl@dpairing\ifparledgroup%
3589     \leavevmode\marks\parledgroup@{begin}%
3590     \marks\parledgroup@series{#1}%
3591     \marks\parledgroup@type{footnoteX}%
3592   \fi\fi\normalcolor
3593   \ifparledgroup%
3594     \ifl@dpairing%
3595     \else%
3596       \setnoteswidthliketwocolumnsX@{#1}%
3597       \setnotesXpositionliketwocolumns@{#1}%
3598       \print@footnoteXrule{#1}%
3599     \fi%
3600   \else%
3601     \setnoteswidthliketwocolumnsX@{#1}%
3602     \setnotesXpositionliketwocolumns@{#1}%
3603     \print@footnoteXrule{#1}%
3604   \fi%
3605   \csuse{bhookgroupX@#1}%
3606   \splittopskip=\ht\strutbox
3607   \expandafter
3608   \rigidbalanceX\csname mpfootins#1\endcsname \tw@ \splittopskip}}
3609
3610 %
3611

```

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
12 \newcommand*{\arrangementX@threecol}[1]{%
3613   \csgdef{series@displayX#1}{threecol}
3614   \expandafter\let\csname regvfootnote#1\endcsname=\threecolvfootnoteX
3615   \expandafter\let\csname footfmt#1\endcsname=\threecolfootfmtX
3616   \expandafter\let\csname footgroup#1\endcsname=\threecolfootgroupX
3617   \dimen\csname footins#1\endcsname=\csuse{maxhnotesX@#1}%
3618   \skip\csname footins#1\endcsname=\csuse{beforenotesX@#1}%
3619   \advance\skip\csname footins#1\endcsname by \csuse{afterruleX@#1}\relax%
3620   \threecolfootsetupX{#1}
3621   \ifnoledgroup\else%
3622     \expandafter\let\csname mpvfootnote#1\endcsname=\mpnormalvfootnoteX
3623     \expandafter\let\csname mpfootgroup#1\endcsname=\mpthreecolfootgroupX
3624     \skip\csname mpfootins#1\endcsname=\csuse{beforenotesX@#1}%
3625     \advance\skip\csname mpfootins#1\endcsname by \csuse{afterruleX@#1}
3626     \mpthreecolfootsetupX{#1}
3627   \fi%
3628 }

```

```

3629
3630 %

\threecolfootsetupX \threecolfootsetupX{<series>}
\mpthreecolfootsetupX
3631 \newcommand*{\threecolfootsetupX}[1]{%
3632   \count\csname footins#1\endcsname 333
3633   \csxdef{default@footins#1}{333}%Use this to confine the notes to one
side only
3634   \multiply\dimen\csname footins#1\endcsname by \thr@@
3635   \newcommand*{\mpthreecolfootsetupX}[1]{%
3636     \count\csname mpfootins#1\endcsname 333
3637     \multiply\dimen\csname mpfootins#1\endcsname by \thr@@
3638
3639 %

\threecolvfootnoteX \threecolvfootnoteX{<series>}{<text>}

3640 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\threecolvfootnoteX}[2]{
%
3641   \csuse{beforeinsertingX@#1}%
3642   \insert\csname footins#1\endcsname\bgroup%
3643     \hsize=\expandafter\dimexpr\csuse{widthX@#1}\relax%
3644     \noindent\csuse{bhooknoteX@#1}%
3645     \csuse{notefontsizeX@#1}%
3646     \footsplitskips%
3647     \@nameuse{footfmt#1}{#1}{#2}\egroup}
3648
3649 %

\threecolfootfmtX \threecolfootfmtX{<series>}

3650 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\threecolfootfmtX}[2]{%
3651   \protected@edef\@currentlabel{%
3652     \@nameuse{thefnmark#1}%
3653   }%
3654   \hangindent=\csuse{hangindentX@#1}%
3655   \everypar{\hangindent=\csuse{hangindentX@#1}}%
3656   \normal@pars%
3657   \hsize \csuse{hsizethreecolX@#1}%
3658   \nottoggle{parindentX@#1}{\parindent=\z@}{}%
3659   \tolerance=5000\relax%
3660   \@tempdima=\parindent%
3661   \csuse{colalignX@#1}%
3662   \parindent=\@tempdima%
3663   {\hspace{\parindent}%
3664     \csuse{notenumfontX@#1}\wrapped@footfootmarkX{#1}\strut%
3665     \csuse{wrapcontentX@#1}{#2}%
3666     \strut\par}\allowbreak}
3667
3668 %

```

```

\threecolfootgroupX \threecolfootgroupX{\series}
\mpthreecolfootgroupX
3669 \newcommand*\threecolfootgroupX[1]{\csuse{bhookgroupX@#1}\csuse{
notefontsizeX@#1}
3670 \splittopskip=\ht\strutbox
3671 \expandafter
3672 \rigidbalanceX\csname footins#1\endcsname \thr@@ \splittopskip}}
3673
3674 \newcommand*\mpthreecolfootgroupX[1]{\%
3675 \vskip\skip\@nameuse{mpfootins#1}
3676 \ifl@dpairing\ifparledgroup
3677 \leavevmode\marks\parledgroup@{begin}%
3678 \marks\parledgroup@series{#1}%
3679 \marks\parledgroup@type{footnoteX}%
3680 \fi\fi\normalcolor
3681 \ifparledgroup%
3682 \ifl@dpairing%
3683 \else%
3684 \setnoteswidthliketwocolumnsX@{#1}%
3685 \setnotesXpositionliketwocolumns@{#1}%
3686 \print@footnoteXrule{#1}%
3687 \fi%
3688 \else%
3689 \setnoteswidthliketwocolumnsX@{#1}%
3690 \setnotesXpositionliketwocolumns@{#1}%
3691 \print@footnoteXrule{#1}%
3692 \fi%
3693 \csuse{bhookgroupX@#1}%
3694 \splittopskip=\ht\strutbox
3695 \expandafter
3696 \rigidbalanceX\csname mpfootins#1\endcsname \thr@@ \splittopskip}}
3697
3698 \%

```

XIII.4.5 Paraphed footnotes

The following macros set footnotes as one paragraph.

```

\arrangementX@threecol \footparagraphX{\series}
3699 \newcommand*\arrangementX@paragraph[1]{\%
3700 \csgdef{series@displayX#1}{paragraph}%
3701 \expandafter\newcount\csname #1prevpage@num\endcsname
3702 \expandafter\let\csname footstart#1\endcsname=\parafootstartX
3703 \expandafter\let\csname regvfootnote#1\endcsname=\para@vfootnoteX
3704 \expandafter\let\csname footfmt#1\endcsname=\parafootfmtX
3705 \expandafter\let\csname footgroup#1\endcsname=\para@footgroupX
3706 \expandafter\let\csname footnoterule#1\endcsname=\normalfootnoteruleX
3707 \count\csname footins#1\endcsname=1000

```

```

3708 \csxdef{default@footins#1}{1000}%Use this to confine the notes to one
side only
3709 \dimen\csname footins#1\endcsname=\csuse{maxhnotesX@#1}
3710 \skip\csname footins#1\endcsname=\csuse{beforenotesX@#1}%
3711 \advance\skip\csname footins#1\endcsname by\csuse{afterruleX@#1}%
3712 \para@footsetupX{#1}
3713 \ifnoledgroup@else
3714 \expandafter\let\csname mpvfootnote#1\endcsname=\mppara@vfootnoteX
3715 \expandafter\let\csname mpfootgroup#1\endcsname=\mppara@footgroupX
3716 \count\csname mpfootins#1\endcsname=1000
3717 \dimen\csname mpfootins#1\endcsname=\csuse{maxhnotesX@#1}
3718 \skip\csname mpfootins#1\endcsname=\csuse{beforenotesX@#1}%
3719 \advance\skip\csname mpfootins#1\endcsname by\csuse{afterruleX@#1}%
3720 \fi
3721 }
3722
3723 %

```

`\para@footsetupX` `\para@footsetupX{<series>}`

```

3724 \newcommand*{\para@footsetupX}[1]{\csuse{bhookgroupX@#1}\csuse{
notefontsizeX@#1}
3725 \setnoteswidthliketwocolumnsX@{#1}%
3726 \ifcempty{widthX@#1}%
3727 {}%
3728 {\columnwidth=\expandafter\dimexpr\csuse{widthX@#1}\relax}%
3729 \dimen0=\baselineskip
3730 \multiply\dimen0 by 1024
3731 \divide\dimen0 by \columnwidth \multiply\dimen0 by \footfudgefiddle\relax
3732 %
3733 \expandafter
3734 \xdef\csname footfudgefactor#1\endcsname{%
\expandafter\strip@pt\dimen0 }}
3735
3736 %

```

`\parafootstartX` `\parafootstartX{<series>}`

```

3737 \newcommand*{\parafootstartX}[1]{%
3738 \ifdimequal{Opt}{\prenotesX@}{}%
3739 {%
3740 \iftoggle{prenotesX@}{%
3741 \togglefalse{prenotesX@}%
3742 \skip\csname footins#1\endcsname=%
3743 \dimexpr\csuse{prenotesX@}+\csuse{afterruleX@#1}\relax%
3744 }%
3745 {}%
3746 }%
3747 \leftskip=\z@
3748 \rightskip=\z@

```



```

3749 \nottoggle{parindentX@#1}{\parindent=\z@}{}%
3750 \vskip\skip\@nameuse{footins#1}%
3751 \setnoteswidthliketwocolumnsX@{#1}%
3752 \setnotesXpositionliketwocolumns@{#1}%
3753 \print@footnoteXrule{#1}%
3754 }
3755
3756 %

```

```

\para@vfootnoteX \para@vfootnoteX{<series>}{<text>}
\mppara@vfootnoteX
3757 \newcommand*{\para@vfootnoteX}[2]{%
3758 \csuse{beforeinsertingX@#1}%
3759 \insert\csname footins#1\endcsname%
3760 \bgroup
3761 \csuse{notefontsizeX@#1}
3762 \footsplitskips
3763 \setbox0=\vbox{\hsize=\maxdimen%
3764 \let\bid@RTL@everypar\@empty%
3765 \noindent\csuse{bhooknoteX@#1}%
3766 \@nameuse{footfmt#1}{#1}{#2}}%
3767 \setbox0=\hbox{\unvXH{0}{#1}}%
3768 \dp0=\z@
3769 \ht0=\csname footfudgefactor#1\endcsname\wd0
3770 \box0
3771 \penalty0
3772 \egroup}
3773 \newcommand*{\mppara@vfootnoteX}[2]{%
3774 \get@thisfootnoteX{#1}%
3775 \get@fnmarkX{#1}{\thisfootnote}%
3776 \edef\this@footnoteX@reading{\the\csname footnote#1@reading\endcsname}%
3777 \global\setbox\@nameuse{mpfootins#1}\vbox{%
3778 \unvbox\@nameuse{mpfootins#1}
3779 \csuse{notefontsizeX@#1}
3780 \footsplitskips
3781 \setbox0=\vbox{\hsize=\maxdimen%
3782 \let\bid@RTL@everypar\@empty%
3783 \noindent\color@begingroup%
3784 \csuse{bhooknoteX@#1}%
3785 \@nameuse{footfmt#1}{#1}{#2}\color@endgroup}%
3786 \setbox0=\hbox{\unvXH{0}{#1}}%
3787 \dp0=\z@
3788 \ht0=\csname footfudgefactor#1\endcsname\wd0
3789 \box0
3790 \penalty0}}
3791
3792 %

```

```

\unvXH93 \newcommand*{\unvXH}[2]{% 2th is optional for retro-compatibility

```

```

3794 \setbox0=\vbox{\unvbox#1%
3795 \global\setbox1=\lastbox}%
3796 \unhbox1
3797 \unskip          % remove \rightskip,
3798 \unskip          % remove \parfillskip,
3799 \unpenalty       % remove \penalty of 10000,
3800 \hskip\csuse{afternoteX@#2}%
3801 \relax}% but add the glue to go between the notes
3802
3803 %

```

`\parafootfmtX` `\parafootfmtX{<series>}`

```

3804 \newcommand*{\parafootfmtX}[2]{%
3805 \protected@edef\@currentlabel{%
3806 \@nameuse{@thefnmark#1}%
3807 }%
3808 \insertparafootsepX{#1}%
3809 \ledsetnormalparstuff@common%
3810 {\csuse{notenumfontX@#1}%
3811 \csuse{notenumfontX@#1}%
3812 \wrapped@footfootmarkX{#1}%
3813 \strut%
3814 \csuse{wrapcontentX@#1}{#2}%
3815 \penalty-10}}
3816
3817 %

```

`\para@footgroupX` `\para@footgroupX{<series>}`
`\mppara@footgroupX`

```

3818 \newcommand*{\para@footgroupX}[1]{%
3819 \hsize=\expandafter\dimexpr\csuse{widthX@#1}\relax%
3820 \unvbox\csname footins#1\endcsname
3821 \ifcsstring{raggedX@#1}{L}{\RaggedLeft}{}%
3822 \ifcsstring{raggedX@#1}{R}{\RaggedRight}{}%
3823 \makehboxofhboxes
3824 \setbox0=\hbox{\unhbox0 \removehboxes}%
3825 \csuse{bhookgroupX@#1}
3826 \csuse{notefontsizeX@#1}
3827 \unhbox0\par}
3828
3829 \newcommand*{\mppara@footgroupX}[1]{%
3830 \setnoteswidthliketwocolumnsX@{#1}%
3831 \vskip\skip\@nameuse{mpfootins#1}
3832 \ifl@dpairing\ifparledgroup
3833 \leavevmode%
3834 \leavevmode\marks\parledgroup@{begin}%
3835 \marks\parledgroup@series{#1}%
3836 \marks\parledgroup@type{footnoteX}%
3837 \fi\fi\normalcolor

```

```

3838 \ifparledgroup%
3839 \ifl@dpairing%
3840 \else%
3841 \setnoteswidthliketwocolumnsX@{#1}%
3842 \setnotesXpositionliketwocolumns@{#1}%
3843 \print@footnoteXrule{#1}%
3844 \fi%
3845 \else%
3846 \setnoteswidthliketwocolumnsX@{#1}%
3847 \setnotesXpositionliketwocolumns@{#1}%
3848 \print@footnoteXrule{#1}%
3849 \fi%
3850 \unvbox\csname mpfootins#1\endcsname
3851 \ifcsstring{raggedX@#1}{L}{\RaggedLeft}{}%
3852 \ifcsstring{raggedX@#1}{R}{\RaggedRight}{}%
3853 \makehboxofhboxes
3854 \setbox0=\hbox{\unhbox0 \removehboxes}%
3855 \csuse{bhookgroupX@#1}%
3856 \csuse{notefontsizeX@#1}%
3857 \nottoggle{parindentX@#1}{\parindent=\z@}{}%
3858 \unhbox0\par}}
3859
3860 %

```

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

```

\prevpage@num61 \newcommand{\insertparafootsepX}[1]{%
\Xinsertparafootsep62 \ifnumequal{\csuse{prevpage#1@num}}{\page@num}%
3863 {\csuse{parafootsepX@#1}}%
3864 {}%
3865 }
3866 %

```

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.

```

3867 \newcommand{\wrapped@footfootmarkX}[1]{%
3868 \ifdefined\hypertarget%
3869 \hyperlink%
3870 {\bodyfootmark#1@\this@footnoteX@reading}%
3871 {\@nameuse{footfootmark#1}}%
3872 \Hy@raisedlink{%
3873 \hypertarget%
3874 {\footnotemark#1@\this@footnoteX@reading}%
3875 {}%

```

```

3876 }%
3877 \else%
3878   \@nameuse{footfootmark#1}%
3879 \fi%
3880 }%
3881 %

```

`\wrapped@bodyfootmarkX` `\wrapped@bodyfootmarkX` prints the footnote mark of the text body, wrapped in `hyperref` package’s commands, if needed.

```

3882 \newcommand{\wrapped@bodyfootmarkX}[1]{%
3883   \ifdefined\hypertarget%
3884     \hyperlink%
3885       { \@footnotemark#1 \@expandafter\the\csname footnote#1@reading\
endcsname}%
3886       { \@nameuse{bodyfootmark#1}}}%
3887   \Hy@raisedlink{%
3888     \hypertarget%
3889       { \@bodyfootmark#1 \@expandafter\the\csname footnote#1@reading\
endcsname}%
3890     {}%
3891   }%
3892 \else%
3893   \@nameuse{bodyfootmark#1}%
3894 \fi%
3895 }%
3896 %

```

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@common97 \newcommand*{\ledsetnormalparstuff@common}{%
\Xledsetnormalparstuff98   \ifluatex%
\ledsetnormalparstuffX99   \textdir\footnote@luatextextdir%
3900   \pardir\footnote@luatexpardir%
3901   \fi%
3902   \csuse{\csuse{footnote@dir}}}%
3903   \normal@pars%

```

```

3904 \parfillskip \z@ \@plus 1fil}%
3905
3906 \newcommand*{\Xledsetnormalparstuff}[1]{%
3907 \ledsetnormalparstuff@common%
3908 \nottoggle{Xparindent@#1}{\parindent=\z@}{\hspace{\parindent}}}%
3909 }%
3910
3911 \newcommand*{\ledsetnormalparstuffX}[1]{%
3912 \ledsetnormalparstuff@common%
3913 \nottoggle{parindentX@#1}{\parindent=\z@}{\hspace{\parindent}}}%
3914 }%
3915 %

```

XV Footnotes' width for two columns

We define here some commands which make sense only with `reledpar`, but must be called when defining notes parameters. These commands change the width of block notes to allow them to have the same size than two parallel columns.

`\old@hsize` These two commands are called at the beginning of critical or familiar notes groups. They set, if the option is enabled, the `\hsize`. They are also called at the on the setup for paragraphed notes.

```

3916
3917 \newdimen\old@hsize%
3918 \AtBeginDocument{\old@hsize=\hsize}%
3919
3920 \newcommand{\setXnoteswidthliketwocolumns@}[1]{%
3921 \global\let\hsize@fornote=\hsize%
3922 \global\old@hsize=\hsize%
3923 \let\old@columnwidth=\columnwidth%
3924 \iftoggle{Xnoteswidthliketwocolumns@#1}%
3925 {%
3926 \csuse{setwidthliketwocolumns@\columns@position}%
3927 \global\let\hsize@fornote=\hsize%
3928 }%
3929 }%
3930 \let\hsize=\hsize@fornote%
3931 \let\columnwidth=\old@columnwidth%
3932 }%
3933
3934 \newcommand{\setnoteswidthliketwocolumnsX@}[1]{%
3935 \global\let\hsize@fornote=\hsize%
3936 \global\old@hsize=\hsize%
3937 \let\old@columnwidth=\columnwidth%
3938 \iftoggle{noteswidthliketwocolumnsX@#1}%
3939 {%
3940 \csuse{setwidthliketwocolumns@\columns@position}%
3941 \global\let\hsize@fornote=\hsize%

```

```

3942 }%
3943 {}%
3944 \let\hsize=\hsize@fornote%
3945 \let\columnwidth=\old@columnwidth%
3946 }%
3947
3948 %

```

`\setXnotespositionliketwocolumns@` These two commands set the position of the critical / familiar footnotes, depending on the hooks `Xnoteswidthliketwocolumns` and `noteswidthliketwocolumnsX`. They call commands which are defined only in `reledpar`, because this feature has no sens without `reledpar`.

```

3949 \newcommand{\setXnotespositionliketwocolumns@}[1]{%
3950   \iftoggle{Xnoteswidthliketwocolumns@#1}{%
3951     \csuse{setnotespositionliketwocolumns@\columns@position}%
3952   }{}%
3953 }%
3954
3955 \newcommand{\setnotesXpositionliketwocolumns@}[1]{%
3956   \iftoggle{noteswidthliketwocolumnsX@#1}{%
3957     \csuse{setnotespositionliketwocolumns@\columns@position}%
3958   }{}%
3959 }%
3960
3961 %

```

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
3962 \def\@fnpos{familiar-critical}
3963 \def\@mpfnpos{critical-familiar}
3964 \newcommand{\fnpos}[1]{\xdef\@fnpos{#1}}
3965 \newcommand{\mpfnpos}[1]{\xdef\@mpfnpos{#1}}
3966 %

```

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`.

```

\printXfootnoterule67 \newcommand{\printXfootnoterule}[1]{%
\print@footnotexrule68   \vskip-\csuse{Xafterrule@#1}%Because count in \dimen\csuse{#1footins}
3969   \nointerlineskip%

```

```

3970 \moveleft-\leftskip\ vbox{\csuse{#1footnoterule}}}%
3971 \nointerlineskip%
3972 \vskip\csuse{Xafterrule@#1}}%
3973 }%
3974
3975 \newcommand{\print@footnoteXrule}[1]{%
3976 \vskip-\csuse{afterruleX@#1}}%Because count in \dimen\csuse{footins#1}
3977 \nointerlineskip%
3978 \moveleft-\leftskip\ vbox{\csuse{footnoterule#1}}}%
3979 \nointerlineskip%
3980 \vskip\csuse{afterruleX@#1}}%
3981 }%
3982
3983 %

```

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.

```

3984 \newtoggle{preXnotes@}
3985 \toggletrue{preXnotes@}
3986 \newcommand{\preXnotes@}{Opt}
3987 \newcommand*{\preXnotes}[1]{\renewcommand{\preXnotes@}{#1}}
3988 %

```

The same, but for familiar footnotes.

```

\preXnotes@ \newtoggle{prenotesX@}
\preXnotes@ \toggletrue{prenotesX@}
3991 \newcommand{\prenotesX@}{Opt}
3992 \newcommand*{\prenotesX}[1]{\renewcommand{\prenotesX@}{#1}}
3993 %

```

XVIII.0.3 Internal commands

```

firstXseries@ \gdef\firstXseries@{}
prepare@preXnotes \newcommand{\prepare@preXnotes}[1]{%
3996 \ifdimequal{Opt}{\preXnotes@}%
3997 }%
3998 {%
3999 \IfStrEq{\firstXseries@}{-}{%
4000 \global\skip\csuse{#1footins}=\preXnotes@%
4001 \global\advance\skip\csname #1footins\endcsname by\csuse{Xafterterrule@
#1}%
4002 \gdef\firstXseries@{#1}%
4003 }%
4004 {%
4005 \ifseriesbefore{#1}{\firstXseries@}%
4006 {%
4007 \global\skip\csuse{#1footins}=\csuse{Xbeforenotes@\firstXseries@}%
4008 \global\advance\skip\csname #1footins\endcsname by\csuse{Xafterterrule@
#1}%
4009 \gdef\firstXseries@{#1}%
4010 }%
4011 }%
4012 }%
4013 }%
4014 }
4015 %

```

The same thing is required for familiar notes and `\prenotesX`.


```

firstseriesX@16 \gdef\firstseriesX@{}
prepare@prenotesX17 \newcommand{\prepare@prenotesX}[1]{%
4018 \ifdimequal{0pt}{\prenotesX@}%
4019 {}%
4020 {}%
4021 \IfStrEq{\firstseriesX@}{-}{%
4022 \global\skip\csuse{footins#1}=\prenotesX@%
4023 \global\advance\skip\csname footins#1\endcsname by\csuse{afterruleX@
#1}%
4024 \gdef\firstseriesX@{#1}%
4025 }%
4026 {}%
4027 \ifseriesbefore{#1}{\firstseriesX@}%
4028 {}%
4029 \global\skip\csuse{footins#1}=\csuse{beforenotesX@\firstseriesX@}%
4030 \global\advance\skip\csname footins#1\endcsname by\csuse{afterruleX@
#1}%
4031 \gdef\firstXseries@{#1}%
4032 }%
4033 {}%
4034 }%
4035 }%
4036 }
4037 %

```

XIX Endnotes

First, check the noend option.

```

4038 \ifbool{noend@}{-}{%Used instead of \ifnoend@ to prevent expansion problem
4039 %

```

`\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.

```

4040 \newcommand{\l@dend@open}[1]{%
4041 \global\booltrue{l@dend@#1}%
4042 \expandafter\immediate%
4043 \expandafter\openout%
4044 \csname l@d@#1end\endcsname%
4045 =\jobname.#1end\relax%
4046 }%
4047 \newcommand{\l@dend@close}[1]{%
4048 \global\boolfalse{l@dend@#1}%
4049 \expandafter\immediate%

```

```

4050 \expandafter\closeout\csname l@d@#1end\endcsname%
4051 }%
4052
4053 %

```

\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.

```

4054 \newcommand{\l@dend@stuff}{%
4055 \def\do##1{%
4056 \ifbool{l@dend@##1}{}%
4057 {\l@dend@open{##1}}%
4058 \expandafter\immediate\expandafter\write\csname l@d@##1end\endcsname{\
string\l@d@section{\the\section@num}}}%
4059 }%
4060 \dolistloop{@series}%
4061 }%
4062
4063 %

```

\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.

```

4064 \global\newcommand{\l@d@section#1#2#3#4#5#6#7}{%
4065 \csuse{Xendhooknote@#4}%
4066 \csuse{Xendnotefontsize@#4}%
4067 \hangindent=\csuse{Xendhangindent@#4}%
4068 \ifXendinsertsep%
4069 \hskip\csuse{Xendafternote@#4}\relax%
4070 \csuse{Xendsep@#4}%
4071 \else%
4072 \iftoggle{Xendparagraph@#4}%

```

```

4073     {\global\Xendinsertsep@true}%
4074     }%
4075 \fi%
4076 \xdef\@currentseries{#4}%
4077 \def\do##1{%
4078     \setkeys[mac]{truefootnoteoption}{##1}%
4079 }%
4080 \notblank{#5}{\docsvlist{#5}}{}%
4081 \IfStrEq{#6}{R}{\ledRcol@true}{}%
4082 \def\@this@crossref@start{#7:start}%
4083 \def\@this@crossref@end{#7:end}%
4084 \printlineendnote{#1}{#4}%
4085 \IfStrEq{#6}{R}{\ledRcol@false}{}%
4086 \undef\@this@crossref@start%
4087 \undef\@this@crossref@end%
4088 \nottoggle{Xendlemmadisablefontselection@#4}%
4089     {\select@lemmafont#1|}%
4090     {}%
4091 \bgroup%
4092     \csuse{Xendlemmafont@#4}%
4093     \csuse{Xendwraplemma@#4}{#2}%
4094 \egroup%
4095 \ifbool{expr}%
4096     togl {nosep@}%
4097     or test{\ifcempty{Xendlemmaseparator@#4}}%
4098     }%
4099     {\hskip\csuse{Xendinplaceoflemmaseparator@#4}\relax}%
4100     {\nobreak%
4101         \hskip\csuse{Xendbeforelemmaseparator@#4}%
4102         \csuse{Xendlemmaseparator@#4}%
4103         \hskip\csuse{Xendafterlemmaseparator@#4}%
4104         \relax%
4105     }%
4106 \csuse{Xendwrapcontent@#4}{#3}%
4107 \nottoggle{Xendparagraph@#4}{\par}{}%
4108 \def\do##1{%
4109     \setkeys[mac]{falsefootnoteoption}{##1}%
4110     }%
4111 \notblank{#5}{\docsvlist{#5}}{}%
4112 }}%
4113
4114 \let\l@d@section=\@gobble
4115
4116 %

```

\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.

```

4117 \newcommand{\printlineendnote}[2]{%

```

```

4118 \l@dp@rsefootspec#1|{%
4119 \iftoggle{Xendnumberonlyfirstintwolines@#2}{%
4120 \edef\lineinfo@{\l@d@p@rsestartpage - \l@d@p@rsestartline - \
l@d@p@rsestartsub - \l@d@p@rseendpage - \l@d@p@rseendline - \
l@d@p@rseendsub}%
4121 }%
4122 {%
4123 \edef\lineinfo@{\l@d@p@rsestartpage - \l@d@p@rsestartline - \
l@d@p@rsestartsub}%
4124 }%
4125 \ifbool{expr}{%
4126 togl {nonum@}%
4127 or togl {Xendnonumber@#2}%
4128 }%
4129 {\hspace{\csuse{Xendinplaceofnumber@#2}}}%
4130 {%
4131 \iftoggle{Xendnumberonlyfirstinline@#2}%
4132 {\ifcsdef{prevendline#2}%
4133 {\ifcsequal{prevendline#2}{\lineinfo@}%
4134 {%
4135 \csuse{Xendbhookinplaceofnumber@#2}%
4136 \ifcsequal{Xendsymmlinenumber@#2}%
4137 {\hspace{\csuse{Xendinplaceofnumber@#2}}}%
4138 {\printsymmlineendnotearea{#2}}}%
4139 \csuse{Xendahookinplaceofnumber@#2}%
4140 }%
4141 {\printlineendnotearea{#1}{#2}}}%
4142 {\printlineendnotearea{#1}{#2}}}%
4143 }%
4144 {\printlineendnotearea{#1}{#2}}%We keep every time line
4145 \csxdef{prevendline#2}{\lineinfo@}%
4146 }%
4147 }%
4148 %

```

```

\printsymmlineendnotearea49 \newcommand{\printsymmlineendnotearea}[1]{%
4150 \hspace{\csuse{Xendbeforemlinenumber@#1}}%
4151 \csuse{Xendnotenumfont@#1}%
4152 \ifdimequal{\csuse{Xendboxsymmlinenumber@#1}}{\z@}%
4153 {\csuse{Xendsymmlinenumber@#1}}%
4154 {\hbox to \csuse{Xendboxsymmlinenumber@#1}%
4155 {\csuse{Xendsymmlinenumber@#1}\hfill}%
4156 }%
4157 \hspace{\csuse{Xendaftersymmlinenumber@#1}}%
4158 }%
4159 %

```

`\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.)

```

4160 \newcommand{\printlineendnotearea}[2]{%
4161   \csuse{Xendbhooklinenumber@#2}%
4162   \hspace{\csuse{Xendbeforenumber@#2}}%
4163   \bgroup%
4164     \csuse{Xendnotenumfont@#2}%
4165     \ifdimequal{\csuse{Xendboxlinenum@#2}}{0pt}%
4166       {\printendlines#1||\ifledRcol@\@Rlineflag\fi}%
4167       {\leavevmode%
4168         \hbox to \csuse{Xendboxlinenum@#2}%
4169         {%
4170           \IfSubStr{RC}{\csuse{Xendboxlinenumalign@#2}}{\hfill}{}}%
4171           \printendlines#1||\ifledRcol@\@Rlineflag\fi%
4172           \IfSubStr{LC}{\csuse{Xendboxlinenumalign@#2}}{\hfill}{}}%
4173         }%
4174     \egroup%
4175     \hspace{\csuse{Xendafternumber@#2}}%
4176     \csuse{Xendahooklinenumber@#2}%
4177   }%
4178   %

```

\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.

```

4179 \newif\ifXendinsertsep@%
4180 \newcommand*{\doendnotes}[1]{%
4181   \l@ndend@close{#1}%
4182   \begingroup
4183     \makeatletter
4184     \expandafter\let\csname #1end\endcsname=\endprint
4185     \input\jobname.#1end%
4186     \global\Xendinsertsep@false%
4187   \endgroup}
4188 %

```

\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.

```

4189 \newcommand*{\doendnotesbysection}[1]{%
4190   \l@ndend@close{#1}%
4191   \global\expandafter\advance\csname #1end@bysection\endcsname by 1%
4192   \begingroup%
4193     \makeatletter%
4194     \def\l@d@section##1{%

```

```

4195     \ifnumequal{##1}{\csname #1end@bysection\endcsname}%
4196     {\cslet{#1end}{\endprint}}}%
4197     {\cslet{#1end}{\@gobblefive}}}%
4198 }%
4199 \input\jobname.#1end%
4200 \global\Xendinsertsep@false%
4201 \endgroup%
4202 }%
4203 %

```

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.

```

4204 }%
4205 %

```

`\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.

```

4206 \newcommand*{\setprintendlines}[6]{%
4207   \l@dpnumfalse \l@ddashfalse
4208   \ifnum#4=#1 \else
4209     \l@dpnumtrue
4210     \l@ddashtrue
4211   \fi
4212 %

```

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.

```

4213   \ifl@dpnum \l@delintrue \else \l@delinfalse \fi
4214   \ifnum#2=#5 \else
4215     \l@delintrue
4216     \l@ddashtrue
4217   \fi
4218 %

```

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

```

4219   \l@ssubfalse
4220   \ifnum#3=0 \else
4221     \l@ssubtrue
4222   \fi
4223 %

```

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.

```

4224 \l@d@eslfalse
4225 \ifnum#6=0 \else
4226   \ifnum#6=#3
4227     \ifl@d@elin \l@d@esltrue \else \l@d@eslfalse \fi
4228   \else
4229     \l@d@esltrue
4230     \l@d@dashtrue
4231   \fi
4232 \fi%
4233 %

4234 \ifl@d@dash%
4235   \ifboolexpr{togl{fulllines@} or test{\ifcsemt{Xendtwolines@}\@currentseries}}}%
4236   {}%
4237   {%
4238   \setistwofollowinglines{#1}{#2}{#4}{#5}%
4239   \ifboolexpr{%
4240     (%
4241       togl {Xendtwolinesbutnotmore@\@currentseries}%
4242       and not%
4243       (%
4244         bool {istwofollowinglines@}%
4245       )%
4246     )%
4247   or%
4248   (%
4249     (not test{\ifnumequal{#1}{#4}})%
4250     and togl{Xendtwolinesonlyinsamepage@\@currentseries}%
4251   )%
4252   }%
4253   {}%
4254   {%
4255     \l@d@dashfalse%
4256     \l@d@Xtwolinestrue%
4257     \l@d@elinfalse%
4258     \l@d@eslfalse%
4259     \ifcsemt{Xendmorethantwolines@\@currentseries}%
4260     {}%
4261     {\ifistwofollowinglines@\else%
4262       \l@d@Xmorethantwolinestrue%
4263     \fi%
4264   }%
4265   }%
4266 }%
4267 \fi%
4268 %

```

End of `\setprintendlines`.

```
4269 }%
4270 %
```

`\printendlines` Now we are ready to print it all.

```
4271 \def\printendlines#1|#2|#3|#4|#5|#6|#7|#8|{\begingroup
4272   \setprintendlines{#1}{#2}{#3}{#4}{#5}{#6}%
4273   %
```

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.

```
4274   \ifdimequal{\csuse{Xendboxstartlinenum@\@currentseries}}{0pt}%
4275   {\bgroup}%
4276   {\leavevmode\hbox to \csuse{Xendboxstartlinenum@\@currentseries}\bgroup
\hfill}%
4277   \wrap@edcrossref{\@this@crossref@start}{\printnpnum{#1}}%
4278   \ifl@d@dash%
4279   \ifl@d@pnum%
4280   \csuse{Xendlineprefixsingle@\@currentseries}%
4281   \else%
4282   \ifcempty{Xendlineprefixmore@\@currentseries}%
4283   {\csuse{Xendlineprefixsingle@\@currentseries}}
4284   {\csuse{Xendlineprefixmore@\@currentseries}}%
4285   \fi%
4286   \else%
4287   \csuse{Xendlineprefixsingle@\@currentseries}%
4288   \fi%
4289   \wrap@edcrossref{\@this@crossref@start}{\linenumrep{#2}}%
4290   \iftoggle{Xendlineflag@\@currentseries}{\ifledRcol@\@Rlineflag\fi}{}%
4291   \ifl@d@ssub%
4292   \csuse{Xendsublinesep@\@currentseries}%
4293   \wrap@edcrossref{\@this@crossref@start}{\sublinenumrep{#3}}%
4294   \fi%
4295   \egroup%
4296   %
```

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

```
4297   \ifdimequal{\csuse{Xendboxendlinenum@\@currentseries}}{0pt}%
4298   {\bgroup}%
4299   {\hbox to \csuse{Xendboxendlinenum@\@currentseries}\bgroup}%
4300   \ifl@d@Xtwoines%
4301   \ifl@d@Xmorethantwoines%
4302   \csuse{Xendmorethantwoines@\@currentseries}%
4303   \else%
4304   \csuse{Xendtwoines@\@currentseries}%
```



```

4305 \fi%
4306 \else%
4307 \ifl@d@dash%
4308 \ifdefined\linrangesep@%
4309 \linrangesep@%
4310 \else%
4311 \csuse{Xendlinrangeseparator@}\@currentseries}%
4312 \fi%
4313 \fi%
4314 \ifl@d@pnum%
4315 \wrap@edcrossref{\@this@crossref@end}\printnpnum{#4}%
4316 \fi%
4317 \ifl@d@elin%
4318 \ifl@d@pnum\csuse{Xendlineprefixsingle@}\@currentseries}\fi%
4319 \wrap@edcrossref{\@this@crossref@end}{\linenumrep{#5}}%
4320 \iftoggle{Xendlineflag@}\@currentseries}{\ifledRcol@\@Rlineflag\fi}{}%
4321 \fi%
4322 \ifl@d@esl%
4323 \ifl@d@elin%
4324 \csuse{Xendsublinesep@}\@currentseries}%
4325 \fi%
4326 \wrap@edcrossref{\@this@crossref@end}{\sublinenumrep{#6}}%
4327 \fi%
4328 \fi%
4329 \ifdimequal{\csuse{Xendboxendlinenum@}\@currentseries}{0pt}%
4330 {}%
4331 {\hfill}%Prevent underfull hbox
4332 \egroup%
4333 \endgroup%
4334 }%
4335 %
4336 %

```

\printnpnum A macro to print a page number in an endnote. Should not be override anymore

```

4337 \newcommand*{\printnpnum}[1]{\csuse{Xendbeforepagenumber@}\@currentseries
4338 }#1\csuse{Xendafterpagenumber@}\@currentseries}}
4339 %

```

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@.

```

4340 \newcommand{\newseries}[1]{%
4341 \def\do##1{\newseries@{##1}}%

```

```

4342 \docsvlist{#1}
4343 }
4344 %

```

\@series The `\series@` macro is an etoolbox list, which contains the name of all series.

```

4345 \newcommand{\@series}{}
4346 %

```

The command `\newseries@``\series` creates a new series of the footnote.

```

\newseries@47 \newcommand{\newseries@}[1]{
4348 %

```

XX.1 Test if series is still existing

```

4349 \xifinlist{#1}{\@series}{\led@warn@SeriesStillExist{#1}}%
4350 {%
4351 %

```

XX.2 Init specific to reledpar

When calling `\newseries@` after having loaded `reledpar`, we need to load specific setting.

```

4352 \ifdefined\newseries@par%
4353 \newseries@par{#1}%
4354 \fi%
4355 %

```

XX.3 For critical footnotes

Critical footnotes are those which start with letters. We look for the `\nocritical` option of `reledmac`.

```

4356 \unless\ifnocritical@
4357 %

```

XX.3.1 Options

```

4358 \newtoggle{Xlineflag@#1}
4359 \newtoggle{Xparindent@#1}
4360 \newtoggle{Xlemmadisablefontselection@#1}
4361 \csgdef{Xwrapcontent@#1}{}%
4362 \csgdef{Xbeforeinserting@#1}{}%
4363 \csgdef{Xhangindent@#1}{Opt}%
4364 \csgdef{Xragged@#1}{}%

```

```

4365 \csgdef{Xhsizetwocol@#1}{0.45 \hsize}%
4366 \csgdef{Xhsizethreecol@#1}{.3 \hsize}%
4367 \csgdef{Xcolalign@#1}{\raggedright}%
4368 \csgdef{Xnotenumfont@#1}{\normalfont}%
4369 \csgdef{Xnotefontsize@#1}{\footnotesize}%
4370 \csgdef{Xbhooknote@#1}{}%
4371 \csgdef{Xbhookgroup@#1}{}%
4372
4373 \csgdef{Xboxlinenum@#1}{Opt}%
4374 \csgdef{Xboxlinenumalign@#1}{L}%
4375
4376 \csgdef{Xboxstartlinenum@#1}{Opt}%
4377 \csgdef{Xboxendlinenum@#1}{Opt}%
4378
4379 \csgdef{Xboxsymlinenum@#1}{Opt}%
4380 \newtoggle{Xnumberonlyfirstinline@#1}%
4381 \newtoggle{Xnumberonlyfirstintwolines@#1}%
4382 \csgdef{Xtwolines@#1}{}%
4383 \csgdef{Xmorethantwolines@#1}{}%
4384 \csgdef{Xsublinesep@#1}{\fullstop}%
4385 \newtoggle{Xtwolinesbutnotmore@#1}%
4386 \newtoggle{Xtwolinesonlyinsamepage@#1}%
4387 \newtoggle{Xonlypstart@#1}%
4388 \newtoggle{Xpstarteverytime@#1}%
4389 \newtoggle{Xpstart@#1}%
4390 \newtoggle{Xstanza@#1}%
4391 \csgdef{Xstanzaseparator@#1}{}%
4392 \csgdef{Xsymlinenum@#1}{}%
4393 \newtoggle{Xnonumber@#1}%
4394 \csgdef{Xbeforenumber@#1}{Opt}%
4395 \csgdef{Xtxtbeforenumber@#1}{}%
4396 \csgdef{Xafternumber@#1}{0.5em}%
4397 \newtoggle{Xnonbreakableafternumber@#1}%
4398 \csgdef{Xbeforesymlinenum@#1}{\csuse{Xbeforenumber@#1}}%
4399 \csgdef{Xaftersymlinenum@#1}{\csuse{Xafternumber@#1}}%
4400 \csgdef{Xinplaceofnumber@#1}{1em}%
4401 \global\cslet{Xlemmaseparator@#1}{\rbracket}%
4402 \csgdef{Xbeforelemmaseparator@#1}{0em}%
4403 \csgdef{Xafterlemmaseparator@#1}{0.5em}%
4404 \csgdef{Xinplaceoflemmaseparator@#1}{1em}%
4405 \csgdef{Xbeforenotes@#1}{1.2em \@plus .6em \@minus .6em}%
4406 \csgdef{Xafterrule@#1}{Opt}%
4407 \csgdef{Xtxtbeforenotes@#1}{}%
4408 \csgdef{Xmaxhnotes@#1}{0.8\vsizel}%
4409 \newtoggle{Xnoteswidthliketwocolumns@#1}%
4410 \csgdef{Xparafootsep@#1}{}%
4411 \csgdef{Xafternote@#1}{1em plus.4em minus.4em}%
4412 \csgdef{Xlinerrangeseparator@#1}{\endashchar}%
4413
4414 \csgdef{Xlemmafont@#1}{}%

```

```

4415 \csgdef{Xwraplemma@#1}{%
4416 \csgdef{Xwidth@#1}{\hsize}%
4417 %

```

XX.3.2 Create inserts, needed to add notes in foot

As regards inserts, see chapter 15 of *The TeXbook* by D. Knuth.

```

4418 \expandafter\newinsert\csname #1footins\endcsname%
4419 \unless\ifnoledgroup%
4420 \expandafter\newinsert\csname mp#1footins\endcsname%
4421 \fi%
4422 %

```

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.

```

4423 \global\newbool{parapparatus@}{\expandafter\newcommand\expandafter
*}{\expandafter\newcommand}\csname #1footnote\endcsname[2][]{%
4424 \ifnum\@edtext@level>0%
4425 \begingroup%
4426 \newcommand{\content}{##2}%
4427 \ifnumberedpar%
4428 \ifledRcol%
4429 \ifluatex%
4430 \footnotelang@lua[R]%
4431 \fi%
4432 \@ifundefined{xpg@main@language}{%if polyglossia
4433 {}%
4434 {\footnotelang@poly[R]}%
4435 \footnoteoptions@{R}{##1}{true}%
4436 \xright@appenditem{%
4437 \ifbool{indtl@innote}%
4438 {\unexpanded{\let\index\nindex}}%
4439 {}%
4440 \ifbool{indtl@notenumber}%
4441 {\unexpanded{\let\index\nindex}}%There is no note
...number so
4442 {}%
4443 \noexpand\Xnote@true%
4444 \noexpand\prepare@preXnotes{#1}%
4445 \noexpand\prepare@edindex@fornote{\ld@nums}%
4446 \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.
4447 \noexpand\setcounter{stanzaR}{\the\c@stanzaR}%Save
stanzaR counter for footnote

```

```

4448 \unexpanded{\def\@this@crossref@start}{\theedtext:
start}%
4449 \unexpanded{\def\@this@crossref@end}{\theedtext:end}%
4450 \noexpand\csuse{v#1footnote}{#1}%
4451 {\l@d@nums}{\expandonce\@tag}{\expandonce\content}}
%
4452 \noexpand\Xnote@false%
4453 \unexpanded{\undef\@this@crossref@start}%
4454 \unexpanded{\undef\@this@crossref@end}%
4455 \ifbool{indtl@innote}%
4456 {\unexpanded{\let\index\orig@@index}}%
4457 {}%
4458 \ifbool{indtl@notenumber}%
4459 {\unexpanded{\let\index\orig@@index}}%
4460 {}%
4461 }\to\inserts@listR
4462 \footnoteoptions@{R}{##1}{false}%
4463 \global\advance\insert@countR \@ne%
4464 \else%
4465 \ifluatex%
4466 \footnotelang@lua%
4467 \fi%
4468 \@ifundefined{xpg@main@language}%if polyglossia
4469 {}%
4470 {\footnotelang@poly}%
4471 \footnoteoptions@{L}{##1}{true}%
4472 \xright@appenditem{%
4473 \ifbool{indtl@innote}%
4474 {\unexpanded{\let\index\nindex}}%
4475 {}%
4476 \ifbool{indtl@notenumber}%
4477 {\unexpanded{\let\index\nindex}}%There is no note
...number so
4478 {}%
4479 \noexpand\Xnote@true%
4480 \noexpand\prepare@preXnotes{#1}%
4481 \noexpand\prepare@edindex@fornote{\l@d@nums}%
4482 \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.
4483 \ifl@dpairing%
4484 \noexpand\setcounter{stanzaL}{\the\c@stanzaL}%Save
stanzaR counter for footnote
4485 \fi%
4486 \unexpanded{\def\@this@crossref@start}{\theedtext:
start}%
4487 \unexpanded{\def\@this@crossref@end}{\theedtext:end}%
4488 \noexpand\csuse{v#1footnote}%
4489 {#1}%

```

```

4490         {{\l@d@nums}{\expandonce\@tag}{\expandonce\content
}}%
4491         \unexpanded{\undef\@this@crossref@start}%
4492         \unexpanded{\undef\@this@crossref@end}%
4493         \noexpand\Xnote@false%
4494         \ifbool{indtl@innote}%
4495         { \unexpanded{\let\index\orig@@index}}%
4496         {}%
4497         \ifbool{indtl@notenumber}%
4498         { \unexpanded{\let\index\orig@@index}}%
4499         {}%
4500         }\to\inserts@list
4501         \global\advance\insert@count \@ne%
4502         \footnoteoptions@{L}{\#1}{false}%
4503         \fi
4504     \else
4505         \csuse{v#1footnote}{\#1}{\{0|0|0|0|0|0|0\}{\}{\#1}}%
4506         \fi%
4507     \endgroup%
4508 \else%
4509     \led@err@FootnoteWithoutEdtext%
4510 \fi%
4511 \ignorespaces%
4512 }
4513 %

```

We need to be able to modify reledmac's footnote macros and restore their

```

4514     \global\csletcs{\#1@footnote}{\#1footnote}
4515 %

```

XX.3.4 Set standard display

```

4516     \Xarrangement@normal{\#1}%
4517 %

```

End of for critical footnotes.

```

4518     \fi
4519 %

```

XX.4 For familiar footnotes

Familiar footnotes are those which end with letters. We look for the `nofamiliar` option of `reledmac`.

```

4520     \unless\ifnofamiliar@
4521 %

```

XX.4.1 Options

```

4522 \newtoggle{parindentX@#1}
4523 \csgdef{wrapcontentX@#1}{}%
4524 \csgdef{hangindentX@#1}{Opt}%
4525 \csgdef{beforeinsertingX@#1}{}%
4526 \csgdef{raggedX@#1}{}%
4527 \csgdef{hsizetwocolX@#1}{0.45 \hsize}%
4528 \csgdef{hsizethreecolX@#1}{.3 \hsize}%
4529 \csgdef{colalignX@#1}{\raggedright}%
4530 \csgdef{notenumfontX@#1}{\normalfont}%
4531 \csgdef{notefontsizeX@#1}{\footnotesize}%
4532 \csgdef{bhooknoteX@#1}{}%
4533 \csgdef{bhookgroupX@#1}{}%
4534 \csgdef{afterruleX@#1}{Opt}
4535 \csgdef{beforenotesX@#1}{1.2em \@plus .6em \@minus .6em}
4536 \csgdef{maxhnotesX@#1}{0.8\vsizex}%
4537 \newtoggle{noteswidthliketwocolumnsX@#1}%
4538 \csgdef{parafootsepX@#1}{}%
4539 \csgdef{afternoteX@#1}{1em plus.4em minus.4em}
4540 \csgdef{widthX@#1}{\hsizex}%
4541 % End of for familiar footnotes.
4542 % \subsubsection{Create inserts, needed to add notes in foot}
4543 % As regards inserts, see chapter 15 of the TeXBook by D. Knuth.
4544 % \begin{macrocode}
4545 \expandafter\newinsert\csname footins#1\endcsname%
4546 \unless\ifnoledgroup%
4547 \expandafter\newinsert\csname mpfootins#1\endcsname%
4548 \fi%
4549 %

```

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.

```

4550 \global\expandafter\newcommand\csname footnote#1\endcsname[1]{%
4551 \begingroup%
4552 \prepare@prenotesX{#1}%
4553 \newcommand{\content}{##1}%
4554 %
4555 %

```

If we are preparing parallel typesetting, we cannot just increase the footnote counter. Read reledpar's handbook about that (V.2.1 p. 45).

```

4556 \global\expandafter\advance\csname footnote#1@reading\
4557 endcsname by \@ne%
4558 \ifl@dpairing%
4559 \ifcsdef{footnote#1reading\the\csname footnote#1@reading\
4560 endcsname=typeset}%

```

```

4559         {\setcounter{footnote#1}{\csuse{footnote#1reading\the\
csname footnote#1@reading\endcsname=typeset}}}%
4560         {\setcounter{footnote#1}{\the\csname footnote#1@reading
\endcsname}}}%
4561     \else%
4562         \stepcounter{footnote#1}%
4563     \fi%
4564 %

```

And now, the feature not depending of whether we are preparing parallel typesetting

```

4565     \protected@csxdef{@thefnmark#1}{\csuse{thefootnote#1}}%
4566     \nottoggle{nomk@}%Nomk is set to true when using \
footnoteXnomk with \parpackage
4567     {\csuse{@footnotemark#1}}%
4568     {}%
4569     \ifluatex%
4570         \xdef\footnote@luatextextdir{\the\textdir}%
4571         \xdef\footnote@luatexpardir{\the\pardir}%
4572     \fi%
4573     \if@ledgroup%
4574         \led@set@index@fornote{#1}%
4575     \fi%
4576     \csuse{vfootnote#1}{#1}{\expandonce\content}\m@mmf@prepare%
4577     \ifbool{indtl@innote}%
4578         {\let\index\orig@@index}%
4579         {}%
4580     \ifbool{indtl@notenumber}%
4581         {\let\index\orig@@index}%
4582         {}%
4583     \endgroup%
4584 }
4585 %

```

Then define the counters. The \LaTeX counter `footnoteX` is the only one manipulated by the user. This is this the one which is printed. The \TeX counter `\footnoteX@reading` is increased at each footnote. It is used for hyperlinks, for using `hyperlink` package, and for getting the correct footnote number when using parallel typesetting (V.2.1 p. 45).

```

4586     \newcounter{footnote#1}
4587     \global\expandafter\renewcommand\csname thefootnote#1\endcsname{\
arabic{footnote#1}}
4588     \expandafter\newcount\csname footnote#1@reading\endcsname%
4589 %

```

Do not forget to initialize series

```

4590     \arrangementX@normal{#1}%
4591 \fi
4592 %

```


XX.5 The endnotes

Endnotes are commands like `\Xendnote`, where X is a series letter. First, we check for the `noend` options.

```
4593 \unless\ifnoend@
4594 %
```

XX.5.1 The auxiliary file

Endnotes of all varieties are saved up in a file, one by series, typically named `⟨jobname⟩.Xend`. `\l@d@Xend` is the output stream number for this file, and `\ifl@dend@X` is a flag that is true when the file is open.

```

\l@d@Xend
\ifl@dend@X
\l@dend@Xtrue
\l@dend@Xfalse
4595 \expandafter\newwrite\csname l@d@#1end\endcsname%
4596 \expandafter\newif\csname ifl@dend@#1\endcsname%
4597 %
```

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.

```

4598
4599 \global\expandafter\newcommandx\csname #1endnote\endcsname[2][1,
usedefault]{%
4600 \bgroup%
4601 \newlinechar='40%
4602 \global\@noneed@Footnotetrue%
4603 \newcommand{\content}{##2}%
4604 \stepcounter{labidx}%
4605 \expandafter\immediate\expandafter\write\csname l@d@#1end\
endcsname{%
4606 \expandafter\string\csname #1end\endcsname%
4607 {\ifnumberedpar@l@d@nums\fi}%
4608 {\ifnumberedpar@\expandonce\@tag\fi}%
4609 {\expandonce\content}%
4610 {\#1}%
4611 {\unexpanded{##1}}%
4612 {\ifledRcol R\else L\fi}%
4613 {\theedtext}%
4614 \@percentchar%
4615 }%
4616 \egroup%
4617 \ignorespaces%
4618 }%
4619 %
```

`\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.

```
4620
4621 \global\cslet{#1end}{\@gobblefive}
4622 %
```

We need to store the number of times `\doendnotesbysection` is called for one series.

```
4623 \global\expandafter\newcount\csname #1endbysection\endcsname%
4624 %
```

XX.5.3 The options

```
4625 \csgdef{Xendwraplemma@#1}{%
4626 \csgdef{Xendwrapcontent@#1}{}%
4627 \csgdef{Xendtwolines@#1}{}%
4628 \csgdef{Xendmorethantwolines@#1}{}%
4629 \newtoggle{Xendtwolinesbutnotmore@#1}{}%
4630 \newtoggle{Xendtwolinesonlyinsamepage@#1}{}%
4631 \newtoggle{Xendlemmadisablefontselection@#1}{}%
4632 \csgdef{Xendnotenumfont@#1}{\normalfont}%
4633 \csgdef{Xendnotefontsize@#1}{\footnotesize}%
4634 \csgdef{Xendbhooknote@#1}{}%
4635
4636 \csgdef{Xendsublinesep@#1}{\fullstop}%
4637
4638 \csgdef{Xendbeforenumber@#1}{Opt}
4639 \csgdef{Xendafternumber@#1}{0.5em}
4640
4641 \csgdef{Xendboxlinenum@#1}{Opt}%
4642 \csgdef{Xendboxlinenumalign@#1}{L}%
4643
4644 \csgdef{Xendboxstartlinenum@#1}{Opt}%
4645 \csgdef{Xendboxendlinenum@#1}{Opt}%
4646
4647 \csgdef{Xendlemmaseparator@#1}{}%
4648 \csgdef{Xendbeforelemmaseparator@#1}{0em}%
4649 \csgdef{Xendafterlemmaseparator@#1}{0.5em}%
4650 \csgdef{Xendinplaceoflemmaseparator@#1}{0.5em}%
4651
4652 \newtoggle{Xendparagraph@#1}%
4653 \csgdef{Xendafternote@#1}{1em plus.4em minus.4em}%
4654 \csgdef{Xendsep@#1}{}%
4655
4656 \csgdef{Xendinplaceofnumber@#1}{Opt}%
4657 \newtoggle{Xendnonumber@#1}%
```

```

4658 \csgdef{Xendhangindent@#1}{Opt}%
4659 \newtoggle{Xendnumberonlyfirstinline@#1}%
4660 \newtoggle{Xendnumberonlyfirstintwolines@#1}%
4661
4662 \csgdef{Xendbeforesymmlinenumber@#1}{\csuse{Xendbeforenumber@#1}}%
4663 \csgdef{Xendaftersymmlinenumber@#1}{\csuse{Xendafternumber@#1}}%
4664 \csgdef{Xendsymmlinenumber@#1}{}%
4665 \csgdef{Xendboxsymmlinenumber@#1}{Opt}%
4666
4667 \csgdef{Xendbhooklinenumber@#1}{}%
4668 \csgdef{Xendehooklinenumber@#1}{}%
4669 \csgdef{Xendbhookinplaceofnumber@#1}{}%
4670 \csgdef{Xendehookinplaceofnumber@#1}{}%
4671
4672 \csgdef{Xendlinangeseparator@#1}{\endashchar}%
4673
4674 \csgdef{Xendbeforepagenumber@#1}{p.}%
4675 \csgdef{Xendafterpagenumber@#1}{ }%
4676 \csgdef{Xendlineprefixsingle@#1}{}%
4677 \csgdef{Xendlineprefixmore@#1}{}%
4678
4679 \newtoggle{Xendlineflag@#1}
4680
4681 \csgdef{Xendlemmafont@#1}{}%
4682 %
4683 %

```

End of endnotes declaration

```

4684 \fi%
4685 %

```

Dump series in \@series

```

4686 \listxadd{\@series}{#1}
4687 }
4688 }% End of \newseries
4689 %

```

XX.6 Init standards series (A,B,C,D,E)

```

4690 \expandafter\newseries\expandafter{\default@series}
4691 %

```

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.

```

4692 \newcommand{\seriesatbegin}[1]{%
4693   \StrDel{\@series}{#1}[\@series]%
4694   \edef\@new{%
4695     \listadd{\@new}{#1}%
4696     \listadd{\@new}{\@series}%
4697   \xdef\@series{\@new}%
4698 }
4699 %

```

\seriesatend And **\seriesatend** moves the series to the end of the list.

```

4700 \newcommand{\seriesatend}[1]{%
4701   \StrDel{\@series}{#1}[\@series]%
4702   \edef\@new{%
4703     \listadd{\@new}{\@series}%
4704     \listadd{\@new}{#1}%
4705   \xdef\@series{\@new}%
4706 }
4707 %

```

XXI.2 Test series order

\ifseriesbefore **\ifseriesbefore**{*<seriesA>*}{*<seriesB>*}{*<true>*}{*<false>*} expands *<true>* if *<seriesA>* is printed before *<seriesB>*, expands *<false>* otherwise.

```

4708 \newcommand{\ifseriesbefore}[4]{%
4709   \StrPosition{\@series}{#1}[\@first]%
4710   \StrPosition{\@series}{#2}[\@second]%
4711   \ifnumgreater{\@second}{\@first}{#3}{#4}%
4712 }
4713 %

```

XXI.2.1 Get the first series

In some specific case, we need to know the first series of the list of series.

```

\@getfirstseries14 \newcommand{\@getfirstseries}{%
4715   \ifdefempty{\@series}%
4716   {\xdef\@firstseries{}}%
4717   {\StrChar{\@series}{1}[\@firstseries]}%
4718 }%
4719 %

```

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`.

```

4720 \newcommandx{\settoggle@series}[5][4,5,usedefault]{%
4721   \def\do##1{%
4722     \global\settoggle{#2@##1}{#3}%
4723     \ifstrequal{#4}{critical}{
4724       \csuse{Xarrangement@}\csuse{series@display##1}}{##1}%
4725     }{}
4726     \ifstrequal{#4}{familiar}{
4727       \csuse{arrangementX@}\csuse{series@displayX##1}}{##1}%
4728     }{}
4729   }%
4730   \ifstreempty{#1}{%
4731     \dolistloop{\@series}%
4732     \ifstreempty{#5}{}%
4733       \docsvlist{#5}%
4734     }
4735   }%

```

```

4736         {%
4737         \docsvlist{#1}%
4738         }%
4739     }
4740     %

```

`\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.

```

4741 \newcommandx{\setcommand@series}[5][4,5,usedefault]{%
4742     \def\do##1{
4743         \csgdef{#2@##1}{#3}
4744         \ifstrequal{#4}{critical}{%
4745             \csuse{Xarrangement@\csuse{series@display##1}}{##1}%
4746         }{
4747             \ifstrequal{#4}{familiar}{%
4748                 \csuse{arrangementX@\csuse{series@displayX##1}}{##1}%
4749             }{
4750             }%
4751         }%
4752         \ifstreempty{#1}{%
4753             \dolistloop{\@series}%
4754             \ifstreempty{#5}{}%
4755             \docsvlist{#5}
4756         }%
4757         {%
4758             \docsvlist{#1}%
4759         }%
4760     }%
4761     %

```

XXI.3.3 Tools to generate options commands

`\newhookcommand@series` `\newhookcommand@series\command` names is a generic command to add new commands for hooks, like `\Xhsizetwocol`. The first argument is the name of the hook, the second a comma-separated list of pseudo-series where the hook can be used, like

appref in the case of `\Xtwolines`. The second argument is also used to create commands named `\<hookname><pseudoserries>`, like `\Xtwolinesappref`.

```

4762 \newcommandx{\newhookcommand@series}[2][2,usedefault]{%
4763   \global\expandafter\newcommand\expandafter*\csname #1\endcsname[2][]{%
4764     \setcommand@series{##1}{#1}{##2}[][#2]%
4765   }%
4766   \ifstrempy{#2}{}{%
4767     \def\do##1{%
4768       \global\expandafter\newcommand\expandafter*\csname #1##1\endcsname
4769       [1]{%
4770         \csuse{#1}[##1]{####1}%
4771       }%
4772     }%
4773     \docsvlist{#2}%
4774   }%
4775 }%

```

`\newhooktoggle@series` `\newhooktoggle@series\command names` is a generic command to add new commands for a new toggle hook, like `\Xnumberonlyfirstinline`. The second argument is also used to create commands named `\<hookname><pseudoserries>`, like `\Xtwolinesbutnotmoreappref`.

```

4776 \newcommandx{\newhooktoggle@series}[2][2,usedefault]{%
4777   \global\expandafter\newcommandx\expandafter*\csname #1\endcsname[2][1,2={
4778     true},usedefault]{%
4779     \settoggle@series{##1}{#1}{##2}[][#2]%
4780   }%
4781   \ifstrempy{#2}{}{%
4782     \def\do##1{%
4783       \global\expandafter\newcommand\expandafter*\csname #1##1\endcsname{%
4784         \csuse{#1}[##1]%
4785       }%
4786     }%
4787     \docsvlist{#2}%
4788   }%
4789 }%

```

`\newhookcommand@toggle@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 or notes

```

4790 \newcommand{\newhooktoggle@series@reload}[2]{%
4791   \global\expandafter\newcommandx\expandafter*\csname #1\endcsname[2][1,2={
4792     true},usedefault]{%
4793     \settoggle@series{##1}{#1}{##2}[#2]%
4794   }%
4795 }%

```

`\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.

```

4796 \newcommand{\newhookcommand@series@reload}[2]{%
4797   \global\expandafter\newcommand\expandafter*\csname #1\endcsname[2][{}]{%
4798     \setcommand@series{##1}{#1}{##2}[#2]%
4799   }%
4800 }
4801 %

```

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.

```

4802 \unless\ifnocritical@
4803   \newhookcommand@series{Xwrapcontent}%
4804   \newhookcommand@series{Xbeforeinserting}%
4805   \newhookcommand@series{Xlemmafont}%
4806   \newhookcommand@series{Xwraplemma}%
4807   \newhooktoggle@series{Xparindent}
4808   \newhookcommand@series{Xhangindent}
4809   \newhookcommand@series{Xragged}
4810   \newhookcommand@series{Xhsizetwocol}
4811   \newhookcommand@series{Xhsizethreecol}
4812   \newhookcommand@series{Xcolalign}%
4813   \newhookcommand@series{Xnotenumfont}
4814   \newhookcommand@series{Xbhooknote}
4815   \newhookcommand@series@reload{Xbhookgroup}{critical}
4816   \newhookcommand@series{Xboxsymlinenum}%
4817   \newhookcommand@series{Xsymlinenum}
4818   \newhookcommand@series{Xbeforenumber}
4819   \newhookcommand@series{Xtxtbeforenumber}
4820   \newhookcommand@series{Xafternumber}
4821   \newhookcommand@series{Xbeforesymlinenum}
4822   \newhookcommand@series{Xaftersymlinenum}
4823   \newhookcommand@series{Xinplaceofnumber}
4824   \newhookcommand@series{Xlemmaseparator}
4825   \newhookcommand@series{Xbeforelemmaseparator}
4826   \newhookcommand@series{Xafterlemmaseparator}
4827   \newhookcommand@series{Xinplaceoflemmaseparator}
4828   \newhookcommand@series{Xtxtbeforenotes}
4829   \newhookcommand@series@reload{Xafterrule}{critical}
4830   \newhooktoggle@series{Xnumberonlyfirstinline}
4831   \newhooktoggle@series{Xnumberonlyfirstintwolines}
4832   \newhooktoggle@series{Xnonumber}
4833   \newhooktoggle@series{Xpstart}
4834   \newhooktoggle@series{Xpstarteverytime}%
4835

```



```

4836 \newhooktoggle@series{Xstanza}%
4837 \newhookcommand@series{Xstanzaseparator}%
4838
4839 \newhooktoggle@series{Xonlypstart}
4840 \newhooktoggle@series{Xnonbreakableafternumber}
4841 \newhooktoggle@series{Xlemmadisablefontselection}
4842 \newhookcommand@series@reload{Xmaxhnotes}{critical}
4843 \newhookcommand@series@reload{Xbeforenotes}{critical}
4844 \newhooktoggle@series@reload{Xnoteswidthliketwocolumns}{critical}%
4845 \newhookcommand@series@reload{Xnotefontsize}{critical}
4846
4847 \newhookcommand@series{Xboxlinenum}%
4848 \newhookcommand@series{Xboxlinenumalign}%
4849
4850 \newhookcommand@series{Xboxstartlinenum}%
4851 \newhookcommand@series{Xboxendlinenum}%
4852
4853 \newhookcommand@series{Xafternote}%
4854 \newhookcommand@series{Xparafootsep}
4855
4856 \newhookcommand@series@reload{Xwidth}{critical}%
4857
4858 \ifundef{\Xhsize}%
4859 {%
4860   \newcommandx{\Xhsize}[2][1,usedefault]{%
4861     \led@warning@Xhsize@deprecated%
4862     \Xwidth[#1]{#2}%
4863   }%
4864 }%
4865 {}%
4866 \fi
4867 \newhooktoggle@series{Xlineflag}[appref,SEref]
4868 \newhookcommand@series{Xtwolines}[appref,SEref]
4869 \newhookcommand@series{Xmorethantwolines}[appref,SEref]
4870 \newhookcommand@series{Xsublinesep}[appref,SEref,side]
4871 \newhooktoggle@series{Xtwolinesbutnotmore}[appref,SEref]
4872 \newhooktoggle@series{Xtwolinesonlyinsamepage}[appref,SEref]
4873 \newhookcommand@series{Xlinrangeseparator}[appref,SEref]
4874 %

```

XXI.3.5 Options for familiar notes

Before generating the optional commands for familiar notes, we check the `\nofamiliar` option.

```

4875 \unless\ifnofamiliar@
4876   \newhookcommand@series{wrapcontentX}%
4877   \newhookcommand@series{beforeinsertingX}%
4878   \newhooktoggle@series{parindentX}
4879   \newhookcommand@series{hangindentX}

```

```

4880 \newhookcommand@series{raggedX}
4881 \newhookcommand@series{hsizetwocolX}
4882 \newhookcommand@series{hsizethreecolX}
4883 \newhookcommand@series{colalignX}%
4884 \newhookcommand@series{notenumfontX}
4885 \newhookcommand@series{bhooknoteX}
4886 \newhookcommand@series@reload{bhookgroupX}{familiar}
4887 \newhookcommand@series@reload{beforenotesX}{familiar}
4888 \newhookcommand@series@reload{maxhnotesX}{familiar}
4889 \newhooktoggle@series@reload{noteswidthliketwocolumnsX}{familiar}%
4890 \newhookcommand@series@reload{afterruleX}{familiar}
4891 \newhookcommand@series@reload{notefontsizeX}{familiar}
4892 \newhookcommand@series{afternoteX}
4893 \newhookcommand@series{parafootsepX}
4894 \newhookcommand@series@reload{widthX}{familiar}%
4895 \ifundef{\hsizeX}%
4896 {
4897   \newcommandx{\hsizeX}[2][1,usedefault]{
4898     \led@warning@hsizeX@deprecated%
4899     \widthX[#1]{#2}%
4900   }%
4901 }%
4902 {}%
4903 \fi
4904 %

```

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.

```

4905 \unless\ifnoend@
4906 \newhookcommand@series{Xendwraplemma}
4907 \newhookcommand@series{Xendwrapcontent}
4908 \newhookcommand@series{Xendnotenumfont}
4909 \newhookcommand@series{Xendlemmafont}%
4910 \newhookcommand@series{Xendbhooknote}
4911
4912 \newhookcommand@series{Xendboxlinenum}%
4913 \newhookcommand@series{Xendboxlinenumalign}%
4914
4915 \newhookcommand@series{Xendboxstartlinenum}%
4916 \newhookcommand@series{Xendboxendlinenum}%
4917
4918 \newhookcommand@series{Xendnotefontsize}
4919 \newhooktoggle@series{Xendlemmadisablefontselection}
4920 \newhookcommand@series{Xendlemmaseparator}
4921 \newhookcommand@series{Xendbeforelemmaseparator}
4922 \newhookcommand@series{Xendafterlemmaseparator}
4923 \newhookcommand@series{Xendinplaceoflemmaseparator}

```

```

4924 \newhookcommand@series{Xendbeforenumber}%
4925 \newhookcommand@series{Xendafternumber}%
4926
4927
4928 \newhooktoggle@series{Xendparagraph}
4929 \newhookcommand@series{Xendafternote}
4930 \newhookcommand@series{Xendsep}
4931
4932 \newhookcommand@series{Xendinplaceofnumber}%
4933 \newhooktoggle@series{Xendnonumber}%
4934
4935 \newhooktoggle@series{Xendnumberonlyfirstinline}%
4936 \newhooktoggle@series{Xendnumberonlyfirstintwolines}%
4937
4938 \newhookcommand@series{Xendsymmlinenum}%
4939 \newhookcommand@series{Xendbeforesymmlinenum}%
4940 \newhookcommand@series{Xendaftersymmlinenum}%
4941 \newhookcommand@series{Xendboxsymmlinenum}%
4942
4943 \newhookcommand@series{Xendbhooklinenumber}%
4944 \newhookcommand@series{Xendahooklinenumber}%
4945 \newhookcommand@series{Xendbhookinplaceofnumber}%
4946 \newhookcommand@series{Xendahookinplaceofnumber}%
4947
4948 \newhookcommand@series{Xendhangindent}%
4949
4950
4951 \fi
4952 \newhooktoggle@series{Xendlineflag}[apprefwithpage,Serefwithpage]
4953 \newhookcommand@series{Xendtwolines}[apprefwithpage,Serefwithpage]
4954 \newhookcommand@series{Xendmorethantwolines}[apprefwithpage,Serefwithpage]
4955 \newhooktoggle@series{Xendtwolinesbutnotmore}[apprefwithpage,Serefwithpage]
4956 \newhooktoggle@series{Xendtwolinesonlyinsamepage}[apprefwithpage,
4957 Serefwithpage]
4958 \newhookcommand@series{Xendlinerangeseparator}[apprefwithpage,Serefwithpage]
4959 ]
4960 \newhookcommand@series{Xendbeforepagenumber}[apprefwithpage,Serefwithpage,
4961 Serefonlypage]
4962 \newhookcommand@series{Xendafterpagenumber}[apprefwithpage,Serefwithpage]
4963 \newhookcommand@series{Xendlineprefixsingle}[apprefwithpage,Serefwithpage]
4964 \newhookcommand@series{Xendlineprefixmore}[apprefwithpage,Serefwithpage]
4965 \newhookcommand@series{Xendsublinesep}[apprefwithpage,Serefwithpage]
4966
4967 %

```

XXI.4 Hooks for a particular footnote

`\newhooktoggle@specific` `\newhooktoggle@specific` is a generic command to create boolean hook specific to a note.

```

4965 \newcommand{\newhooktoggle@specific}[1]{%
4966   \newtoggle{#1}%
4967   \define@key[mac]{truefootnoteoption}{#1}[]{\global\settoggle{#1}{true}}%
When enabling footnote option
4968   \define@key[mac]{falsefootnoteoption}{#1}[]{\global\settoggle{#1}{false}
}}
4969 }
4970 %

```

\newhookarg@specific \newhookarg@specific is a generic command to create argumen hook specific to a note.

```

4971 \newcommand{\newhookarg@specific}[1]{%
4972   \define@key[mac]{truefootnoteoption}{#1}{\global\def\linrangesep@{##1}}%
When enabling footnote option
4973   \define@key[mac]{falsefootnoteoption}{#1}{\global\undef\linrangesep@}%
When
4974 }
4975 %

```

And now, we define some hooks specific to a note.

```

4976 \newhooktoggle@specific{fulllines}%
4977 \newhooktoggle@specific{nonum}
4978 \newhooktoggle@specific{nosep}
4979 \newhookarg@specific{linrangesep}
4980 %

```

linrangesep@ \linrangesep@ is defined by the option `linrangesep` of critical notes to change temporarily the line range separator for a specific line. As we have to define it before typesetting the line and undefine it after, we use the family of `xkeyval` package's key.

```

4981 %

```

\nomk@ \nomk@ toggle is used by `reledpar` to remove the footnote mark in the text when using `\footnoteXmk`. Read `reledpar` handbook.

```

4982 \newtoggle{nomk@}%
4983 %

```

XXI.5 Alias

\Xnolemmaseparator \Xnolemmaseparator[*<series>*] is just an alias for `\Xlemmaseparator[<series>]{}`.

```

4984 \newcommandx*{\Xnolemmaseparator}[1][1]{\Xlemmaseparator[#1]{}}
4985 %

```

XXII Output routine

Now we begin the output routine and associated things.

XXII.0.1 Page number management

`\pageno` `\advancepageno` `\pageno` is a page number, starting at 1, and `\advancepageno` increments the number.

```

4986 \countdef\pageno=0 \pageno=1
4987 \newcommand*{\advancepageno}{\ifnum\pageno<\z@ \global\advance\pageno\m@ne
4988   \else\global\advance\pageno\@ne\fi}
4989
4990 %

```

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

`\doextrafeet` `\doextrafeet` 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.

```

4991 \newcommand*{\l@ddextrafeet}{%
4992   \IfStrEq{familiar-critical}{\@fnpos}
4993     {\do@feetX\Xdo@feet}%
4994     {%
4995       \IfStrEq{critical-familiar}{\@fnpos}%
4996       {\Xdo@feet\do@feetX}%
4997       {\do@feetX\Xdo@feet}%
4998     }%
4999 }%
5000
5001 %

```

`\Xdo@feet` `\Xdo@feet` is the code extending `\@makecol` to cater for the extra critical feet.

```

5002 \newcommand*{\Xdo@feet}{%
5003   \setbox\@outputbox \vbox{%
5004     \unvbox\@outputbox
5005     \@opXfeet}}
5006 %

```

`\@opXfeet` `\print@Xnotes` The extra critical feet to be added to the output. The normal way to add one series, `\print@Xnotes`, is replaced by `reledpar` when using `\Pages`.

```

5007 \newcommand\print@Xnotes[1]{%
5008   \xdef\@currentseries{#1}%
5009   \csuse{#1footstart}{#1}%
5010   \csuse{#1footgroup}{#1}%
5011 }%
5012 %

```

We print all series of notes by looping on them. We check before printing them that they are not voided.

```

5013 \newcommand*\@opXfeet{%
5014   \unless\ifnocritical@%
5015   \gdef\firstXseries@{%
5016     \def\do##1{%
5017       \ifvoid\csuse{##1footins}\else%
5018         \global\skip\csuse{##1footins}=\csuse{Xbeforenotes@##1}%
5019         \global\advance\skip\csuse{##1footins} by\csuse{Xafterrule@##1}%
5020         \print@Xnotes{##1}%
5021       \fi%
5022     }%
5023     \dolistloop{\@series}%
5024   \fi%
5025 }%
5026 %

```

\l@ddodoreintrafeet \l@ddodoreintrafeet is the code for catering for the extra footnotes within \@reinserts. We use the same category and ordering as in \l@ddoxtrafeet.

```

5027 \newcommand*\l@ddodoreintrafeet{%
5028   \IfStrEq{familiar-critical}{\@fnpos}
5029   {\@doreinfeetX\X@doreinfeet}%
5030   {%
5031     \IfStrEq{critical-familiar}{\@fnpos}%
5032     {\X@doreinfeet\@doreinfeetX}%
5033     {\@doreinfeetX\X@doreinfeet}%
5034   }%
5035 }
5036
5037 %

```

\X@doreinfeet \X@doreinfeet is the code for catering for the extra critical footnotes within \@reinserts.

```

5038 \newcommand*\X@doreinfeet{%
5039   \unless\ifnocritical@%
5040   \def\do##1{%
5041     \ifvoid\csuse{##1footins}\else%
5042       \insert\csuse{##1footins}{\unvbox\csuse{##1footins}}%
5043     \fi%
5044   \dolistloop{\@series}
5045   \fi%
5046 }

```

5047
5048 %

`\print@notesX` We have to add all the new kinds of familiar footnotes to the output routine. The normal way to add one series. `\print@Xnotes` is replaced by `reledpar` when using `\Pages`.

`\do@feetX`
`\@doreinfeetX`
5049 `\newcommand\print@notesX[1]{%`
5050 `\xdef\@currentseries{#1}%`
5051 `\csuse{footstart#1}{#1}%`
5052 `\csuse{footgroup#1}{#1}%`
5053 `}%`
5054 `%`

We print all the series of notes by looping on them. We check before printing them that they are not voided.

5055 `\newcommand*{\do@feetX}{%`
5056 `\unless\ifnofamiliar%`
5057 `\gdef\firstseriesX@{}`
5058 `\setbox\@outputbox \vbox{%`
5059 `\unvbox\@outputbox%`
5060 `\def\do##1{%`
5061 `\ifvoid\csuse{footins##1}\else%`
5062 `\global\skip\csuse{footins##1}=\csuse{beforenotesX@##1}%`
5063 `\global\advance\skip\csuse{footins##1} by\csuse{afterterruleX@##1}%`
5064 `\print@notesX{##1}%`
5065 `\fi%`
5066 `}%`
5067 `\dolistloop{\@series}}%`
5068 `\fi%`
5069 `}%`
5070
5071 `\newcommand{\@doreinfeetX}{%`
5072 `\unless\ifnofamiliar%`
5073 `\def\do##1{%`
5074 `\ifvoid\csuse{footins##1}\else`
5075 `\insert%`
5076 `\csuse{footins##1}`
5077 `{\unvbox\csuse{footins##1}}%`
5078 `\fi%`
5079 `}%`
5080 `\dolistloop{\@series}}%`
5081 `\fi%`
5082 `}%`
5083
5084 `%`

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.)

```

5085 \ifclassloaded{memoir}{%
5086 %

memoir is loaded so we use memoir's built in hooks.

5087 \g@addto@macro{\m@mdoextrafeet}{\l@ddoxtrafeet}%
5088 \g@addto@macro{\m@mdodoreinextrafeet}{\l@ddodoreinextrafeet}%
5089 }{%
5090 %

memoir has not been loaded, so patch \@makecol and \@reinserts.

5091 \@ifpackageloaded{fancyhdr}{%
5092   \patchcmd%
5093     {\latex@makecol}%
5094     {\xdef\@freelist{\@freelist\@midlist}}%
5095     {\xdef\@freelist{\@freelist\@midlist}\l@ddoxtrafeet}%
5096     {}%
5097     {\led@error@fail@patch@@makecol}%
5098   }{%
5099     \patchcmd%
5100       {\@makecol}%
5101       {\xdef\@freelist{\@freelist\@midlist}}%
5102       {\xdef\@freelist{\@freelist\@midlist}\l@ddoxtrafeet}%
5103       {}%
5104       {\led@error@fail@patch@@makecol}%
5105     }%
5106
5107   \patchcmd%
5108     {\@reinserts}%
5109     {\ifvbox}%
5110     {\l@ddodoreinextrafeet\ifvbox}%
5111     {}%
5112     {\led@error@fail@patch@@reinserts}%
5113   }
5114
5115 %

```

It turns out that `\@doclearpage` also needs modifying.

`\if@led@nofoot` We have to check if there are any leftover feet.

```

5116 \newif\if@led@nofoot
5117
5118 %

5119 \@ifclassloaded{memoir}{%
5120 %

```

If the memoir class is loaded we hook into its modified `\@doclearpage`.


```

\@mem@extranofeet21 \g@addto@macro{\@mem@extranofeet}{%%
5122 \def\do#1{%
5123 \unless\ifnocritical@%
5124 \ifvoid\csuse{#1footins}\else\@mem@nofootfalse\fi%
5125 \fi%
5126 \unless\ifnofamiliar@%
5127 \ifvoid\csuse{footins#1}\else\@mem@nofootfalse\fi%
5128 \fi%
5129 }
5130 \dolistloop{\@series}%
5131 }%
5132 }{%
5133 %

```

As memoir is not loaded we have patch \@doclearpage.

```

\@led@testifnofoot34 \newcommand*{\@led@testifnofoot}{%
\@doclearpage35 \@led@nofoottrue%
5136 \ifvoid\footins\else%
5137 \@led@nofootfalse%
5138 \fi%
5139 \def\do##1{%
5140 \unless\ifnocritical@%
5141 \ifvoid\csuse{##1footins}\else%
5142 \@led@nofootfalse%
5143 \fi%
5144 \fi%
5145 \unless\ifnofamiliar@%
5146 \ifvoid\csuse{footins##1}\else%
5147 \@led@nofootfalse%
5148 \fi%
5149 \fi%
5150 }%
5151 \dolistloop{\@series}%
5152 }%
5153
5154 \pretocmd%
5155 {\@doclearpage}%
5156 {\@led@testifnofoot}%
5157 {}%
5158 {\led@error@fail@patch@@doclearpage}%
5159
5160 \patchcmd%
5161 {\@doclearpage}%
5162 {\ifvoid\footins}%
5163 {\if@led@nofoot}%
5164 {}%
5165 {\led@error@fail@patch@@doclearpage}%
5166

```

```
5167 }
5168
5169 %
```

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.

```
5170 \list@create{\labelref@list}
5171 %
```

\zz@@@ A convenience macro to zero two labeling counters in one go.

```
5172 \newcommand*{\zz@@@}{000|000} % set two counters to zero in one go
5173
5174 %
```

\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.

```
5175 \newcommand*{\edlabel}[1]{%
5176   \ifl@dpairing\ifautopar%
5177     \strut%
5178     \fi\fi%
5179     \@bsphack%
5180     \ifboolexpr{bool{ledRcol} or bool{ledRcol@}}{%
```

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.

```

5181 \ifXnote@%
5182 \protected@write\@auxout{%
5183   {\string\l@dmake@labelsR\space\thepage|\l@dparsedstartline|\
l@dparsedstartsub|\the\c@pstartR|{#1}}}%
5184 \ifdef{\hypertarget}%
5185   {\Hy@raisedlink{\hypertarget{#1}{}}}%
5186   {}%
5187 \else%
5188   \write\linenum@outR{\string\@lab}%
5189   \ifx\labelref@listR\empty%
5190     \xdef\label@refs{\zz@@@}%
5191   \else%
5192     \gl@p\labelref@listR\to\label@refs%
5193   \fi%
5194   \ifvmode%
5195     \advancelabel@refs%
5196   \fi%
5197 %

```

Use code from the kernel `\label` command to write the correct page number. Also define an `hypertarget` if `hyperref` package is loaded.

```

5198 \protected@write\@auxout{%
5199   {\string\l@dmake@labelsR\space\thepage|\label@refs|\the\c@pstartR
|{#1}}}%
5200 \ifdef{\hypertarget}%
5201   {\Hy@raisedlink{\hypertarget{#1}{}}}%
5202   {}%
5203 \fi%
5204 }{%
5205 \ifXnote@%
5206 \protected@write\@auxout{%
5207   {\string\l@dmake@labelsR\space\thepage|\l@dparsedstartline|\
l@dparsedstartsub|\the\c@pstartR|{#1}}}%
5208 \ifdef{\hypertarget}%
5209   {\Hy@raisedlink{\hypertarget{#1}{}}}%
5210   {}%
5211 \else%
5212   \write\linenum@out{\string\@lab}%
5213   \ifx\labelref@list\empty%
5214     \xdef\label@refs{\zz@@@}%
5215   \else%
5216     \gl@p\labelref@list\to\label@refs%
5217   \fi%
5218   \ifvmode%
5219     \advancelabel@refs%
5220   \fi%
5221 \protected@write\@auxout{%
5222   {\string\l@dmake@labelsR\space\thepage|\label@refs|\the\c@pstart
|{#1}}}%
5223 \ifdef{\hypertarget}%

```

```

5224         {\Hy@raisedlink{\hypertarget{#1}{}}}%
5225         }%
5226     \fi%
5227 }%
5228 \@esphack}%
5229
5230 %

```

`\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.

```

5231 \newcounter{line}%
5232 \newcounter{subline}%
5233 \newcommand{\advancelabel@refs}{%
5234     \setcounter{line}{\expandafter\labelrefsparseline\label@refs}%
5235     \stepcounter{line}%
5236     \ifsublines%
5237         \setcounter{subline}{\expandafter\labelrefsparsesubline\label@refs}%
5238     %
5239     \stepcounter{subline}{1}%
5240     \def\label@refs{\theline|\thesubline}%
5241 \else%
5242     \def\label@refs{\theline|0}%
5243 \fi%
5244 }
5245 \def\labelrefsparseline#1|#2{#1}
5246 \def\labelrefsparsesubline#1|#2{#2}
5247 %

```

`\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.

```

5247 \newcommand*{\l@dmake@labels}{%
5248 \def\l@dmake@labels#1|#2|#3|#4|#5{%
5249     \expandafter\ifx\csname the@label\csuse{XR@prefix}#5\endcsname \relax\
5250 else
5251     \led@warn@DuplicateLabel{\csuse{XR@prefix}#5}%
5252     \fi
5253     \expandafter\gdef\csname the@label\csuse{XR@prefix}#5\endcsname
5254     {#1|#2|#3|#4|\relax}%
5255     \ignorespaces}

```

```

5254
5255 %

```

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.

```

5256 \AtBeginDocument{%
5257   \def\l@make@labels#1|#2|#3|#4|#5{%
5258 }
5259
5260 %

```

\@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.

```

5261
5262 \newcommand*{\@lab}{%
5263   \ifledRcol
5264     \xright@appenditem{\linenumr@p{\line@numR}}|{%
5265       \ifsublines@ \sublinenumr@p{\subline@numR}\else 0\fi}%
5266     \to\labelref@listR
5267   \else
5268     \xright@appenditem{\linenumr@p{\line@num}}|{%
5269       \ifsublines@ \sublinenumr@p{\subline@num}\else 0\fi}%
5270     \to\labelref@list
5271   \fi}
5272 %

```

\applabel \applabel, if called in \edtext will insert automatically both a start and an end label for the current edtext lines.

```

5273 \newcommand*{\applabel}[1]{%
5274   \ifnum\@edtext@level>0%
5275 %

```

Label should not be already defined.

```

5276   \ifcsundef{the@label#1}{%
5277     \csdef{the@label#1}{\applabel}%
5278   }%
5279   {%
5280     \led@warn@DuplicateLabel{#1 (\applabel)}%
5281   }%
5282 %

```

Parse the `\edtext` line numbers.

```
5283 \expandafter\l@dp@rsefootsspec\l@d@nums|%
5284 %
```

Use the \TeX standard hack for label.

```
5285 \@bsphack%
5286 %
```

And now, write the data in the auxiliary file.

```
5287 \ifledRcol%
5288 \protected@write\@auxout{}%
5289 {\string\l@dmake@labelsR\space\l@dparsedstartpage|\l@dparsedstartline|\l@dparsedstartsub|\the\c@pstartR|{#1:start}}%
5290 \ifdef\hypertarget{%
5291 {\Hy@raisedlink{\hypertarget{#1:start}}}}%
5292 {}%
5293 \protected@write\@auxout{}%
5294 {\string\l@dmake@labelsR\space\l@dparsedendpage|\l@dparsedendline|\l@dparsedendsub|\the\c@pstartR|{#1:end}}%
5295 \else%
5296 \protected@write\@auxout{}%
5297 {\string\l@dmake@labels\space\l@dparsedstartpage|\l@dparsedstartline|\l@dparsedstartsub|\the\c@pstart|{#1:start}}%
5298 \ifdef\hypertarget{%
5299 {\Hy@raisedlink{\hypertarget{#1:start}}}}%
5300 {}%
5301 \protected@write\@auxout{}%
5302 {\string\l@dmake@labels\space\l@dparsedendpage|\l@dparsedendline|\l@dparsedendsub|\the\c@pstart|{#1:end}}%
5303 \fi%
5304 %
```

Use the \TeX standard hack for label.

```
5305 \@esphack%
5306 %
```

Warning if `\applabel` is called outside of `\edtext`.

```
5307 \else%
5308 \led@warn@AppLabelOutEdtext{#1}%
5309 \fi%
5310 %
```

End of `\applabel`

```
5311 }%
5312 %
```

`\edlabels` `\edlabels` and `\edlabelE` are just used to mark the beginning and the end of a passage.
`\edlabelE`
`\edlabelSE`

```

5313 \newcommand{\edlabelS}[1]{%
5314   \edlabel{#1:start}%
5315 }
5316 \newcommand{\edlabelE}[1]{%
5317   \edlabel{#1:end}%
5318 }
5319 \newcommand{\edlabelSE}[1]{%
5320   \edlabelS{#1}%
5321   \edlabelE{#1}%
5322 }
5323 %

```

\wrap@edcrossref \wrap@edcrossref is called around all reledmac crossref commands, except those which start with x. It adds the hyperlink.

```

5324 \newrobustcmd{\wrap@edcrossref}[2]{%
5325   \ifdef{\hyperlink}%
5326     {\hyperlink{#1}{#2}}%
5327     {#2}%
5328 }
5329 %

```

\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.

```

5330 \newcommand*{\edpageref}[1]{\l@def@undefined{#1}\wrap@edcrossref{#1}{\l@
5331   dgetref@num{1}{#1}}}
5332 \newcommand*{\xpageref}[1]{\l@dgetref@num{1}{#1}}
5333 %

```

\edlineref If the specified label exists, \lineref gives its line number.

\xlineref

```

5334 \newcommand*{\edlineref}[1]{\l@def@undefined{#1}\wrap@edcrossref{#1}{\l@
5335   dgetref@num{2}{#1}\xflagref{#1}}}%
5336 \newcommand*{\xlineref}[1]{\l@dgetref@num{2}{#1}}%
5337 %

```

\sublineref If the specified label exists, \sublineref gives its sub-line number.

\xsublineref

```

5338 \newcommand*{\sublineref}[1]{\l@def@undefined{#1}\wrap@edcrossref{#1}{\l@
5339   dgetref@num{3}{#1}}}

```

```

5339 \newcommand*\xsublineref}[1]{\l@dgetref@num{3}{#1}}
5340
5341 %

```

\pstartref If the specified label exists, \pstartref gives its pstart number.

```

\pstartref
5342 \newcommand*\pstartref}[1]{\l@dref@undefined{#1}\wrap@edcrossref{#1}{\l@dgetref@num{4}{#1}}}
5343 \newcommand*\xpstartref}[1]{\l@dgetref@num{4}{#1}}
5344
5345 %

```

\xflagref \xflagref finds the side flag of any ref defined with \edlabel.

```

5346 \newcommand*\xflagref}[1]{\l@dgetref@num{5}{#1}}
5347 %

```

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.

```

5348 \newcommand*\l@dref@undefined}[1]{%
5349   \expandafter\ifx\csname the@label#1\endcsname\relax
5350     \led@warn@RefUndefined{#1}%
5351   \fi}
5352
5353 %

```

\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.

```

5354 \newcommand*\l@dgetref@num}[2]{%
5355   \expandafter
5356   \ifx\csname the@label#2\endcsname \relax
5357     000%
5358   \else
5359     \expandafter\expandafter\expandafter
5360     \l@dlabel@parse\csname the@label#2\endcsname|#1%
5361   \fi}
5362
5363 %

```


`\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`.)

```

5364 \newcommand*{\l@dlabel@parse}{%
5365 \def\l@dlabel@parse#1|#2|#3|#4|#5|#6{%
5366   \ifcase #6%
5367   \or #1%
5368   \or #2%
5369   \or #3%
5370   \or #4%
5371   \or #5%
5372   \fi}
5373 %

```

`\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.

```

5374 \newcommand*{\xxref}[2]{%
5375 {%
5376   \expandafter\ifx\csname the@label#1\endcsname \relax%
5377   \expandafter\let\csname the@@label#1\endcsname\zz@@@%
5378   \else%
5379     \expandafter\def\csname the@@label#1\endcsname{\l@dgetref@num
5380 {1}{#1}|\l@dgetref@num{2}{#1}|\l@dgetref@num{3}{#1}}%
5381   \fi%
5382   \expandafter\ifx\csname the@label#2\endcsname \relax%
5383   \expandafter\let\csname the@@label#2\endcsname\zz@@@%
5384   \else%
5385     \expandafter\def\csname the@@label#2\endcsname{\l@dgetref@num
5386 {1}{#2}|\l@dgetref@num{2}{#2}|\l@dgetref@num{3}{#2}}%
5387   \fi%
5388   \letcs{\@tempa}{the@@label#1}%
5389   \letcs{\@tempb}{the@@label#2}%
5390   \linenum{\@tempa|
5391     \@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.

`\SErefwithpage` So, first declare the default value of the hooks for the pseudo-series. Also declare the internal toggle which are switch by `reledmac`.

```

5392 \def\Xtwolines@appref{}%
5393 \def\Xtwolines@SEref{}%
5394
5395 \def\Xmorethantwolines@appref{}%
5396 \def\Xmorethantwolines@SEref{}%
5397
5398 \def\Xlinerangeseparator@appref{\endashchar}%
5399 \def\Xlinerangeseparator@SEref{\endashchar}%
5400
5401 \def\Xsublinesep@appref{\fullstop}%
5402 \def\Xsublinesep@SEref{\fullstop}%
5403
5404 \newtoggle{Xtwolinesbutnotmore@appref}%
5405 \newtoggle{Xtwolinesbutnotmore@SEref}%
5406
5407 \newtoggle{Xtwolinesonlyinsamepage@appref}%
5408
5409 \newtoggle{Xtwolinesonlyinsamepage@SEref}%
5410
5411 \newtoggle{Xlineflag@appref}%
5412 \toggletrue{Xlineflag@appref}%Here exception
5413 \newtoggle{Xlineflag@SEref}%
5414 \toggletrue{Xlineflag@SEref}%%Here exception
5415
5416 \def\Xendtwolines@apprefwithpage{}%
5417 \def\Xendtwolines@SErefwithpage{}%
5418
5419 \def\Xendmorethantwolines@apprefwithpage{}%
5420 \def\Xendmorethantwolines@SErefwithpage{}%
5421
5422 \def\Xendlinerangeseparator@apprefwithpage{\endashchar}
5423 \def\Xendlinerangeseparator@SErefwithpage{\endashchar}
5424 \def\Xendlinerangeseparator@SErefonlypage{\endashchar}
5425
5426 \def\Xendbeforepagenumber@apprefwithpage{p.}%
5427 \def\Xendbeforepagenumber@SErefwithpage{p.}%
5428 \def\Xendbeforepagenumber@SEonlypage{p.}%
5429
5430 \def\Xendafterpagenumber@apprefwithpage{ }%
5431 \def\Xendafterpagenumber@SErefwithpage{ }%
5432
5433

```

```

5434 \def\Xendlineprefixsingle@apprefwithpage{}%
5435 \def\Xendlineprefixsingle@SErefwithpage{}%
5436
5437 \def\Xendlineprefixmore@apprefwithpage{}%
5438 \def\Xendlineprefixmore@SErefwithpage{}%
5439
5440 \newtoggle{Xendtwolinesbutnotmore@apprefwithpage}%
5441 \newtoggle{Xendtwolinesbutnotmore@SErefwithpage}%
5442
5443 \def\Xendsublinesep@apprefwithpage{\fullstop}%
5444 \def\Xendsublinesep@SErefwithpage{\fullstop}%
5445
5446 \newtoggle{Xendtwolinesonlyinsamepage@apprefwithpage}%
5447 \newtoggle{Xendtwolinesonlyinsamepage@SErefwithpage}%
5448
5449 \newtoggle{Xendlineflag@apprefwithpage}
5450 \toggletrue{Xendlineflag@apprefwithpage}%Here, exception
5451 \newtoggle{Xendlineflag@SErefwithpage}
5452 \toggletrue{Xendlineflag@SErefwithpage}%Here, exception
5453
5454 %

```

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.

```

5455
5456 \xdef\Xboxstartlinenum@appref{Opt}
5457 \xdef\Xboxstartlinenum@SEref{Opt}
5458
5459 \xdef\Xboxendlinenum@appref{Opt}
5460 \xdef\Xboxendlinenum@SEref{Opt}
5461
5462 \xdef\Xendboxstartlinenum@apprefwithpage{Opt}
5463 \xdef\Xendboxstartlinenum@SErefwithpage{Opt}
5464
5465 \xdef\Xendboxendlinenum@apprefwithpage{Opt}
5466 \xdef\Xendboxendlinenum@SErefwithpage{Opt}
5467
5468 %

```

Now, declare the default values of \@apprefprefixsingle and \@apprefprefixmore, \@SErefprefix, \@SErefprefixmore and the commands which defines them.

```

5469 \newcommand\@apprefprefixsingle{}%
5470 \newcommand\@SErefprefixsingle{}%
5471
5472 \newcommand\@apprefprefixmore{}%
5473 \newcommand\@SErefprefixmore{}%
5474
5475 \newcommand{\setapprefprefixsingle}[1]{%

```

```

5476 \gdef\@apprefprefixsingle{#1}%
5477 }
5478 \newcommand{\setSErefprefixsingle}[1]{%
5479 \gdef\@SErefprefixsingle{#1}%
5480 }
5481
5482 \newcommand{\setapprefprefixmore}[1]{%
5483 \gdef\@apprefprefixmore{#1}%
5484 }
5485 \newcommand{\setSErefprefixmore}[1]{%
5486 \gdef\@SErefprefixmore{#1}%
5487 }
5488
5489 %

```

And not `\setSErefonlypageprefixsingle` and `\setSErefonlypageprefixmore`.

```

5490 \let\setSErefonlypageprefixsingle\XendbeforepagenumberSErefonlypage%
5491 \newcommand{\setSErefonlypageprefixmore}[1]{%
5492 \gdef\SErefonlypage@prefixmore{#1}%
5493 }%
5494 %

```

And now, the main commands: `\appref`, `\apprefwithpage`, `\SEref` and `\SErefwithpage`. These commands call `\reformatted@` and `\reformattedwithpage`, which calls `\printlines` and `\printendlines`. That is why we have previously declared all hooks values tested inside these last commands.

```

5495
5496 \newcommandx{\appref}[2][1,usedefault]{\reformatted@{#1}{#2}{appref}}
5497 \newcommandx{\SEref}[2][1,usedefault]{\reformatted@{#1}{#2}{SEref}}
5498
5499 \newcommandx{\apprefwithpage}[2][1,usedefault]{\reformattedwithpage@
5500 {#1}{#2}{appref}}
5501 \newcommandx{\SErefwithpage}[2][1,usedefault]{\reformattedwithpage@
5502 {#1}{#2}{SEref}}
5503
5504 \newcommandx{\SErefonlypage}[2][1,usedefault]{\reformattedonlypage@
5505 {#1}{#2}{SEref}}
5506
5507 \newcommand{\reformatted@}[3]{%
5508 \def\do##1{%
5509 \setkeys[mac]{truefootnoteoption}{##1}%
5510 }%
5511 \notblank{#1}{\docsvlist{#1}}}%
5512 \xdef\@currentseries{#3}%
5513 \ifcempty{#3prefixmore}%
5514 {\@apprefprefixsingle}%
5515 {%
5516 \IfEq{\xlineref{#2:start}}{\xlineref{#2:end}}%
5517 {\csuse{#3prefixsingle}}%
5518 }%
5519 }%

```

```

5515     {\csuse{@#3prefixmore}}}%
5516   }%
5517 \ifboolexpr{%
5518   test{\ifcsundef{the@label#2:start}}}%
5519   or test{\ifcsundef{the@label#2:end}}}%
5520 }%
5521 {\led@warn@pairRefUndefined{#2}\nfss@text{\reset@font\bfseries ??}}}%
5522 {%
5523   \def\@this@crossref@start{#2:start}%
5524   \def\@this@crossref@end{#2:end}%
5525   \printlines\xpageref{#2:start}|\xlineref{#2:start}|\xsublineref{#2:
start}|\xpageref{#2:end}|\xlineref{#2:end}|\xsublineref{#2:end}|\relax|\
xflagref{#2:start}|\%
5526   \undef\@this@crossref@end%
5527   \undef\@this@crossref@start%
5528 }%
5529 \def\do##1{%
5530   \setkeys[mac]{falsefootnoteoption}{##1}%
5531 }%
5532 \notblank{#1}{\docsvlist{#1}}{}%
5533 }%
5534
5535 \newcommand{\reformattedwithpage@}[3]{%
5536   \def\do##1{%
5537     \setkeys[mac]{truefootnoteoption}{##1}%
5538   }%
5539   \notblank{#1}{\docsvlist{#1}}{}%
5540   \xdef\@currentseries{#3withpage}%
5541   \ifboolexpr{%
5542     test{\ifcsundef{the@label#2:start}}}%
5543     or test{\ifcsundef{the@label#2:end}}}%
5544   }%
5545   {\led@warn@pairRefUndefined{#2}\nfss@text{\reset@font\bfseries ??}}}%
5546   {%
5547     \def\@this@crossref@start{#2:start}%
5548     \def\@this@crossref@end{#2:end}%
5549     \printendlines\xpageref{#2:start}|\xlineref{#2:start}|\xsublineref{#2:
start}|\xpageref{#2:end}|\xlineref{#2:end}|\xsublineref{#2:end}|\relax|\
xflagref{#2:start}|\%
5550     \undef\@this@crossref@end%
5551     \undef\@this@crossref@start%
5552   }%
5553   \def\do##1{%
5554     \setkeys[mac]{falsefootnoteoption}{##1}%
5555   }%
5556   \notblank{#1}{\docsvlist{#1}}{}%
5557 }%
5558
5559 \newcommand{\reformattedonlypage@}[3]{%
5560   \def\do##1{%

```

```

5561 \setkeys[mac]{truefootnoteoption}{##1}%
5562 }%
5563 \notblank{#1}{\docsvlist{#1}}{ }%
5564 \xdef\@currentseries{#3onlypage}%
5565 \ifboolexpr{%
5566   test{\ifcsundef{the@label#2:start}}%
5567   or test{\ifcsundef{the@label#2:end}}%
5568 }%
5569 {\led@warn@pairRefUndefined{#2}\nfss@text{\reset@font\bfseries ??}}%
5570 {\ifnumequal{\xpageref{#2:end}}{\xpageref{#2:start}}%
5571   {%
5572     \printnpnum{%
5573       \wrap@edcrossref{#2:start}{\xpageref{#2:start}}%
5574     }%
5575   }%
5576 }%
5577 \ifcsvoid{#3onlypage@prefixmore}%
5578 {}%
5579 {\csletcs{Xendbeforepagenumber@#3onlypage}{#3onlypage@prefixmore}}%
5580 \ifdefined\linerangesep@%
5581   \printnpnum{%
5582     \wrap@edcrossref{#2:start}{\xpageref{#2:start}}%
5583     \linerangesep@%
5584     \wrap@edcrossref{#2:end}{\xpageref{#2:end}}%
5585   }%
5586 \else%
5587   \printnpnum{%
5588     \wrap@edcrossref{#2:start}{\xpageref{#2:start}}%
5589     \csuse{Xendlinerangeseparator@\@currentseries}%
5590     \wrap@edcrossref{#2:end}{\xpageref{#2:end}}%
5591   }%
5592 \fi%
5593 }%
5594 }%
5595 \def\do##1{%
5596   \setkeys[mac]{falsefootnoteoption}{##1}%
5597 }%
5598 \notblank{#1}{\docsvlist{#1}}{ }%
5599 }%
5600 %

```

`\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`.

```

5601 \newcommand*{\edmakelabel}[2]{\expandafter\xdef\csname the@label#1\

```

```
endcsname{#2}}
```

```
5602
5603 %
```

(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. 119 and V.9 p. 89), 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`.

```
5604 \pretocmd{\XR@test}%
5605   {\XR@test@mac+++#1#2#3#4+++}%
5606   {}%
5607   {}%
5608   %
```

\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`.

```
5609 \long\def\xR@test@mac+++#1+++{\XR@test@mac@test#1}
5610 %
```

\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`.

```
5611 \long\def\xR@test@mac@test#1#2...{%The triple dots (NOT \ldots) are because
5612   of the line 22 of xr.sty v5.02 1994/05/28
5613   \ifx#1\l@dmake@labels%
5614     \l@dmake@labels#2%
5615   \else
5616     \ifx#1\l@dmake@labelsR%
5617       \l@dmake@labelsR #2%
5618     \fi%
5619   }%
5620   %
```

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.

```

5621 \pretocmd{\@xympar}%
5622   {\ifnumberedpar@
5623     \led@warn@NoMarginpars
5624     \@esphack
5625   \else}%
5626   {}%
5627   {}%
5628
5629 \apptocmd{\@xympar}%
5630   {\fi}%
5631   {}
5632   {}
5633
5634 %

```

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

```

5635 \newcount\sidenote@margin
5636 \newcommand*{\sidenotemargin}[1]{\{%
5637   \l@dgetsidenote@margin{#1}%
5638   \ifnum\@l@dtmptcntb>\m@ne
5639     \ifledRcol
5640       \global\sidenote@marginR=\@l@dtmptcntb
5641     \else
5642       \global\sidenote@margin=\@l@dtmptcntb
5643     \fi
5644   \fi}}
5645 \newcommand*{\l@dgetsidenote@margin}[1]{\%
5646   \def\@tempa{#1}\def\@tempb{left}%
5647   \ifx\@tempa\@tempb
5648     \l@dtmptcntb \z@
5649   \else
5650     \def\@tempb{right}%
5651     \ifx\@tempa\@tempb

```



```

5652 \l@dtempcntb \@ne
5653 \else
5654 \def\@tempb{outer}%
5655 \ifx\@tempa\@tempb
5656 \l@dtempcntb \tw@
5657 \else
5658 \def\@tempb{inner}%
5659 \ifx\@tempa\@tempb
5660 \l@dtempcntb \thr@@
5661 \else
5662 \led@warn@BadSidenotemargin
5663 \l@dtempcntb \m@ne
5664 \fi
5665 \fi
5666 \fi
5667 \fi}
5668 \sidenotemargin{right}
5669
5670 %

```

`\l@dlp@rbox` We need two boxes to store sidenote texts.

```

\l@drp@rbox
5671 \newbox\l@dlp@rbox
5672 \newbox\l@drp@rbox
5673
5674 %

```

`\ledlsnotewidth` These specify the width of the left/right boxes (initialised to `\marginparwidth`), their distance from the text (initialised to `\linenumsep`), and the fonts used.

```

\ledrsnotewidth
\ledlsnotesep
5675 \newdimen\ledlsnotewidth \ledlsnotewidth=\marginparwidth
\ledrsnotesep
5676 \newdimen\ledrsnotewidth \ledrsnotewidth=\marginparwidth
\ledlsnotefontsetup
5677 \newdimen\ledlsnotesep \ledlsnotesep=\linenumsep
\ledrsnotefontsetup
5678 \newdimen\ledrsnotesep \ledrsnotesep=\linenumsep
5679 \newcommand*\ledlsnotefontsetup{\raggedleft\footnotesize}
5680 \newcommand*\ledrsnotefontsetup{\raggedright\footnotesize}
5681
5682 %

```

`\ledleftnote` `\ledrightnote`, `\ledinnernote`, `\ledouternote` are the user commands for left, right, inner and outer sidenotes. The two last one are just alias for the two first one, depending of the page number. `\ledsidenote{<text>}` is the command for a moveable sidenote.

```

\ledsidenote
5683 \newcommand*\ledleftnote[1]{\edtext{\l@dlsnote{#1}}}
5684 \newcommand*\ledrightnote[1]{\edtext{\l@drsnote{#1}}}
5685
5686 \newcommand*\ledinnernote[1]{%
5687 \ifodd\c@page% Do not use \page@num, because it is not yet calculated
when command is called

```

```

5688 \ledleftnote{#1}%
5689 \else%
5690 \ledrightnote{#1}%
5691 \fi%
5692 }
5693
5694 \newcommand*{\ledouternote}[1]{%
5695 \ifodd\c@page% Do not use \page@num, because it is not yet calculated
5696 when command is called
5697 \ledrightnote{#1}%
5698 \else%
5699 \ledleftnote{#1}%
5700 \fi%
5701 }
5702 \newcommand*{\ledsidenote}[1]{\edtext{}{\l@dcsnote{#1}}}
5703 %

```

`\l@dlsnote` . The ‘footnotes’ for left, right, and moveable sidenotes. The whole scheme is reminiscent of the critical footnotes code.

`\l@drsnote`

`\l@dcsnote`

```

5704 \newif\ifrighnoteup
5705 \righnoteuptrue
5706
5707 \newcommand*{\l@dlsnote}[1]{%
5708 \begingroup%
5709 \newcommand{\content}{#1}%
5710 \ifnumberedpar@
5711 \ifledRcol%
5712 \xright@appenditem{\noexpand\l@dlsnote{\expandonce\content}}%
5713 \to\inserts@listR
5714 \global\advance\insert@countR \@ne%
5715 \else%
5716 \xright@appenditem{\noexpand\l@dlsnote{\expandonce\content}}%
5717 \to\inserts@list
5718 \global\advance\insert@count \@ne%
5719 \fi
5720 \fi\ignorespaces\endgroup}
5721
5722 \newcommand*{\l@drsnote}[1]{%
5723 \begingroup%
5724 \newcommand{\content}{#1}%
5725 \ifnumberedpar@
5726 \ifledRcol%
5727 \xright@appenditem{\noexpand\l@drsnote{\expandonce\content}}%
5728 \to\inserts@listR
5729 \global\advance\insert@countR \@ne%
5730 \else%
5731 \xright@appenditem{\noexpand\l@drsnote{\expandonce\content}}%
5732 \to\inserts@list

```

```

5733 \global\advance\insert@count \@ne%
5734 \fi
5735 \fi\ignorespaces\endgroup}
5736
5737 \newcommand*{\l@dcsnote}[1]{%
5738 \begingroup%
5739 \newcommand{\content}{#1}%
5740 \ifnumberedpar@
5741 \ifledRcol%
5742 \xright@appenditem{\noexpand\l@dcsnote{\expandonce\content}}{%
5743 \to\inserts@listR
5744 \global\advance\insert@countR \@ne%
5745 \else%
5746 \xright@appenditem{\noexpand\l@dcsnote{\expandonce\content}}{%
5747 \to\inserts@list
5748 \global\advance\insert@count \@ne%
5749 \fi
5750 \fi\ignorespaces\endgroup}
5751
5752 %

```

`\vl@dlsnote` Put the left/right text into boxes, but just save the moveable text. `\l@dcsnotetext`, `\vl@drsnote` `\l@dcsnotetext@l` and `\l@dcsnotetext@r` are etoolbox's lists which will store the content of side notes. We store the content in lists, because we need to loop later on them, in case many sidenote co-exist for the same line. That is there some special test to do, in order to:

- Store the content of `\ledsidenote` to `\l@dcsnotetext` in any cases.
- Store the content of `\rightsidenote` to:
 - `\l@dcsnotetext` if `\ledsidenote` is to be put on right.
 - `\l@dcsnotetext@r` if `\ledsidenote` is to be put on left.
- Store the content of `\leftsidenote` to:
 - `\l@dcsnotetext` if `\ledsidenote` is to be put on left.
 - `\l@dcsnotetext@l` if `\ledsidenote` is to be put on right.

```

5753 \newcommand*{\vl@dlsnote}[1]{%
5754 \ifledRcol%
5755 \@l@dttempcntb=\sidenote@marginR%
5756 \ifnum\@l@dttempcntb>\@ne%
5757 \advance\@l@dttempcntb by\page@numR%
5758 \fi%
5759 \else%
5760 \@l@dttempcntb=\sidenote@margin%
5761 \ifnum\@l@dttempcntb>\@ne%
5762 \advance\@l@dttempcntb by\page@num%

```

```

5763 \fi%
5764 \fi%
5765 \ifodd\@l@dttempcntb%
5766 \listgadd{\l@dcsnotetext@l}{#1}%
5767 \else%
5768 \listgadd{\l@dcsnotetext}{#1}%
5769 \fi
5770 }
5771 \newcommand*{\vl@drsnote}[1]{%
5772 \ifledRcol%
5773 \l@dttempcntb=\sidenote@marginR%
5774 \ifnum\l@dttempcntb>\@ne%
5775 \advance\l@dttempcntb by\page@numR%
5776 \fi%
5777 \else%
5778 \l@dttempcntb=\sidenote@margin%
5779 \ifnum\l@dttempcntb>\@ne%
5780 \advance\l@dttempcntb by\page@num%
5781 \fi%
5782 \fi%
5783 \ifodd\@l@dttempcntb%
5784 \listgadd{\l@dcsnotetext}{#1}%
5785 \else%
5786 \listgadd{\l@dcsnotetext@r}{#1}%
5787 \fi%
5788 }
5789 \newcommand*{\vl@dcsnote}[1]{\listgadd{\l@dcsnotetext}{#1}}
5790
5791 %

```

`\setl@dlp@rbox` `\setl@dlprbox{<lednums>}{<tag>}{<text>}` puts `<text>` into the `\l@dlp@rbox` box.
`\setl@drpr@box` And similarly for the right side box. It is these boxes that finally get displayed in the margins.

```

5792 \newcommand*{\setl@dlp@rbox}[1]{%
5793 \begingroup%
5794 \parindent\z@\hspace=\ledlsnotewidth%
5795 \ledlsnotefontsetup%We kept it outside of the vbox, because can affect
the ragging
5796 \global\setbox\l@dlp@rbox%
5797 \ifleftnoteup%
5798 =\vbox to\z@{\ledlsnotefontsetup\vss #1}}%We put \
ledlsnotefontsetup inside footnote because required for color command. Note
the {} to kept setting local.
5799 \else%
5800 =\vbox to 0.70\baselineskip{\ledlsnotefontsetup\strut#1\vss}}%
5801 \fi%
5802 \endgroup%
5803 }
5804

```

```

5805 \newcommand*{\setl@drp@rbox}[1]{%
5806   \begingroup%
5807   \parindent\z@\hsize=\ledrsnotewidth%
5808   \ledrsnotefontsetup%We kept it outside of the vbox, because can affect
the ragging
5809   \global\setbox\l@drp@rbox%
5810   \ifrightnoteup%
5811     =\vbox to\z@{\{\ledrsnotefontsetup\vss#1\}}%We put \ledrsnotefontsetup
inside footnote because required for color command. Note the {} to kept
setting local.
5812   \else%
5813     =\vbox to0.7\baselineskip{\{\ledrsnotefontsetup\strut#1\vss\}}%
5814   \fi%
5815   \endgroup%
5816 }%
5817 \newif\ifleftnoteup
5818 \leftnoteuptrue
5819 %

```

\@sidenotesep This macro is used to separate sidenotes of the same line.

```

5820 \newcommand{\setsidenotesep}[1]{\gdef\@sidenotesep{#1}}
5821 \newcommand{\@sidenotesep}{, }
5822 %

```

\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.

```

5823 \newcommand*{\affixside@note}{%
5824   \def\sidenotecontent@{}%
5825   \numgdef{\itemcount@}{0}%
5826   \def\do##1{%
5827     \ifnumequal{\itemcount@}{0}%
5828       {%
5829         \appto\sidenotecontent@{##1}}% Not print not separator before
the 1st note
5830       {\appto\sidenotecontent@{\@sidenotesep ##1}}%
5831     }%
5832     \numgdef{\itemcount@}{\itemcount@+1}%
5833   }%
5834   \dolistloop{\l@dcsnotetext}%
5835   \ifnumgreater{\itemcount@}{1}{\led@err@ManySidenotes}{}%
5836 %

```

And we do the same for left and right notes (not movable).

```

5837 \gdef\@templ@d{}%

```

```

5838 \gdef\@templ@n{\l@dcsnotetext\l@dcsnotetext@l\l@dcsnotetext@r}%
5839 \ifx\@templ@d\@templ@n \else%
5840   \if@twocolumn%
5841     \if@firstcolumn%
5842       \setl@dlp@rbox{##1}{\sidenotecontent@}%
5843     \else%
5844       \setl@drp@rbox{\sidenotecontent@}%
5845     \fi%
5846   \else%
5847     \l@dtempcntb=\sidenote@margin%
5848     \ifnum\l@dtempcntb>\@ne%
5849       \advance\l@dtempcntb by\page@num%
5850     \fi%
5851     \ifodd\l@dtempcntb%
5852       \setl@drp@rbox{\sidenotecontent@}%
5853       \gdef\sidenotecontent@{}%
5854       \numgdef{\itemcount@}{0}%
5855       \dolistloop{\l@dcsnotetext@l}%
5856       \ifnumgreater{\itemcount@}{1}{\led@err@ManyLeftnotes}{}%
5857       \setl@dlp@rbox{\sidenotecontent@}%
5858     \else%
5859       \setl@dlp@rbox{\sidenotecontent@}%
5860       \gdef\sidenotecontent@{}%
5861       \numgdef{\itemcount@}{0}%
5862       \dolistloop{\l@dcsnotetext@r}%
5863       \ifnumgreater{\itemcount@}{1}{\led@err@ManyRightnotes}{}%
5864       \setl@drp@rbox{\sidenotecontent@}%
5865     \fi%
5866   \fi%
5867 \fi%
5868 }
5869 %

```

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 `\@iiiminipage` and `\endminipage` macros. We will arrange this so that additional series can be easily added.

`\l@dfeetbeginmini` These will be the hooks in `\@iiiminipage` and `\endminipage`.
`\l@dfeetendmini` They can be extended to handle other things if necessary.

```

5870 \ifnoledgroup\else%
5871 \newcommand*{\l@dfeetbeginmini}{\@ledgrouptrue\l@dedbeginmini\l@dfambeginmini}
5872 \newcommand*{\l@dfeetendmini}{%
5873   \IfStrEq{critical-familiar}{\@mpfnpos}%
5874   {\l@dedendmini\l@dfamendmini}%

```

```

5875      {%
5876      \IfStrEq{familiar-critical}{\@mpfnpos}%
5877      {\l@dfamendmini\l@dedendmini}%
5878      {\l@dedendmini\l@dfamendmini}%
5879      }%
5880    }%
5881  %

```

\l@dedbeginmini These handle the initiation and closure of critical footnotes in a minipage environment.

```

\l@dedendmini
5882 \newcommand*{\l@dedbeginmini}{%
5883   \unless\ifnocritical@%
5884   \def\do##1{\csletcs{v##1footnote}{mpv##1footnote}}%
5885   \dolistloop{\@series}%
5886   \fi%
5887 }
5888 \newcommand*{\l@dedendmini}{%
5889   \unless\ifnocritical@%
5890   \ifl@dpairing%
5891     \ifledRcol%
5892       \flush@notesR%
5893     \else%
5894       \flush@notes%
5895     \fi%
5896   \fi
5897   \def\do##1{%
5898     \ifvoid\csuse{mp##1footins}\else%
5899     \ifl@dpairing\ifparledgroup%
5900       \ifledRcol%
5901       \dingdef{\parledgroup@beforenotesR}{\parledgroup@beforenotesR+\
skip\@nameuse{mp##1footins}}%
5902     \else%
5903       \dingdef{\parledgroup@beforenotesL}{\parledgroup@beforenotesL
+\skip\@nameuse{mp##1footins}}%
5904     \fi%
5905     \fi\fi%
5906     \csuse{mp##1footgroup}{##1}%
5907     \fi}%
5908   \dolistloop{\@series}%
5909   \fi%
5910 }%
5911 %
5912 %

```

\l@dfambeginmini These handle the initiation and closure of familiar footnotes in a minipage environment.

```

\l@dfamendmini
5913 \newcommand*{\l@dfambeginmini}{%
5914   \unless\ifnofamiliar@%
5915   \def\do##1{\csletcs{vfootnote##1}{mpvfootnote##1}}%
5916   \dolistloop{\@series}%

```

```

5917 \fi%
5918 }%
5919
5920 \newcommand*{\l@dfamendmini}{%
5921 \unless\ifnofamiliar%
5922 \def\do##1{%
5923 \ifvoid\csuse{mpfootins##1}\else%
5924 \csuse{mpfootgroup##1}{##1}%
5925 \fi}%
5926 \dolistloop{\@series}%
5927 \fi%
5928 }%
5929 %

```

`\@iiiminipage` This is our extended form of the kernel `\@iiiminipage` defined in `ltboxes.dtx`.

```

5930 \patchcmd%
5931 {\@iiiminipage}%
5932 {\let\@footnotetext\@mpfootnotetext}%
5933 {\let\@footnotetext\@mpfootnotetext\l@dfeetbeginmini}%
5934 {}%
5935 {\led@error@fail@patch@%iiiminipage}%
5936 %

```

`\endminipage` This is our extended form of the kernel `\endminipage` defined in `ltboxes.dtx`.

```

5937 \patchcmd%
5938 {\endminipage}%
5939 {\footnoterule}%
5940 {\footnoterule\l@advance@parledgroup@beforenormalnotes}%
5941 {}%
5942 {\led@error@fail@patch@endminipage}
5943
5944 \patchcmd%
5945 {\endminipage}%
5946 {\@minipagefalse}%
5947 {\l@dfeetendmini\@minipagefalse}%
5948 {}%
5949 {\led@error@fail@patch@endminipage}
5950
5951 %

```

`\l@dunboxmpfoot` `\@ldunboxmpfoot` insert normal footnotes for ledgroup.
`\l@advance@parledgroup@beforenormalnotes`

```

5952 \newcommand*{\l@dunboxmpfoot}{%
5953 \vskip\skip\@mpfootins
5954 \normalcolor
5955 \footnoterule
5956 \l@advance@parledgroup@beforenormalnotes
5957 \unvbox\@mpfootins%

```



```

5958 }
5959 %

```

When using parallel `ledgroup`, we need to store the vertical space added before footnote, in order to compensate them between left and right pages.

```

5960 \newcommand{\l@advance@parledgroup@beforenormalnotes}{%
5961   \ifparledgroup
5962     \ifl@dpairing
5963       \ifledRcol
5964         \dimdef{\parledgroup@beforenotesR}{\parledgroup@beforenotesR+
skip\@mpfootins}
5965       \else
5966         \dimdef{\parledgroup@beforenotesL}{\parledgroup@beforenotesL+
skip\@mpfootins}
5967       \fi
5968     \fi
5969   \fi
5970 }
5971 %

```

`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.

```

5972 \newenvironment{ledgroup}{%
5973   \resetprevpage@num%
5974   \def\@mpfn{mpfootnote}\def\thempfn{\thempfootnote}\c@mpfootnote\z@%
5975   \let\@footnotetext\@mpfootnotetext
5976   \l@dfeetbeginmini%
5977 }{%
5978   \par
5979   \unskip
5980   \ifvoid\@mpfootins\else
5981     \l@dunboxmpfoot
5982   \fi
5983   \l@dfeetendmini%
5984   \@ledgroupfalse%
5985 }
5986
5987
5988
5989 %

```

`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.

```

5990 \newenvironment{ledgroupsize}[2][1]{%
5991 %

```

Set the various text measures.

```
5992 \hspace #2\relax
5993 %
```

Initialize fills for centering.

```
5994 \let\ledllfill\hfil
5995 \let\ledrlfill\hfil
5996 \def\@tempa{#1}\def\@tempb{1}%
5997 %
```

Left adjusted numbered lines

```
5998 \ifx\@tempa\@tempb
5999 \let\ledllfill\relax
6000 \else
6001 \def\@tempb{r}%
6002 \ifx\@tempa\@tempb
6003 %
```

Right adjusted numbered lines

```
6004 \let\ledrlfill\relax
6005 \fi
6006 \fi
6007 %
```

Set up the footnoting.

```
6008 \def\@mpfn{mpfootnote}\def\thempfn{\thempfootnote}\c@mpfootnote\z@
6009 \let\@footnotetext\@mpfootnotetext
6010 \l@dfetbeginmini%
6011 }{ %
6012 \par
6013 \unskip
6014 \ifvoid\@mpfootins\else
6015 \l@duboxmpfoot
6016 \fi
6017 \l@dfetendmini%
6018 }
6019
6020 %
```

Close the \ifnoledgroup@\else.

```
6021 \fi %
6022 %
```

`\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`.

```
6023 \newif\ifledgroupnotesL@
6024 \newif\ifledgroupnotesR@
6025 %
```

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`.

```

6026 \AtBeginDocument{%
6027   \unless\ifl@imakeidx%
6028     \@ifpackageloaded{imakeidx}{\led@error@ImakeidxAfterEledmac}{}%
6029   \fi%
6030   \unless\ifl@indextools%
6031     \@ifpackageloaded{indextools}{\led@error@indextoolsAfterEledmac}{}%
6032   \fi%
6033 }
6034 %

```

XXVI.2 Auxiliary macros for `\edindex`

`\pagelinesep` In order to get a correct line number we have to use the label/ref mechanism. These macros are for that.

```

\edindexlab
\c@labidx
6035 \newcommand{\pagelinesep}{-}
6036 \newcommand{\edindexlab}{${&}}
6037 \newcounter{labidx}
6038 \setcounter{labidx}{0}
6039
6040 %

```

`\doedindexlabel` This macro sets an `\edlabel`.

```

6041 \newcommand{\doedindexlabel}{%
6042   \stepcounter{labidx}%
6043   \edlabel{\edindexlab\thelabidx}%
6044 }
6045
6046 %

```

`\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`.

```

6047 \newcounter{pageline}%
6048 \renewcommand{\thepageline}{%
6049   \thepage%
6050   \pagelinesep%
6051   \xlineref{\edindexlab\thelabidx}%
6052 }
6053 %

```

`\thestartpageline` `\theendpageline` These macros make up the page/line start/end number when the `\edindex` command is called in critical notes.

```

6054 \newcommand{\thestartpageline}{%
6055   \l@dparsedstartpage%
6056   \pagelinesep%
6057   \l@dparsedstartline%
6058 }
6059 \newcommand{\theendpageline}{%
6060   \l@dparsedendpage%
6061   \pagelinesep%
6062   \l@dparsedendline%
6063 }
6064 %

```

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.

```

6065 \newif\if@edindex@fornote@
6066 %

```

`\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 |.

```

6067 \newcommand{\prepare@edindex@fornote}[1]{%
6068   \l@dp@rsefootspec#1|%
6069   \@edindex@fornote@true%
6070 }
6071 %

```

`\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.

```

6072 \newcommand{\get@edindex@ledinnote@command}{%
6073   \ifxindy@%
6074     \gdef\@ledinnote@command{%
6075       ledinnote\thelabidx%
6076     }%
6077     \ifxindyhyperref@%
6078       \immediate\write\eledmac@xindy@out{%
6079         (define-attributes ("ledinnote\thelabidx"))^^J
6080         \space\space(markup-locref^^J
6081         \eledmacmarkuplocrefdepth^^J

```

```

6082 :open "\string\ledinnote[\edindexlab\thelabidx]{\@index@command
}{^^J
6083 :close "}"^^J
6084 :attr "ledinnote\thelabidx"^^J
6085 )
6086 }%
6087 \else%
6088 \immediate\write\eledmac@xindy@out{%
6089 (define-attributes ("ledinnote\thelabidx"))^^J
6090 \space\space(markup-locref^^J
6091 \eledmacmarkuplocdepth^^J
6092 :open "\string\ledinnote{\@index@command}{^^J
6093 :close "}"^^J
6094 :attr "ledinnote\thelabidx"^^J
6095 )
6096 }%
6097 \fi%
6098 %

```

If we do not use xindy option, `\@ledinnote@command` will produce something like `ledinnote{formattingcommand}`.

```

6099 \else%
6100 \gdef\@ledinnote@command{%
6101 ledinnote[\edindexlab\thelabidx]{\@index@command}%
6102 }%
6103 \fi%
6104 }
6105 %

```

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 |.

```

6106 \def\get@index@command#1|#2+{%
6107 \gdef\@index@txt{#1}%
6108 \gdef\@index@command{#2}%
6109 \xdef\@index@parenthesis{}%
6110 \IfBeginWith{\@index@command}{({}%
6111 \StrGobbleLeft{\@index@command}{1}[\@index@command@]%
6112 \global\let\@index@command\@index@command@%
6113 \xdef\@index@parenthesis{({}%
6114 }{}%
6115 \IfBeginWith{\@index@command}{)}{%
6116 \StrGobbleLeft{\@index@command}{1}[\@index@command@]%
6117 \global\let\@index@command\@index@command@%
6118 \xdef\@index@parenthesis{)}%
6119 }{}%
6120 }
6121 %

```

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`

```

6122 \newcommandx{\ledinnote}[3][1,usedefault]{%
6123   \ifboolexpr{%
6124     test{\ifdefequal{\iftrue}{\ifHy@hyperindex}}%
6125     or%
6126     bool {xindyhyperref}%
6127   }%
6128   {%
6129     \csuse{#2}{\hyperlink{#1}{\ledinnotemark{#3}}}%
6130   }%
6131   {%
6132     \csuse{#2}{\ledinnotemark{#3}}%
6133   }%
6134 }%
6135 \newcommand{\ledinnotehyperpage}[2]{\csuse{#1}{\ledinnotemark{\hyperpage
6136   {#2}}}}%
6137 \newcommand{\ledinnotemark}[1]{#1\emph{n}}%
6138 %

```

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`

```

6138 \newcommandx{\@wredindex}[2][1=\expandonce\jobname,usedefault]{%#1 = the
6139   index name, #2 = the text
6140   \ifl@imakeidx%
6141     \if@edindex@fornote%
6142       \IfSubStr[1]{#2}{|}{\get@index@command#2+}{\get@index@command#2|+}%
6143       \get@edindex@ledinnote@command%
6144       \expandafter\imki@wrindexentry{#1}{\@index@txt|(\@ledinnote@command
6145       ){\thestartpageline}}%
6146       \expandafter\imki@wrindexentry{#1}{\@index@txt|)\@ledinnote@command
6147       }{\theendpageline}}%
6148     \else%
6149       \get@edindex@hyperref{#2}%
6150       \imki@wrindexentry{#1}{\@index@txt\@edindex@hyperref}{\thepageline}%

```

```

6148 \fi%
6149 \else%
6150 \if@edindex@fornote@%
6151 \IfSubStr[1]{#2}{|}{\get@index@command#2+}{\get@index@command#2|+}%
6152 \get@edindex@ledinnote@command%
6153 \expandafter\protected@write\@indexfile{%
6154 {\string\indexentry{\@index@txt|(\@ledinnote@command){\thestartpageline}
6155 }%
6156 \expandafter\protected@write\@indexfile{%
6157 {\string\indexentry{\@index@txt|)\@ledinnote@command}{\theendpageline}
6158 }%
6159 \else%
6160 \protected@write\@indexfile{%
6161 {\string\indexentry{#2}{\thepageline}
6162 }%
6163 \fi%
6164 \fi%
6165 \endgroup
6166 \@esphack%
6167 }
6168 %

```

Need to add the definition of `\edindex` to `\makeindex`, and initialise `\edindex` to do nothing.

```

6169 \pretocmd{\makeindex}{%
6170 \def\edindex{\@bsphack
6171 \doedindexlabel
6172 \begingroup
6173 \@sanitize
6174 \@wredindex}}{}{}
6175 \newcommand{\edindex}[1]{\@bsphack\@esphack}
6176 %

```

XXVI.7 **Hyperlink**

`\hyperlinkformat` `\hyperlinkformat` command is to be used to have both a internal hyperlink and a format, when indexing.

```

6177 \newcommand{\hyperlinkformat}[3]{%
6178 \ifstrempy{#1}%
6179 {\hyperlink{#2}{#3}}%
6180 {\csuse{#1}{\hyperlink{#2}{#3}}%
6181 }}
6182 %

```

`\hyperlinkR` `\hyperlinkR` command is to be used to create a internal hyperlink and `\ledRflag`, when indexing.

```

6183 \newcommand{\hyperlinkR}[2]{%

```

```

6184 \hyperlink{#1}{#2\@Rlineflag}%
6185 }%
6186
6187 %

```

\hyperlinkformatR \hyperlinkformatR command is to be used to create a internal hyperlink, a format and a \@Rlineflag, when indexing.

```

6188 \newcommand{\hyperlinkformatR}[3]{%
6189 \hyperlinkformat{#1}{#2}{#3\@Rlineflag}%
6190 }%
6191
6192 %

```

\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).

```

6193 \newcommand{\get@edindex@hyperref}[1]{%
6194 %

```

We have to disable temporary spaces to work through a xstring bug (or feature?)

```

6195 \edef\temp@{%
6196 \catcode`\ =9 %space need for catcode
6197 \detokenize{#1}%For active character in unicode
6198 \catcode`\ =10 % space need for catcode
6199 }%
6200 %

```

Now, we define \@edindex@hyperref if the hyperindex of hyperref is enabled.

```

6201 \ifdefequal{\iftrue}{\ifHy@hyperindex}{%
6202 \IfSubStr{\temp@}{|}%
6203 {\get@index@command#1+%
6204 \iflabeledRcol%
6205 \gdef\@edindex@hyperref{|\@index@parenthesis %space kept
6206 hyperlinkformatR{\@index@command}%
6207 {\edindexlab\thelabidx}}%
6208 \else%
6209 \gdef\@edindex@hyperref{|\@index@parenthesis %space kept
6210 hyperlinkformat{\@index@command}%
6211 {\edindexlab\thelabidx}}%
6212 \fi%
6213 }%
6214 {\get@index@command#1|+%
6215 \iflabeledRcol%
6216 \gdef\@edindex@hyperref{|\hyperlinkR{\edindexlab\thelabidx}}%
6217 \else%
6218 \gdef\@edindex@hyperref{|\hyperlink{\edindexlab\thelabidx}}%
6219 \fi%
6220 }%
6221 }%
6222 %

```



```

6223 % If we use both xindy and hyperref, first get the \protect\cs{
        index@command} command.
6224 % Then define \protect\cs{@edindex@hyperref} in the form \verb+eledmacXXX+
6225 % \begin{macrocode}
6226 {\ifxindyhyperref%
6227   \IfSubStr{\temp@}{|}%
6228   {\get@index@command#1+}%
6229   {\get@index@command#1|+}%
6230   \gdef\@edindex@hyperref{|eledmac\thelabidx}%
6231 %

```

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`.

```

6232   \IfStrEq{\@index@parenthesis}{(}%
6233   {%
6234     \csxdef{xindyparenthesis@\@index@txt}{\thelabidx}%
6235     \gdef\@edindex@hyperref{| (eledmac\thelabidx}%
6236   }%
6237   {}%
6238 %

```

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`.

```

6239   \IfStrEq{\@index@parenthesis}{)}%
6240   {%
6241     \xdef\@edindex@hyperref{|)eledmac\csuse{xindyparenthesis@\@index@txt}}%
6242     \global\csundef{xindyparenthesis@\@index@txt}%
6243   }%
6244 %

```

Write in the `.xdy` file the attributes of the location.

```

6245   {%
6246     \immediate\write\eledmac@xindy@out{%
6247       (define-attributes ("eledmac\thelabidx"))^^J
6248       \space\space(markup-locoref^^J
6249         \eledmacmarkuplocorefdepth^^J
6250         :open "\string\hyperlink%
6251           \ifledRcol R\fi%
6252           {\edindexlab\thelabidx}%
6253           {\ifdefempty{\@index@command}%
6254             {}%
6255             {\@backslashchar\@index@command}%
6256           {"^^J
6257         :close "})"^^J
6258         :attr "eledmac\thelabidx"^^J
6259       )

```

```

6260     }%
6261   }%
6262 %

```

And now, in any other case.

```

6263   \else%
6264     \gdef\@index@txt{#1}%
6265     \gdef\@edindex@hyperref{}%
6266   \fi%
6267 }%
6268 }
6269 %

```

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.

```

6270 \newcommand{\led@set@index@fornote}[1]{%
6271   \ifbool{indtl@innote}%
6272     {\let\index\nindex}%
6273     {}%
6274   \ifbool{indtl@notenumber}%
6275     {%
6276       \renewcommand{\index}[2][\indtl@jobname]{%
6277         \orig@@index[##1]{%
6278           ##2|innotenumber{\csuse{thefootnote#1}}}%
6279         }%
6280       }%
6281     }%
6282   {}%
6283 }%
6284 %

```

\led@reinit@index@fornote The `\led@reinit@index@fornote` just reset the default value of `\index`.

```

6285 \newcommand{\led@reinit@index@fornote}{%
6286   \ifbool{indtl@innote}%
6287     {\let\index\orig@@index}%
6288     {}%
6289   \ifbool{indtl@notenumber}%
6290     {\let\index\orig@@index}%
6291     {}%
6292 }%
6293 %

```

XXVII Glossaries

Here, we define the \gls-like commands prefixed by ed, only if the package glossaries is loaded.

```

6294 \AtBeginDocument{%
6295   \@ifpackageloaded{glossaries}{%
6296     %

```

First those which arguments are [*options*]{*label*}[*insert*].

```

6297   \renewcommand{\do}[1]{%
6298     \expandafter\DeclareRobustCommand\csname ed#1\endcsname[3][1,3,
usedefault]{%
6299       \doedindexlabel%
6300       \csname#1\endcsname[counter=pageline,##1]{##2}[##3]%
6301     }%
6302     \expandafter\WithSuffix\expandafter\DeclareRobustCommand\csname ed
#1\endcsname*[3][1,3,usedefault]{%
6303       \doedindexlabel%
6304       \csname#1\endcsname*[counter=pageline,##1]{##2}[##3]%
6305     }%
6306   }%
6307   \docsvlist{%
6308     gls,%
6309     Gls,%
6310     GLS,%
6311     glspl,%
6312     Glspl,%
6313     GLSpl,%
6314     glstext,%
6315     Glstext,%
6316     GLStext,%
6317     Glsfirst,%
6318     GLSfirst,%
6319     glsplural,%
6320     Glsplural,%
6321     GLSplural,%
6322     glsfirstplural,%
6323     Glsfirstplural,%
6324     GLSfirstplural,%
6325     glsname,%
6326     Glsname,%
6327     GLSname,%
6328     glssymbol,%
6329     Glsymbol,%
6330     GLSsymbol,%
6331     glsdesc,%
6332     Glsdesc,%
6333     GLSdesc,%
6334     glsuseri,%

```

```

6335     Glsuseri,%
6336     GLSuseri,%
6337     glsuserii,%
6338     Glsuserii,%
6339     GLSuserii,%
6340     glsuseriii,%
6341     Glsuseriii,%
6342     GLSuseriii,%
6343     glsuseriv,%
6344     Glsuseriv,%
6345     GLSuseriv,%
6346     glsuserv,%
6347     Glsuserv,%
6348     GLSuserv,%
6349     glsuservi,%
6350     Glsuservi,%
6351     GLSuservi%
6352     }%
6353 %

```

First those which arguments are [*options*]{*label*}{*link text*}.

```

6354     \renewcommand{\do}[1]{%
6355         \expandafter\DeclareRobustCommand\csname ed#1\endcsname[3][1,
usedefault]{%
6356             \doedindexlabel%
6357             \csname#1\endcsname[counter=pageline,##1]{##2}{##3}%
6358             }%
6359         \expandafter\WithSuffix\expandafter\DeclareRobustCommand\csname ed
#1\endcsname*[3][1,usedefault]{%
6360             \doedindexlabel%
6361             \csname#1\endcsname*[counter=pageline,##1]{##2}{##3}%
6362             }%
6363         }%
6364         \docsvlist{glstdisp,glslink}%
6365     }{}%
6366 }%
6367 %

```

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.

```
6368 \def\@hangingsymbol{}
6369 \newcommand*{\sethangingsymbol}[1]{%
6370   \gdef\@hangingsymbol{#1}%
6371 }%
6372 \newif\ifinstanza
6373 %
```

`\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.

```
6374 \newif\ifinserthangingsymbol
6375 \newcommand{\inserthangingsymbol}{%
6376   \ifinserthangingsymbol%
6377     \ifinstanza%
6378       \@hangingsymbol%
6379     \fi%
6380   \fi%
6381 }
6382 %
```

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.

```
6383 \newcommand*{\ampersand}{\char`\&}
6384
6385 %
```

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.

```
6386 \chardef\body=\catcode`\@
6387 \catcode`\@=11
6388 \chardef\next=\catcode`\&
6389 \catcode`\&=\active
6390
6391 %
```

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.

```
6392 \newcount\stanza@count
6393 \newlength{\stanzaindentbase}
6394 \setlength{\stanzaindentbase}{20pt}
6395
6396 %
```

`\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.

```
6397 \def\strip@szacnt#1,#2|{\def\@tempb{#1}\def\@tempa{#2|}}
6398 \newcommand*\setstanzavalues[2]{\def\@tempa{#2,,|}%
6399   \stanza@count\z@
6400   \def\next{\expandafter\strip@szacnt\@tempa
6401     \ifx\@tempb\empty\let\next\relax\else
6402       \expandafter\mathchardef\csname #1@number\stanza@count
6403       \@endcsname\@tempb\relax
6404       \advance\stanza@count\@ne\fi\next}%
6405   \next}
6406
6407 %
```

`\setstanzaindents` In the original edmac, `\setstanzavalues{sza}{⟨...⟩}` had to be called to set the in-
`\setstanzapenalties` dents, and similarly `\setstanzavalues{szp}{⟨...⟩}` to set the penalties. `\setstanzaindents` and `\setstanzapenalties` macros are a convenience to give the user one less thing to worry about (misspelling the first argument).

```
6408 \newcommand*\setstanzaindents[1]{\setstanzavalues{sza}{#1}}
6409 \newcommand*\setstanzapenalties[1]{\setstanzavalues{szp}{#1}}
6410 %
6411 %
```

`\managestanza@modulo` Since version 0.13, the `stanzaindentrepetition` 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 `stanzaindentrepetition` counter, the command restarts it.

```
6412 \newcounter{stanzaindentrepetition}
6413 \newcount\stanza@modulo
6414
```

```

6415 \newcommand*{\managestanza@modulo}[0]{%
6416   \advance\stanza@modulo\@ne%
6417   \ifnum\stanza@modulo>\value{stanzaindentsrepetition}%
6418     \stanza@modulo\@ne%
6419   \fi%
6420 }
6421 %

```

\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.

```

6422 \newcommand{\stanzaindent}[1]{%
6423   \hspace{\dimexpr#1\stanzaindentbase-\parindent\relax}%
6424   \ignorespaces%
6425 }%
6426 \WithSuffix\newcommand\stanzaindent*[1]{%
6427   \stanzaindent{#1}%
6428   \global\advance\stanza@modulo-\@ne%
6429   \ifnum\stanza@modulo=0%
6430     \global\stanza@modulo=\value{stanzaindentsrepetition}%
6431   \fi%
6432   \ignorespaces%
6433 }%
6434 %

```

XXVIII.5 Numbering stanza

Here, macro for numbering stanza. First, the stanza counter.

```

\thestanza 35 \newcounter{stanza}
6436 \renewcommand{\thestanza}{%
6437   \textbf{\arabic{stanza}}%
6438 }
6439 %

```

\ifnumberstanza Then, macro to activate automatically numbering of stanza.

```

6440 \newif\ifnumberstanza%
6441 %

```

\@insertstanzanumber Now, macro called at the first line of of verse of a stanza.

```

6442 \newcommand{\@insertstanzanumber}[0]{%
6443   \ifnumberstanza%
6444     \ifl@dpairing%
6445       \ifledRcol%
6446         \stanzanumwrapper{\thestanzaR}%
6447       \else%

```

```

6448     \stanzanumwrapper{\thestanzaL}%
6449     \fi%
6450     \else%
6451     \stanzanumwrapper{\thestanza}%
6452     \fi%
6453     \setline{1}%
6454     \fi%
6455 }%
6456 %

```

`\advancestanza` Also a command to advance the counter of stanza.

```

6457 \newcommand{\advancestanza}[0]{%
6458   \ifnumberstanza%
6459   \ifl@dpairing%
6460   \ifledRcol%
6461   \addtocounter{stanzaR}{1}%
6462   \else%
6463   \addtocounter{stanzaL}{1}%
6464   \fi%
6465   \else%
6466   \addtocounter{stanza}{1}%
6467   \fi%
6468   \fi%
6469 }%
6470 %

```

`\stanzanumwrapper` And finally, the wrapper for stanza number

```

6471 \newcommand{\stanzanumwrapper}[1]{%
6472   \flagstanza{#1}%
6473 }%
6474 %

```

XXVIII.6 Stanza number in note

Here, the command called when printing stanza number in notes.

```

6475 \newcommand{\printstanza}[0]{%
6476   \ifboolexpr{bool{l@dpairing} or bool{l@dprintingpages} or bool{
6477     l@dprintingcolumns}}{%
6478     \ifledRcol%
6479     \thestanzaR%
6480     \else%
6481     \thestanzaL%
6482     \fi%
6483   }{%
6484     \thestanza%
6485   }%

```



```
6485 }
6486 %
```

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.

```
6487 \newcommandx{\stanza@line}[1][1]{
6488   \ifnum\value{stanzaindentrepetition}=0
6489     \parindent=\csname sza@number\stanza@count
6490       @\endcsname\stanzaindentbase
6491   \else
6492     \parindent=\csname sza@number\stanza@modulo
6493       @\endcsname\stanzaindentbase
6494     \managestanza@modulo
6495   \fi
6496   \pstart[#1]\stanza@hang\ignorespaces}
6497 \xdef\stanza@hang{\noexpand\leavevmode\noexpand\startlock
6498   \hangindent\expandafter
6499   \noexpand\csname sza@0@\endcsname\stanzaindentbase
6500   \hangafter\@ne}
6501 \def\sza@penalty{\count@\csname szp@number\stanza@count @\endcsname
6502   \ifnum\count@>\@M\advance\count@-\@M\penalty-\else
6503   \penalty\fi\count@}
6504 %
```

`\@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 `\&`.

```
6505 \xdef\@startstanza[#1]{%
6506   \noexpand\instanzatrue\expandafter
6507   \begingroup%
6508   \catcode`\noexpand\&\active%
6509   \global\stanza@count\@ne\stanza@modulo\@ne
6510   \noexpand\ifnum\expandafter\noexpand
```

```

6511 \csname sza@0@\endcsname=\z@\let\noexpand\stanza@hang\relax
6512 \let\noexpand\endlock\relax\noexpand\else\interlinepenalty
6513 \@M\rightskip\z@ plus 1fil\relax\noexpand\fi\noexpand\ifnum
6514 \expandafter\noexpand\csname szp@0@\endcsname=\z@
6515 \let\noexpand\sza@penalty\relax\noexpand\fi%
6516 \def\noexpand&{%
6517     \noexpand\newverse[] []}%
6518 \def\noexpand\&{\noexpand\@stopstanza}%
6519 \noexpand\@advancestanza%
6520 \noexpand\stanza@line[#1]%
6521 \noexpand\@insertstanza%
6522 \let\par\relax\ignorespaces%No paragraph in verses
6523 }
6524
6525 \newcommandx{\stanza}[1][1,usedefault]{%
6526     \ifboolexpr{not test{\ifdefvoid{\at@every@stanza}} and test{\ifstrempy
6527     {#1}}}%
6528     {\@startstanza[\at@every@stanza]}%
6529     {\@startstanza[#1]}%
6530 }%
6531 \newcommandx{\@stopstanza}[1][1,usedefault]{%
6532     \unskip%
6533     \endlock%
6534     \ifboolexpr{not test{\ifdefvoid{\at@every@stop@stanza}} and test{\
6535     ifstrempy{#1}}}%
6536     {\pend[\at@every@stop@stanza]}%
6537     {\pend[#1]}%
6538     \endgroup%
6539     \instanzafalse%
6540 }
6541 \newcommand{\AtEveryStopStanza}[1]{%
6542     \ifstrempy{#1}%
6543     {\xdef\at@every@stop@stanza{}}%
6544     {\gdef\at@every@stop@stanza{#1}}%
6545 }%
6546 \def\at@every@stop@stanza{}%
6547
6548 \newcommand{\AtEveryStanza}[1]{%
6549     \ifstrempy{#1}%
6550     {\xdef\at@every@stanza{}}%
6551     {\gdef\at@every@stanza{#1}}%
6552 }%
6553 \def\at@every@stanza{}%
6554
6555 \newcommandx*{\newverse}[2][1,2,usedefault]{%
6556     \unskip%
6557     \endlock\pend[#1]\sza@penalty\global%

```

```

6559 \advance\stanza@count\@ne\stanza@line[#2]%
6560 }
6561
6562 %

```

\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`.

```

6563 \newcommand*{\flagstanza}[2][\stanzaindentbase]{%
6564 \hskip -#1\llap{#2}\hskip #1\ignorespaces}
6565
6566 %

```

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.

```

6567 \catcode`\&=\next
6568 \catcode`\@=\body
6569 \setstanzavalues{szp}{0}
6570
6571 %

```

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.

```

6572 \newtoks\@emptytoks
6573
6574 %

```

The rest is from `amsmath`.

\l@denvbody A token register to contain the body.

```

6575 \newtoks\l@denbody
6576
6577 %

```

\addtol@denbody `\addtol@denbody{arg}` adds arg to the token register `\l@denbody`.

```

6578 \newcommand{\addtol@denbody}[1]{%
6579   \global\l@denbody\expandafter{\the\l@denbody#1}}
6580
6581 %

```

\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`.

```

6582 \newcommand{\l@dcollect@body}[1]{%
6583   \l@denbody{\expandafter#1\expandafter{\the\l@denbody}}%
6584   \edef\processl@denbody{\the\l@denbody\noexpand\end{\@currenvir}}%
6585   \l@denbody\@emptytoks \def\l@dbegin@stack{b}%
6586   \begingroup
6587     \expandafter\let\csname\@currenvir\endcsname\l@dcollect@@body
6588     \edef\processl@denbody{\expandafter\noexpand\csname\@currenvir\endcsname}%
6589     \processl@denbody%
6590   }%
6591
6592 %

```

\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.

```

6593 \def\l@dpush@begins#1\begin#2{%
6594   \ifx\end#2\else b\expandafter\l@dpush@begins\fi}
6595
6596 %

```

\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.

```

6597 \def\l@dcollect@@body#1\end#2{%
6598   \edef\l@dbegin@stack{\l@dpush@begins#1\begin\end
6599     \expandafter\@gobble\l@dbegin@stack}%
6600   \ifx\@empty\l@dbegin@stack

```

```

6601 \endgroup
6602 \@checkend{#2}%
6603 \addtol@denbody{#1}%
6604 \else
6605 \addtol@denbody{#1\end{#2}}%
6606 \fi
6607 \processl@denbody % A little tricky! Note the grouping
6608 }
6609
6610 %

```

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.

```
6611 \newcommand*{\l@dtabnoexpands}{%
6612   \let\rtab=0%
6613   \let\ctab=0%
6614   \let\ltab=0%
6615   \let\rtabtext=0%
6616   \let\ltabtext=0%
6617   \let\ctabtext=0%
6618   \let\edbeforetab=0%
6619   \let\edaftertab=0%
6620   \let\edatleft=0%
6621   \let\edatright=0%
6622   \let\edvertline=0%
6623   \let\edvertdots=0%
6624   \let\edrowfill=0%
6625 }
6626
6627 %
```

`\disable@familiarnotes` Macros to disable and restore familiar notes, to prevent them from printing multiple times in `edtabularx` and `edarrayx` environments.

```
6628 \newcommand{\disable@familiarnotes}{%
6629   \unless\ifnofamiliar%
6630     \def\do##1{%
6631       \csletcs{footnote@##1}{footnote##1}%
6632       \expandafter\renewcommand \csname footnote##1\endcsname[1]{%
6633         \protected@csxdef{@thefnmark##1}{\csuse{thefootnote##1}}%
6634         \csuse{@footnotemark##1}%
6635       }%
6636     }
```

```

6636 }%
6637 \dolistloop{\@series}%
6638 \fi%
6639 }%
6640 \newcommand{\restore@familiarnotes}{%
6641 \unless\ifnofamiliar@%
6642 \def\do##1{%
6643 \csletcs{footnote##1}{footnote@@##1}%
6644 }%
6645 \dolistloop{\@series}%
6646 \fi%
6647 }%
6648
6649 %

```

\disable@sidenotes The same, for side notes.

```

\restore@sidenotes
6650 \newcommand{\disable@sidenotes}{%
6651 \let\@@ledrightnote\ledrightnote%
6652 \let\@@ledleftnote\ledleftnote%
6653 \let\@@ledsidenote\ledsidenote%
6654 \let\ledrightnote@gobble%
6655 \let\ledleftnote@gobble%
6656 \let\ledsidenote@gobble%
6657 }%
6658 \newcommand{\restore@sidenotes}{%
6659 \let\ledrightnote\@@ledrightnote%
6660 \let\ledleftnote\@@ledleftnote%
6661 \let\ledsidenote\@@ledsidenote%
6662 }%
6663 %

```

\disable@notes Disable/restore side and familiar notes.

```

\restore@notes
6664 \newcommand{\disable@notes}{%
6665 \disable@sidenotes%
6666 \disable@familiarnotes%
6667 }%
6668 \newcommand{\restore@notes}{%
6669 \restore@sidenotes%
6670 \restore@familiarnotes%
6671 }%
6672 %

```

\EDTEXT We need to be able to modify the `\edtext` macros and also restore their original definitions.

```

\edtext
6673 \let\EDTEXT=\edtext
6674 \newcommand{\edtext}[2]{\EDTEXT{#1}{#2}}
6675 %

```


`\EDLABEL` We need to be able to modify and restore the `\edlabel` macro.

```
\xedlabel
6676 \let\EDLABEL=\edlabel
6677 \newcommand*{\xedlabel}[1]{\EDLABEL{#1}}
6678 %
```

`\EDINDEX` Macros supporting modification and restoration of `\edindex`.

```
\xedindex
\nulledindex
6679 \let\EDINDEX=\edindex
6680 \newcommand{\xedindex}{\@bsphack%
6681 \ifnextchar [{\l@d@index}{\l@d@index[\jobname]}}
6682 \newcommand{\nulledindex}[2][\jobname]{\@bsphack\@esphack}
6683
6684 %
```

`\@line@num` Macro supporting restoration of `\linenum`.

```
6685 \let\@line@num=\linenum
6686 %
```

`\l@dgobblearg` `\l@dgobbleoptarg[⟨arg⟩]{⟨arg⟩}` replaces these two arguments (first is optional) by `\relax`.

```
6687 \newcommand*{\l@dgobbleoptarg}[2][\relax]{\relax}%
6688
6689 %
```

```
\Relax90 \let\Relax=\relax
```

```
\NEXT91 \let\NEXT=\next
```

```
6692
6693 %
```

`\l@dmodforedtext` Modify and restore various macros for when `\edtext` is used.

```
\l@drestoreforedtext
6694 \newcommand{\l@dmodforedtext}{%
6695 \let\edtext\relax
6696 \def\do##1{\global\csletcs{##1footnote}{\l@dgobbleoptarg}}%
6697 \dolistloop{\@series}%
6698 \let\edindex\nulledindex
6699 \let\linenum\@gobble}
6700 \newcommand{\l@drestoreforedtext}{%
6701 \def\do##1{\global\csletcs{##1footnote}{##1@@footnote}}
6702 \dolistloop{\@series}%
6703 \let\edindex\xedindex}
6704 %
```

`\l@dnullfills` Nullify and restore some column fillers, etc.

`\l@drestorefills`

```

6705 \newcommand{\l@dnnullfills}{%
6706   \def\edlabel##1{%
6707     \def\edrowfill##1##2##3{%
6708     }
6709 \newcommand{\l@drestorefills}{%
6710   \def\edrowfill##1##2##3{\@EDROWFILL@{##1}{##2}{##3}}%
6711 }
6712
6713 %

```

\letsforverteilen Gathers some lets and other code that is common to the **verteilen** macros.

```

6714 \newcommand{\letsforverteilen}{%
6715   \let\edtext\xedtext
6716   \let\edindex\xedindex
6717   \def\do##1{\global\csletcs{##1footnote}{##1@footnote}}
6718   \dolistloop{\@series}%
6719   \let\linenum\@line@num
6720   \hilfe skip=\l@dcolwidth%
6721   \advance\hilfe skip by -\wd\hilfe box
6722   \def\edlabel##1{\xedlabel{##1}}
6723
6724 %

```

\disablel@dtabfeet Declarations for using or using \edtext inside tabulars. The default at this point is for

\enablel@dtabfeet \edtext.

```

6725 \newcommand\disablel@dtabfeet{\l@dmodforedtext}%
6726 \newcommand\enablel@dtabfeet{\l@drestoreforedtext}%
6727 %

```

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.

```

6728 \newcount\l@dampcount
6729 \l@dampcount=1\relax
6730 \newcount\l@dcolcount
6731 \l@dcolcount=0\relax
6732
6733 %

```

\hilfe box Some (temporary) helper items.

```

6734 \hilfe skip
6735 \hilfe box
6736 \hilfe count
6737 \newbox\hilfe box
6738 \newskip\hilfe skip
6739 \newcount\hilfe count

```

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).

```

6740 \newdimen\dcoli
6741 \newdimen\dcolii
6742 \newdimen\dcoliii
6743 \newdimen\dcoliv
6744 \newdimen\dcolv
6745 \newdimen\dcolvi
6746 \newdimen\dcolvii
6747 \newdimen\dcolviii
6748 \newdimen\dcolix
6749 \newdimen\dcolx
6750 \newdimen\dcolxi
6751 \newdimen\dcolxii
6752 \newdimen\dcolxiii
6753 \newdimen\dcolxiv
6754 \newdimen\dcolxv
6755 \newdimen\dcolxvi
6756 \newdimen\dcolxvii
6757 \newdimen\dcolxviii
6758 \newdimen\dcolxix
6759 \newdimen\dcolxx
6760 \newdimen\dcolxxi
6761 \newdimen\dcolxxii
6762 \newdimen\dcolxxiii
6763 \newdimen\dcolxxiv
6764 \newdimen\dcolxxv
6765 \newdimen\dcolxxvi
6766 \newdimen\dcolxxvii
6767 \newdimen\dcolxxviii
6768 \newdimen\dcolxxix
6769 \newdimen\dcolxxx
6770 \newdimen\dcolerr % added for error handling
6771 %
6772 %

```

`\l@dcolwidth` This is a cunning way of storing the columnwidths indexed by the column number `\l@dcolcount`, like an array. (was `\Dimenzuordnung`)

```

6773 \newcommand{\l@dcolwidth}{\ifcase \the\l@dcolcount \dcoli %???
6774 \or \dcoli \or \dcolii \or \dcoliii
6775 \or \dcoliv \or \dcolv \or \dcolvi
6776 \or \dcolvii \or \dcolviii \or \dcolix \or \dcolx
6777 \or \dcolxi \or \dcolxii \or \dcolxiii
6778 \or \dcolxiv \or \dcolxv \or \dcolxvi
6779 \or \dcolxvii \or \dcolxviii \or \dcolxix \or \dcolxx
6780 \or \dcolxxi \or \dcolxxii \or \dcolxxiii
6781 \or \dcolxxiv \or \dcolxxv \or \dcolxxvi
6782 \or \dcolxxvii \or \dcolxxviii \or \dcolxxix \or \dcolxxx

```

```

6783 \else \dcolerr \fi}
6784
6785 %

```

\step1@dcolcount This increments the column counter, and issues an error message if it is too large.

```

6786 \newcommand*{\step1@dcolcount}{\advance\l@dcolcount\@ne
6787 \ifnum\l@dcolcount>30\relax
6788 \led@err@TooManyColumns
6789 \fi}
6790
6791 %

```

\l@dsetmaxcolwidth Sets the column width to the maximum value seen so far.

```

6792 \newcommand{\l@dsetmaxcolwidth}{%
6793 \ifdim\l@dcolwidth < \wd\hilfsbox
6794 \l@dcolwidth = \wd\hilfsbox
6795 \else \relax \fi}
6796
6797 %

```

\measuremcell Measure (recursively) the width required for a math cell.

```

6798 \def\measuremcell #1{%
6799 \ifx #1\ \ifnum\l@dcolcount=0\let\NEXT\relax%
6800 \else\l@dcheckcols%
6801 \l@dcolcount=0%
6802 \let\NEXT\measuremcell%
6803 \fi%
6804 \else\setbox\hilfsbox=\hbox{$\displaystyle{#1}$}%
6805 \step1@dcolcount%
6806 \l@dsetmaxcolwidth%
6807 \let\NEXT\measuremcell%
6808 \fi\NEXT}
6809
6810 %

```

\measuretcell Measure (recursively) the width required for a text cell.

```

6811 \def\measuretcell #1{%
6812 \ifx #1\ \ifnum\l@dcolcount=0\let\NEXT\relax%
6813 \else\l@dcheckcols%
6814 \l@dcolcount=0%
6815 \let\NEXT\measuretcell%
6816 \fi%
6817 \else\setbox\hilfsbox=\hbox{#1}%
6818 \step1@dcolcount%
6819 \l@dsetmaxcolwidth%
6820 \let\NEXT\measuretcell%

```

```

6821 \fi\NEXT}
6822
6823 %

```

\measuremrow Measure (recursively) the width required for a math row.

```

6824 \def\measuremrow #1\{\{%
6825 \ifx #1&\let\NEXT\relax%
6826 \else\measuremcell #1&\&\&%
6827 \let\NEXT\measuremrow%
6828 \fi\NEXT}
6829 %

```

\measuretrow Measure (recursively) the width required for a text row.

```

6830 \def\measuretrow #1\{\{%
6831 \ifx #1&\let\NEXT\relax%
6832 \else\measuretcell #1&\&\&%
6833 \let\NEXT\measuretrow%
6834 \fi\NEXT}
6835
6836 %

```

\edtabcolsep The length \edtabcolsep controls the distance between columns.

```

6837 \newskip\edtabcolsep
6838 \global\edtabcolsep=10pt
6839
6840 %

```

```

\variab41 \newcommand{\variab}{\relax}
6842
6843 %

```

\l@dcheckcols Check that the number of columns is consistent.

```

6844 \newcommand*{\l@dcheckcols}{\%
6845 \ifnum\l@dcolcount=1\relax
6846 \else
6847 \ifnum\l@dampcount=1\relax
6848 \else
6849 \ifnum\l@dcolcount=\l@dampcount\relax
6850 \else
6851 \l@d@err@UnequalColumns
6852 \fi
6853 \fi
6854 \l@dampcount=\l@dcolcount
6855 \fi}
6856
6857 %

```

`\edfilldimen` A length.

```
6858 \newdimen\edfilldimen
6859 \edfilldimen=0pt
6860
6861 %
```

`\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).

`\theaddcolcount`

```
6862 \newcounter{addcolcount}
6863 \renewcommand{\theaddcolcount}{\romannumeral \c@addcolcount}
6864 %
```

XXIX.2.3 Tabular typesetting

`\setmcellright` Typeset (recursively) cells of display math right justified.

```
6865 \def\setmcellright #1{\def\edlabel##1{}%
6866     \let\edindex\nulledindex
6867     \ifx #1\\ \ifnum\l@dcolcount=0%\removelastskip
6868         \let\Next\relax%
6869     \else\l@dcolcount=0%
6870         \let\Next=\setmcellright%
6871     \fi%
6872 \else%
6873     \disablel@dtabfeet%
6874     \step1@dcolcount%
6875     \disable@notes%
6876     \setbox\hilfsbox=\hbox{$\displaystyle{#1}$}%
6877     \restore@notes%
6878     \letsforverteilen%
6879     \hskip\hilfsskip$\displaystyle{#1}$%
6880     \hskip\edtabcolsep%
6881     \let\Next=\setmcellright%
6882 \fi\Next}
6883
6884 %
```

`\settcellright` Typeset (recursively) cells of text right justified.

```
6885 \def\settcellright #1{\def\edlabel##1{}%
6886     \let\edindex\nulledindex
6887     \ifx #1\\ \ifnum\l@dcolcount=0%\removelastskip
6888         \let\Next\relax%
6889     \else\l@dcolcount=0%
6890         \let\Next=\settcellright%
6891     \fi%
6892 \else%
```

```

6893         \disablel@dtabfeet%
6894         \stepl@dc colcount%
6895         \disable@notes%
6896         \setbox\hilfsbox=\hbox{#1}%
6897         \restore@notes%
6898         \letsforverteilen%
6899         \hskip\hilfsskip#1%
6900         \hskip\edtabcolsep%
6901         \let\Next=\settcellright%
6902     \fi\Next}
6903 %

```

\setmcellleft Typeset (recursively) cells of display math left justified.

```

6904 \def\setmcellleft #1&{\def\edlabel##1{}}%
6905     \let\edindex\nulledindex
6906     \ifx #1\\ \ifnum\l@dc colcount=0 \let\Next\relax%
6907         \else\l@dc colcount=0%
6908             \let\Next=\setmcellleft%
6909         \fi%
6910     \else \disablel@dtabfeet%
6911         \stepl@dc colcount%
6912         \disable@notes%
6913         \setbox\hilfsbox=\hbox{$\displaystyle{#1}$}%
6914         \restore@notes%
6915         \letsforverteilen%
6916         $\displaystyle{#1}$\hskip\hilfsskip\hskip\edtabcolsep%
6917         \let\Next=\setmcellleft%
6918     \fi\Next}
6919 %
6920 %

```

\settcellleft Typeset (recursively) cells of text left justified.

```

6921 \def\settcellleft #1&{\def\edlabel##1{}}%
6922     \let\edindex\nulledindex
6923     \ifx #1\\ \ifnum\l@dc colcount=0 \let\Next\relax%
6924         \else\l@dc colcount=0%
6925             \let\Next=\settcellleft%
6926         \fi%
6927     \else \disablel@dtabfeet%
6928         \stepl@dc colcount%
6929         \disable@notes%
6930         \setbox\hilfsbox=\hbox{#1}%
6931         \restore@notes%
6932         \letsforverteilen%
6933         #1\hskip\hilfsskip\hskip\edtabcolsep%
6934         \let\Next=\settcellleft%
6935     \fi\Next}
6936 %

```

\setmcellcenter Typeset (recursively) cells of display math centered.

```

6937 \def\setmcellcenter #1&{\def\edlabel##1{}%
6938   \let\edindex\nulledindex
6939   \ifx #1\\ \ifnum\l@dc@lcount=0\let\Next\relax%
6940     \else\l@dc@lcount=0%
6941       \let\Next=\setmcellcenter%
6942     \fi%
6943   \else \disablel@dtabfeet%
6944     \step1@dc@lcount%
6945     \disable@notes%
6946     \setbox\hilfsbox=\hbox{$\displaystyle{#1}$}%
6947     \restore@notes%
6948     \letsforverteilen%
6949     \hskip 0.5\hilfsskip$\displaystyle{#1}$\hskip0.5\hilfsskip%
6950     \hskip\edtabcolsep%
6951     \let\Next=\setmcellcenter%
6952   \fi\Next}
6953
6954 %

```

\settcclcenter Typeset (recursively) cells of text centered.

```

6955 \def\settcclcenter #1&{\def\edlabel##1{}%
6956   \let\edindex\nulledindex
6957   \ifx #1\\ \ifnum\l@dc@lcount=0 \let\Next\relax%
6958     \else\l@dc@lcount=0%
6959       \let\Next=\settcclcenter%
6960     \fi%
6961   \else \disablel@dtabfeet%
6962     \step1@dc@lcount%
6963     \disable@notes%
6964     \setbox\hilfsbox=\hbox{#1}%
6965     \restore@notes%
6966     \letsforverteilen%
6967     \hskip 0.5\hilfsskip #1\hskip 0.5\hilfsskip%
6968     \hskip\edtabcolsep%
6969     \let\Next=\settcclcenter%
6970   \fi\Next}
6971
6972 %

```

\NEXT \let\NEXT=\relax

```

6974
6975 %

```

\setmrowright Typeset (recursively) rows of right justified math.

```

6976 \def\setmrowright #1\\{

```



```

6977 \ifx #1& \let\NEXT\relax
6978 \else \centerline{\setmcellright #1&\\&\\&}
6979 \let\NEXT=\setmrowright
6980 \fi\NEXT}
6981 %

```

\settroright Typeset (recursively) rows of right justified text.

```

6982 \def\settroright #1\\{%
6983 \ifx #1& \let\NEXT\relax
6984 \else \centerline{\settcclright #1&\\&\\&}
6985 \let\NEXT=\settroright
6986 \fi\NEXT}
6987
6988 %

```

\setmrowleft Typeset (recursively) rows of left justified math.

```

6989 \def\setmrowleft #1\\{%
6990 \ifx #1& \let\NEXT\relax
6991 \else \centerline{\setmcellleft #1&\\&\\&}
6992 \let\NEXT=\setmrowleft
6993 \fi\NEXT}
6994 %

```

\settrorleft Typeset (recursively) rows of left justified text.

```

6995 \def\settrorleft #1\\{%
6996 \ifx #1& \let\NEXT\relax
6997 \else \centerline{\settcclleft #1&\\&\\&}
6998 \let\NEXT=\settrorleft
6999 \fi\NEXT}
7000
7001 %

```

\setmrowcenter Typeset (recursively) rows of centered math.

```

7002 \def\setmrowcenter #1\\{%
7003 \ifx #1& \let\NEXT\relax%
7004 \else \centerline{\setmcellcenter #1&\\&\\&}
7005 \let\NEXT=\setmrowcenter
7006 \fi\NEXT}
7007 %

```

\settrorcenter Typeset (recursively) rows of centered text.

```

7008 \def\settrorcenter #1\\{%
7009 \ifx #1& \let\NEXT\relax
7010 \else \centerline{\settcclcenter #1&\\&\\&}
7011 \let\NEXT=\settrorcenter

```

```

7012 \fi\NEXT}
7013
7014 %

```

```

\newcommand{\nullsetzen}{%
7015
7016 \step1@dcolcount%
7017 \l@dcolwidth=0pt%
7018 \ifnum\l@dcolcount=30\let\NEXT\relax%
7019 \l@dcolcount=0\relax
7020 \else\let\NEXT\nullsetzen%
7021 \fi\NEXT}
7022
7023 %

```

\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.

```

7024 \newcommand{\edatleft}[3][\@empty]{%
7025 \ifx#1\@empty
7026 \vbox to 10pt{\vss\hbox{$\left#2\vrule width0pt height #3
7027 \depth 0pt \right. $\hss}\vfil}
7028 \else
7029 \vbox to 4pt{\vss\hbox{$#1\left#2\vrule width0pt height #3
7030 \depth 0pt \right. $\}\vfil}
7031 \fi}
7032 %

```

\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.

```

7033 \newcommand{\edatright}[3][\@empty]{%
7034 \ifx#1\@empty
7035 \vbox to 10pt{\vss\hbox{$\left.\vrule width0pt height #3
7036 \depth 0pt \right#2 $\hss}\vfil}
7037 \else
7038 \vbox to 4pt{\vss\hbox{$\left.\vrule width0pt height #3
7039 \depth 0pt \right#2 #1 $\}\vfil}
7040 \fi}
7041
7042 %

```

\edvertline `\edvertline{ $\langle len \rangle$ }` vertical line $\langle len \rangle$ high.

```

7043 \newcommand{\edvertline}[1]{\vbox to 8pt{\vss\hbox{\vrule height #1}\vfil}}
7044
7045 %

```

\edvertdots `\edvertdots{ $\langle len \rangle$ }` vertical dotted line $\langle len \rangle$ high.

```

7046 \newcommand{\edvertdots}[1]{\vbox to 1pt{\vss\vbox to #1%
7047     {\cleaders\hbox{$\m@th\hbox{.}\vbox to 0.5em{ }$}\vfil}}}
7048
7049 %

```

\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@`.

```

7050 \newcommand{\l@dtabaddcols}[2]{%
7051     \l@dcheckstartend{#1}{#2}%
7052     \ifl@dstartendok
7053     \setcounter{addcolcount}{#1}%
7054     \@whilenum \value{addcolcount}<#2\relax \do
7055     {\advance\edfilldimen by \the\csname dcol\theaddcolcount\endcsname
7056     \advance\edfilldimen by \edtabcolsep
7057     \stepcounter{addcolcount}}%
7058     \advance\edfilldimen by \the\csname dcol\theaddcolcount\endcsname
7059     \fi
7060 }
7061
7062 %

```

\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.

```

7063 \newif\ifl@dstartendok
7064 \newcommand{\l@dcheckstartend}[2]{%
7065     \l@dstartendoktrue
7066     \ifnum #1<\@ne
7067     \l@dstartendokfalse
7068     \led@err@LowStartColumn
7069     \fi
7070     \ifnum #2>30\relax
7071     \l@dstartendokfalse
7072     \led@err@HighEndColumn
7073     \fi
7074     \ifnum #1>#2\relax
7075     \l@dstartendokfalse
7076     \led@err@ReverseColumns
7077     \fi
7078 }
7079
7080 %

```

\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.

```

7081 \newcommand*\edrowfill}[3]{%
7082   \l@dtabaddcols{#1}{#2}%
7083   \hb@xt@ \the\l@dcolwidth{\hb@xt@ \the\edfilldimen{#3}\hss}}
7084 \let\@edrowfill=\edrowfill
7085 \def\@EDROWFILL@#1#2#3{\@edrowfill@{#1}{#2}{#3}}
7086
7087 %

```

\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`.

```

7088 \newcommand{\leftltab}[1]{%
7089   \hb@xt@ \z@{\vbox{\edtabindent%
7090     \moveleft\Hilfsskip\hbox{\ #1}}\hss}}
7091
7092 %

```

\leftrtab `\leftrtab{<text>}{<math>}` for `\edbeforetab` in `\rtab`.

```

7093 \newcommand{\leftrtab}[2]{%
7094   #2\hb@xt@ \z@{\vbox{\edtabindent%
7095     \advance\Hilfsskip by\dcoli%
7096     \moveleft\Hilfsskip\hbox{\ #1}}\hss}}
7097
7098 %

```

\leftctab `\leftctab{<text>}{<math>}` for `\edbeforetab` in `\ctab`.

```

7099 \newcommand{\leftctab}[2]{%
7100   \hb@xt@ \z@{\vbox{\edtabindent\l@dcolcount=\l@dampcount%
7101     \advance\Hilfsskip by 0.5\dcoli%
7102     \setbox\hilfsbox=\hbox{\def\edlabel##1{}}%
7103     \disablel@dtabfeet$\displaystyle{#2}$}%
7104     \advance\Hilfsskip by -0.5\wd\hilfsbox%
7105     \moveleft\Hilfsskip\hbox{\ #1}}\hss}%
7106   #2}
7107
7108 %

```

\rightctab `\rightctab{<math>}{<text>}` for `\edaftertab` in `\ctab`.

```

7109 \newcommand{\rightctab}[2]{%
7110   \setbox\hilfsbox=\hbox{\def\edlabel##1{}}%
7111   \disablel@dtabfeet#2\l@dampcount=\l@dcolcount%
7112   #1\hb@xt@ \z@{\vbox{\edtabindent\l@dcolcount=\l@dampcount%

```

```

7113 \advance\Hilfsskip by 0.5\l@dcwidth%
7114 \advance\Hilfsskip by -\wd\hilfsbox%
7115 \setbox\hilfsbox=\hbox{\def\edlabel##1{}}%
7116 \disablel@dtabfeet$\displaystyle{#1}$}%
7117 \advance\Hilfsskip by -0.5\wd\hilfsbox%
7118 \advance\Hilfsskip by \edtabcolsep%
7119 \moveright\Hilfsskip\hbox{ #2}}\hss}%
7120 }
7121
7122 %

```

\rightltab `\rightltab{<math><math>}{<text>}` for `\edaftertab` in `\ltab`.

```

7123 \newcommand{\rightltab}[2]{%
7124 \setbox\hilfsbox=\hbox{\def\edlabel##1{}}%
7125 \disablel@dtabfeet#2}\l@dampcount=\l@dcwidth%
7126 #1\hb@xt@#2{\vbox{\edtabindent\l@dcwidth=\l@dampcount%
7127 \advance\Hilfsskip by\l@dcwidth%
7128 \advance\Hilfsskip by-\wd\hilfsbox%
7129 \setbox\hilfsbox=\hbox{\def\edlabel##1{}}%
7130 \disablel@dtabfeet$\displaystyle{#1}$}%
7131 \advance\Hilfsskip by-\wd\hilfsbox%
7132 \advance\Hilfsskip by\edtabcolsep%
7133 \moveright\Hilfsskip\hbox{ #2}}\hss}%
7134 }
7135
7136 %

```

\rightrtab `\rightrtab{<math><math>}{<text>}` for `\edaftertab` in `\rtab`.

```

7137 \newcommand{\rightrtab}[2]{%
7138 \setbox\hilfsbox=\hbox{\def\edlabel##1{}}%
7139 \disablel@dtabfeet#2}%
7140 #1\hb@xt@#2{\vbox{\edtabindent%
7141 \advance\Hilfsskip by-\wd\hilfsbox%
7142 \advance\Hilfsskip by\edtabcolsep%
7143 \moveright\Hilfsskip\hbox{ #2}}\hss}%
7144 }
7145
7146 %

```

\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.

```

7147 \newcommand{\rtab}[1]{%
7148 \l@dnnullfills
7149 \def\edbeforetab##1##2{\lefttab{##1}{##2}}%
7150 \def\edaftertab##1##2{\righttab{##1}{##2}}%
7151 \measurembody{#1}%

```

```

7152 \l@drestorefills
7153 \variab
7154 \setmrowright #1\\&\\%
7155 \enablel@dtabfeet}
7156
7157 %

```

\measurembody `\measurembody{⟨body⟩}` measures the array `⟨body⟩`.

```

7158 \newcommand{\measurembody}[1]{%
7159 \disablel@dtabfeet%
7160 \l@dcolcount=0%
7161 \nullsetzen%
7162 \l@dcolcount=0
7163 \measuremrow #1\\&\\%
7164 \global\l@dampcount=1}
7165
7166 %

```

\rtabtext `\rtabtext{⟨body⟩}` typesets `⟨body⟩` as a tabular with the entries right justified.

```

7167 \newcommand{\rtabtext}[1]{%
7168 \l@dnullfills
7169 \measuretbody{#1}%
7170 \l@drestorefills
7171 \variab
7172 \settroright #1\\&\\%
7173 \enablel@dtabfeet}
7174
7175 %

```

\measuretbody `\measuretbody{⟨body⟩}` measures the tabular `⟨body⟩`.

```

7176 \newcommand{\measuretbody}[1]{%
7177 \disable@notes%
7178 \disablel@dtabfeet%
7179 \l@dcolcount=0%
7180 \nullsetzen%
7181 \l@dcolcount=0
7182 \measuretbody #1\\&\\%
7183 \restore@notes%
7184 \global\l@dampcount=1}
7185
7186 %

```

\ltab Array with entries left justified.

```

\edbeforetab
\edaftertab
7187 \newcommand{\ltab}[1]{%
7188 \l@dnullfills
7189 \def\edbeforetab##1##2{\leftltab{##1}{##2}}%

```

```

7190 \def\edaftertab##1##2{\rightltab{##1}{##2}}%
7191 \measuretbody{#1}%
7192 \l@drestorefills
7193 \variab
7194 \setmrowleft #1\\&\\%
7195 \enablel@dtabfeet}
7196
7197 %

```

\ltabtext Tabular with entries left justified.

```

7198 \newcommand{\ltabtext}[1]{%
7199 \l@dnnullfills
7200 \measuretbody{#1}%
7201 \l@drestorefills
7202 \variab
7203 \settrrowleft #1\\&\\%
7204 \enablel@dtabfeet}
7205
7206 %

```

\ctab Array with centered entries.

```

\edbeforetab \newcommand{\ctab}[1]{%
\edaftertab 7207 \l@dnnullfills
7208 \def\edbeforetab##1##2{\leftctab{##1}{##2}}%
7209 \def\edaftertab##1##2{\rightctab{##1}{##2}}%
7210 \measuretbody{#1}%
7211 \l@drestorefills
7212 \variab
7213 \setmrowcenter #1\\&\\%
7214 \enablel@dtabfeet}
7215
7216 %
7217

```

\ctabtext Tabular with entries centered.

```

7218 \newcommand{\ctabtext}[1]{%
7219 \l@dnnullfills
7220 \measuretbody{#1}%
7221 \l@drestorefills
7222 \variab
7223 \settrrowcenter #1\\&\\%
7224 \enablel@dtabfeet}
7225
7226 %

```

```

\spreadtext 7227 \newcommand{\spreadtext}[1]{%\l@dcolcount=\l@dampcount%
7228 \hb@xt@ \the\l@dcolwidth{\hbox{#1}\hss}}
7229 %

```

```

\spreadmath7230 \newcommand{\spreadmath}[1]{%
7231   \hb@xt@ \the\l@dcolwidth{\hbox{$\displaystyle{#1}$}\hss}}
7232
7233 %

```

\HILFSskip More helpers.

```

\Hilfsskip7234 \newskip\HILFSskip
7235 \newskip\Hilfsskip
7236
7237 %

```

```

\EDTABINDENT7238 \newcommand{\EDTABINDENT}{%
7239   \ifnum\l@dcolcount=30\let\NEXT\relax\l@dcolcount=0%
7240   \else\step\l@dcolcount%
7241     \advance\Hilfsskip by\l@dcolwidth%
7242     \ifdim\l@dcolwidth=0pt\advance\hilfscout\@ne
7243     \else\advance\Hilfsskip by \the\hilfscout\edtabcolsep%
7244     \hilfscout=1\fi%
7245     \let\NEXT=\EDTABINDENT%
7246   \fi\NEXT}%
7247 %

```

\edtabindent (was \tabindent)

```

7248 \newcommand{\edtabindent}{%
7249   \l@dcolcount=0\relax
7250   \Hilfsskip=0pt%
7251   \hilfscout=1\relax
7252   \EDTABINDENT%
7253   \hilfsskip=\hsize%
7254   \advance\hilfsskip -\Hilfsskip%
7255   \Hilfsskip=0.5\hilfsskip%
7256   }%
7257
7258 %

```

\EDTAB (was \TAB)

```

7259 \def\EDTAB #1|#2|{%
7260   \setbox\tabhilfbox=\hbox{$\displaystyle{#1}$}%
7261   \setbox\tabHilfbox=\hbox{$\displaystyle{#2}$}%
7262   \advance\tabelskip -\wd\tabhilfbox%
7263   \advance\tabelskip -\wd\tabHilfbox%
7264   \unhbox\tabhilfbox\hskip\tabelskip%
7265   \unhbox\tabHilfbox}%
7266
7267 %

```


`\EDTABtext` (was `\TABtext`)

```

7268 \def\EDTABtext #1|#2|{%
7269   \setbox\tabhilfbox=\hbox{#1}%
7270   \setbox\tabHilfbox=\hbox{#2}%
7271   \advance\tabelskip -\wd\tabhilfbox%
7272   \advance\tabelskip -\wd\tabHilfbox%
7273   \unhbox\tabhilfbox\hskip\tabelskip%
7274   \unhbox\tabHilfbox}%
7275 %

```

`\tabhilfbox` Further helpers.

`\tabHilfbox`

```

7276 \newbox\tabhilfbox
7277 \newbox\tabHilfbox
7278
7279 %

```

XXIX.2.4 Environments

`\edarrayl` `\edarrayc` `\edarrayr` The ‘environment’ forms for `\ltab`, `\ctab` and `\rtab`.

```

7280 \newenvironment{edarrayl}{\l@collect@body\ltab}{}
7281 \newenvironment{edarrayc}{\l@collect@body\ctab}{}
7282 \newenvironment{edarrayr}{\l@collect@body\rtab}{}
7283
7284 %

```

`\edtabularl` `\edtabularc` `\edtabularr` The ‘environment’ forms for `\ltabtext`, `\ctabtext` and `\rtabtext`.

```

7285 \newenvironment{edtabularl}{\l@collect@body\ltabtext}{}
7286 \newenvironment{edtabularc}{\l@collect@body\ctabtext}{}
7287 \newenvironment{edtabularr}{\l@collect@body\rtabtext}{}
7288
7289 %

```

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`

```

7290 \newcommand{\initnumbering@quote}{
7291   \ifnoquotation@else
7292     \renewcommand{\quotation}{\par\leavevmode%
7293                                   \parindent=1.5em%
7294                                   \skipnumbering%

```

```

7295         \ifautopar%
7296             \vskip-\parskip%
7297         \else%
7298             \vskip\topsep%
7299         \fi%
7300         \global\leftskip=\leftmargin%
7301         \global\rightskip=\leftmargin%
7302     }
7303     \renewcommand{\endquotation}{\par%
7304         \global\leftskip=0pt%
7305         \global\rightskip=0pt%
7306         \leavevmode%
7307         \skipnumbering%
7308         \ifautopar%
7309             \vskip-\parskip%
7310         \else%
7311             \vskip\topsep%
7312         \fi%
7313     }
7314     \renewcommand{\quote}{\par\leavevmode%
7315         \parindent=0pt%
7316         \skipnumbering%
7317         \ifautopar%
7318             \vskip-\parskip%
7319         \else%
7320             \vskip\topsep%
7321         \fi%
7322         \global\leftskip=\leftmargin%
7323         \global\rightskip=\leftmargin%
7324     }
7325     \renewcommand{\endquote}{\par%
7326         \global\leftskip=0pt%
7327         \global\rightskip=0pt%
7328         \leavevmode%
7329         \skipnumbering%
7330         \ifautopar%
7331             \vskip-\parskip%
7332         \else%
7333             \vskip\topsep%
7334         \fi%
7335     }
7336     \fi
7337 }
7338 %

```

XXXI Section's title commands

XXXI.1 Commands to disable some feature

`\ledsectnotoc` The `\ledsectnotoc` only disables the `\addcontentsline` macro.

```
7339 \newcommand{\ledsectnotoc}{\let\addcontentsline@gobblethree}
7340 %
```

`\ledsectnomark` The `\ledsectnomark` only disables the `\chaptermark`, `\sectionmark` and `\subsectionmark` macros.

```
7341 \newcommand{\ledsectnomark}{%
7342   \let\chaptermark@gobble%
7343   \let\sectionmark@gobble%
7344   \let\subsectionmark@gobble%
7345 }
7346 %
```

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 \LaTeX is run a other time, this file is read. That:
 - Adds the `pstart` number to a list of `pstarts` where a sectioning command is used.
 - Defines a command, the name of which contains the `pstart` number, and which calls the normal \LaTeX sectioning command.
4. This last command is called when the `pstart` is effectively printed.

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).

```
7347 \catcode`\#=12
7348 \notbool{@noeled@sec}{%
7349 %
```

\beforeeledchapter For technical reasons, not yet solved, page-breaking before chapters can't be made automatically by eledmac. Users have to use \beforeeledchapter.

```
7350 \ifl@dmemoir
7351 \newcommand\beforeeledchapter{%
7352 \clearforchapter%
7353 }
7354 \else
7355 \newcommand\beforeeledchapter{%
7356 \if@openright%
7357 \cleardoublepage%
7358 \else%
7359 \clearpage%
7360 \fi%
7361 }
7362 \fi
7363 %
```

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.

```
7364 \newif\if@eled@sectioning
7365 %
```

\print@leftmargin@eledsection **\print@rightmargin@eledsection** \print@leftmargin@eledsection and \print@rightmargin@eledsection are added by reledmac inside the code of sectioning command, in order to affix lines numbers. They include tests for RTL languages.

```
7366 \def\print@rightmargin@eledsection{%
7367 \if@eled@sectioning%
7368 \begingroup%
7369 \if@RTL%
7370 \let\llap\rlap%
7371 \let\leftlinenum\rightlinenum%
```

```

7372 \let\leftlinenum\rightlinenumR%
7373 \let\l@dld@ta\l@dld@ta%
7374 \let\l@drsn@te\l@dlsn@te%
7375 \fi%
7376 \hfill\l@dld@ta \csuse{LR}{\l@drsn@te}%
7377 \endgroup%
7378 \fi%
7379 }%
7380
7381 \def\print@leftmargin@eledsection{%
7382 \if@eled@sectioning%
7383 \leavevmode%
7384 \begingroup%
7385 \if@RTL%
7386 \let\rlap\llap%
7387 \let\rightlinenum\leftlinenum%
7388 \let\rightlinenumR\leftlinenumR%
7389 \let\l@dld@ta\l@dld@ta%
7390 \let\l@dlsn@te\l@drsn@te%
7391 \fi%
7392 \l@dld@ta\csuse{LR}{\l@dlsn@te}%
7393 \endgroup%
7394 \fi%
7395 }%
7396
7397 %

```

XXXI.5 Patching standard commands

`\chapter` We have to patch \LaTeX , book and memoir sectioning commands in order to:

- Disable `\edtext` inside.
- Disable page breaking (for `\chapter`).
- Add line numbers and sidenotes.

Unfortunately, Maïeul Rouquette was not able to try if memoir is loaded. That is why `eledmac` tries to define for both standard class and memoir class.

```

7398 \AtBeginDocument{%
7399 \patchcmd{\chapter}{\clearforchapter}{%
7400 \if@eled@sectioning\else%
7401 \ifl@dprintingpages\else%
7402 \clearforchapter%
7403 \fi%
7404 \fi%
7405 }
7406 {}
7407 {}

```

```

7408
7409
7410 \pretocmd{\M@sect}
7411   {\let\old@edtext=\edtext%
7412   \let\edtext=\dummy@edtext@showlemma%
7413   }
7414   {}
7415   {}
7416
7417 \apptocmd{\M@sect}
7418   {\let\edtext=\old@edtext}
7419   {}
7420   {}
7421
7422 \patchcmd{\M@sect}
7423   { #9}
7424   { #9%
7425   \print@rightmargin@eledsection%
7426   }
7427   {}
7428   {}
7429
7430 \patchcmd{\M@sect}
7431   {\hskip #3\relax}
7432   {\hskip #3\relax%
7433   \print@leftmargin@eledsection%
7434   }
7435   {}
7436   {}
7437
7438 \patchcmd{\@mem@old@ssect}
7439   {#5}
7440   {#5%
7441   \print@leftmargin@eledsection%
7442   }
7443   {}
7444   {}
7445
7446 \patchcmd{\@mem@old@ssect}
7447   {\hskip #1}
7448   {\hskip #1%
7449   \print@rightmargin@eledsection%
7450   }
7451   {}
7452   {}
7453
7454 \patchcmd{\chapter}{\if@openright\cleardoublepage\else\clearpage\fi}{%
7455   \if@eled@sectioning\else%
7456   \ifl@dprintingpages\else%
7457   \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
    \fi%
7458 \fi%
7459 \fi%
7460 }%
7461 {}%
7462 {}%
7463
7464 \patchcmd{\scr@startchapter}{\if@openright\cleardoublepage\else\clearpage\fi}{%
\if@eled@sectioning\else%
7465 \ifl@dprintingpages\else%
7466 \if@openright\cleardoublepage\else\clearpage\fi}%No clearpage inside a
\Pages: will keep critical notes from printing on the title page. Here for
scrbook.
7468 \fi%
7469 \fi%
7470 }
7471 {}
7472 {}
7473
7474 \patchcmd{\@makechapterhead}
7475 {#1}
7476 {\print@leftmargin@eledsection%
7477 #1%
7478 \print@rightmargin@eledsection%
7479 }
7480 {}
7481 {}
7482
7483 \patchcmd{\@makechapterhead}% For BIDI
7484 {\if@RTL\raggedleft\else\raggedright\fi}%
7485 {\if@eled@sectioning\else%
7486 \if@RTL\raggedleft\else\raggedright\fi%
7487 \fi%
7488 }%
7489 {}%
7490 {}%
7491
7492 \patchcmd{\@makeschapterhead}
7493 {#1}
7494 {\print@leftmargin@eledsection%
7495 #1%
7496 \print@rightmargin@eledsection%
7497 }
7498 {}
7499 {}
7500
7501 \pretocmd{\@sect}
7502 {\let\old@edtext=\edtext

```

```

7503 \let\edtext=\dummy@edtext@showlemma%
7504 }
7505 {}
7506 {}
7507
7508 \apptocmd{\@sect}
7509 {\let\edtext=\old@edtext}
7510 {}
7511 {}
7512
7513 \pretocmd{\@ssect}
7514 {\let\old@edtext=\edtext%
7515 \let\edtext=\dummy@edtext@showlemma%
7516 }
7517 {}
7518 {}
7519
7520 \apptocmd{\@ssect}
7521 {\let\edtext=\old@edtext}
7522 {}
7523 {}
7524
7525 %

```

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.

```

7526 \@ifpackageloaded{nameref}{
7527
7528 \patchcmd{\NR@sect}
7529 {#8}
7530 {#8%
7531 \print@rightmargin@eledsection%
7532 }
7533 {}
7534 {}
7535
7536 \patchcmd{\NR@sect}
7537 {\hskip #3\relax}
7538 {\hskip #3\relax%
7539 \print@leftmargin@eledsection%
7540 }
7541 {}
7542 {}
7543
7544 \patchcmd{\NR@ssect}
7545 {#5}
7546 {#5%
7547 \print@rightmargin@eledsection%
7548 }
7549 {}

```



```

7550 {}
7551
7552 \patchcmd{\NR@ssect}
7553 {\hskip #1}
7554 {\hskip #1%
7555 \print@leftmargin@eledsection%
7556 }
7557 {}
7558 {}
7559 }%
7560 {
7561 \patchcmd{\@sect}
7562 {#8}
7563 {#8%
7564 \print@rightmargin@eledsection%
7565 }
7566 {}
7567 {}
7568
7569 \patchcmd{\@sect}
7570 {\hskip #3\relax}
7571 {\hskip #3\relax%
7572 \print@leftmargin@eledsection%
7573 }
7574 {}
7575 {}
7576
7577 \patchcmd{\@ssect}
7578 {#5}
7579 {#5%
7580 \print@rightmargin@eledsection%
7581 }
7582 {}
7583 {}
7584
7585 \patchcmd{\@ssect}
7586 {\hskip #1}
7587 {\hskip #1%
7588 \print@leftmargin@eledsection%
7589 }
7590 {}
7591 {}
7592 }%
7593 }
7594 %

```

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.

```

7595 {}}%

```

```

7596 \protect\catcode`\#=6 %Space NEEDS by \catcode
7597 \notbool{@noeled@sec}{%
7598 %

```

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.

```

7599 \newwrite\eled@sectioning@out
7600 %

```

`\eledchapter` `\eledsection` And now, the user sectioning commands, which write to the file, and also add content as a “normal” line.

```

\eledsubsection \newcommand{\eledchapter}[2] [] {%
\eledsubsubsection #2%
\eledchapter* \ifledRcol%
\eledsection* \immediate\write\eled@sectioning@out{%
\eledsubsection* \string\eled@chapter{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR}{R}
\eledsubsubsection* }%
7601 \else%
7602 \immediate\write\eled@sectioning@out{%
7603 \string\eled@chapter{#1}{\unexpanded{#2}}{\the\l@dnumpstartsL}{L}
7604 }%
7605 \fi%
7606 }
7607
7608 \newcommand{\eledsection}[2] [] {%
7609 #2%
7610 \ifledRcol%
7611 \immediate\write\eled@sectioning@out{%
7612 \string\eled@section{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR}{R}
7613 }%
7614 \else%
7615 \immediate\write\eled@sectioning@out{%
7616 \string\eled@section{#1}{\unexpanded{#2}}{\the\l@dnumpstartsL}{L}
7617 }%
7618 \fi%
7619 }
7620
7621 \newcommand{\eledsubsection}[2] [] {%
7622 #2%
7623 \ifledRcol%
7624 \immediate\write\eled@sectioning@out{%
7625 \string\eled@subsection{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR}{R}
7626 }%
7627 \else%
7628 \immediate\write\eled@sectioning@out{%
7629 \string\eled@subsection{#1}{\unexpanded{#2}}{\the\l@dnumpstartsL}{L}
7630 }%
7631 \fi%
7632 }
7633

```

```

7634 \immediate\write\eled@sectioning@out{%
7635 \string\eled@subsection{#1}{\unexpanded{#2}}{\the\l@dnumpstartsL}{-}{-}
7636 }%
7637 \fi%
7638 }
7639 \newcommand{\eledsubsubsection}[2][]{%
7640 #2%
7641 \ifledRcol%
7642 \immediate\write\eled@sectioningR@out{%
7643 \string\eled@subsubsection{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR
7644 }{-}{R}
7645 }%
7646 \else%
7647 \immediate\write\eled@sectioning@out{%
7648 \string\eled@subsubsection{#1}{\unexpanded{#2}}{\the\l@dnumpstartsL
7649 }{-}{-}
7650 }%
7651 \fi%
7652 }
7653 \WithSuffix\newcommand\eledchapter*[2][]{%
7654 #2%
7655 \ifledRcol%
7656 \immediate\write\eled@sectioningR@out{%
7657 \string\eled@chapter{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR}{*}{R}
7658 }%
7659 \else%
7660 \immediate\write\eled@sectioning@out{%
7661 \string\eled@chapter{#1}{\unexpanded{#2}}{\the\l@dnumpstartsL}{*}{-}
7662 }%
7663 \fi%
7664 }
7665
7666 \WithSuffix\newcommand\eledsection*[2][]{%
7667 #2%
7668 \ifledRcol%
7669 \immediate\write\eled@sectioningR@out{%
7670 \string\eled@section{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR}{*}{R}
7671 }%
7672 \else%
7673 \immediate\write\eled@sectioning@out{%
7674 \string\eled@section{#1}{\unexpanded{#2}}{\the\l@dnumpstartsL}{*}{-}
7675 }%
7676 \fi%
7677 }
7678
7679 \WithSuffix\newcommand\eledsubsection*[2][]{%
7680 #2%
7681 \ifledRcol%
```

```

7682 \immediate\write\eled@sectioningR@out{%
7683 \string\eled@subsection{#1}{\unexpanded{#2}}{\the\l@dumpstartsR}{*}{
R}
7684 }%
7685 \else%
7686 \immediate\write\eled@sectioning@out{%
7687 \string\eled@subsection{#1}{\unexpanded{#2}}{\the\l@dumpstartsL
}{*}{
7688 }%
7689 \fi%
7690 }
7691
7692 \WithSuffix\newcommand\eledsubsubsection*[2][]{%
7693 #2%
7694 \ifledRcol%
7695 \immediate\write\eled@sectioningR@out{%
7696 \string\eled@subsubsection{#1}{\unexpanded{#2}}{\the\l@dumpstartsR
}{*}{R}
7697 }%
7698 \else%
7699 \immediate\write\eled@sectioning@out{%
7700 \string\eled@subsubsection{#1}{\unexpanded{#2}}{\the\l@dumpstartsL
}{*}{
7701 }%
7702 \fi%
7703 }
7704 %

```

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.

```

7705 \def\eled@chapter#1#2#3#4#5{%
7706 \ifstrempy{#4}%
7707 {%
7708 \ifstrempy{#1}%
7709 {%
7710 \global\csdef{eled@sectioning@#3#5}{\let\edtext=\
dummy@edtext@showlemma\chapter{#2}}%

```

```

7711 \global\csdef{eled@sectmark@#3#5}{\let\edtext=\dummy@edtext{}\
chaptermark{#2}}%
7712 }%Need for \pairs, because of using parbox.
7713 {%
7714 \global\csdef{eled@sectioning@#3#5}{\let\edtext=\
dummy@edtext@showlemma\chapter[#1]{#2}}%
7715 \global\csdef{eled@sectmark@#3#5}{\let\edtext=\dummy@edtext{}\
chaptermark{#2}}%Need for \pairs, because of using parbox.
7716 }%
7717 }%
7718 {%
7719 \ifstrempy{#1}%
7720 {\global\csdef{eled@sectioning@#3#5}{\let\edtext=\
dummy@edtext@showlemma\chapter*{#2}}}%
7721 {\global\csdef{eled@sectioning@#3#5}{\let\edtext=\
dummy@edtext@showlemma\chapter*{#1}{#2}}}%Bug in LaTeX!
7722 }%
7723 \listcsgadd{eled@sections#5@@}{#3}%
7724 }
7725 \def\eled@section#1#2#3#4#5{%
7726 \ifstrempy{#4}%
7727 {\ifstrempy{#1}%
7728 {%
7729 \global\csdef{eled@sectioning@#3#5}{\section{#2}}%
7730 \global\csdef{eled@sectmark@#3#5}{\let\edtext=\dummy@edtext{}\
sectionmark{#2}}%Need for \pairs, because of using parbox.
7731 }%
7732 {%
7733 \global\csdef{eled@sectioning@#3#5}{\section[#1]{#2}}%
7734 \global\csdef{eled@sectmark@#3#5}{\let\edtext=\dummy@edtext{}\
sectionmark{#1}}%Need for \pairs, because of using parbox.
7735 }%
7736 }%
7737 {\ifstrempy{#1}%
7738 {\global\csdef{eled@sectioning@#3#5}{\section*{#2}}}%
7739 {\global\csdef{eled@sectioning@#3#5}{\section*{#1}{#2}}}%Bug in
LaTeX!
7740 }
7741 \listcsgadd{eled@sections#5@@}{#3}%
7742 }
7743 \def\eled@subsection#1#2#3#4#5{%
7744 \ifstrempy{#4}%
7745 {\ifstrempy{#1}%
7746 {%
7747 \global\csdef{eled@sectioning@#3#5}{\subsection{#2}}%
7748 \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)
7749 }%
7750 {%

```

```

7751 \global\csdef{eled@sectioning@#3#5}{\subsection[#1]{#2}}%
7752 \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)
7753 }%
7754 }%
7755 {\ifstrempy{#1}%
7756 {\global\csdef{eled@sectioning@#3#5}{\subsection*{#2}}}%
7757 {\global\csdef{eled@sectioning@#3#5}{\subsection*{#1}{#2}}}%Bug in
LaTeX!
7758 }
7759 \listcsgadd{eled@sections#5@@}{#3}%
7760 }
7761 \def\eled@subsubsection#1#2#3#4#5{%
7762 \ifstrempy{#4}%
7763 {\ifstrempy{#1}%
7764 {\global\csdef{eled@sectioning@#3#5}{\subsubsection{#2}}}%
7765 {\global\csdef{eled@sectioning@#3#5}{\subsubsection[#1]{#2}}}%
7766 }%
7767 {\ifstrempy{#1}%
7768 {\global\csdef{eled@sectioning@#3#5}{\subsubsection*{#2}}}%
7769 {\global\csdef{eled@sectioning@#3#5}{\subsubsection*{#1}{#2}}}%Bug
in LaTeX!
7770 }
7771 \listcsgadd{eled@sections#5@@}{#3}%
7772 }
7773
7774 %

```

End of the conditional test about noeledsec option.

```

7775 }{}
7776 %

```

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.

```

7777 \def\normal@page@break{}
7778 %

```

`\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.

```
7779 \def\l@prev@pb{}
7780 \def\l@prev@nopb{}
7781 %
```

`\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.

```
7782 \newcommand{\ledpb}{\write\linenumout{\string\led@pb}}
7783 \newcommand{\ledpbnum}[1]{\write\linenumout{\string\led@pbnum{#1}}}
7784 \newcommand{\lednopb}{\write\linenumout{\string\led@nopb}}
7785 \newcommand{\lednopbnum}[1]{\write\linenumout{\string\led@nopbnum{#1}}}
7786 %
```

`\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.

```
7787 \newcommand{\led@pb}{\listxadd{\l@prev@pb}{\the\absline@num}}
7788 \newcommand{\led@pbnum}[1]{\listxadd{\l@prev@pb}{#1}}
7789 \newcommand{\led@nopb}{\listxadd{\l@prev@nopb}{\the\absline@num}}
7790 \newcommand{\led@nopbnum}[1]{\listxadd{\l@prev@nopb}{#1}}
7791 %
```

`\ledpbsetting` The `\ledpbsetting` macro only changes the value of `\led@pb@macro`, for which the default value is before.

```
7792 \def\led@pb@setting{before}
7793 \newcommand{\ledpbsetting}[1]{\gdef\led@pb@setting{#1}}
7794 %
```

`\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`.

```
7795 \newcommand{\led@check@pb}{\xifinlist{\the\absline@num}{\l@prev@pb}{\pagebreak[4]}}
7796 \newcommand{\led@check@nopb}{%
7797   \IfStrEq{\led@pb@setting}{before}{%
7798     \xifinlist{\the\absline@num}{\l@prev@nopb}%
7799     {\numdef{\abs@prevline}{\the\absline@num-1}}%
7800     \xifinlist{\abs@prevline}{\normal@page@break}%
7801     {\nopagebreak[4]\enlargethispage{\baselineskip}}%
7802     {}}%
7803   {}}%
```

```

7804     {}%
7805     {}%
7806     \IfStrEq{\led@pb@setting}{after}{%
7807         \xifinlist{\the\absline@num}{\l@prev@nopb}{%
7808             \xifinlist{\the\absline@num}{\normal@page@break}%
7809                 {\nopagebreak[4]\enlargethispage{\baselineskip}}%
7810             {}%
7811     }%
7812         {}}%
7813     {}%
7814     {}%
7815 }
7816 %

```

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.

```

7817 \newcommand{\check@pb@in@verse}{%
7818     \ifinstanza\iflednopbinverse\ifinserthangingsymbol% Using stanzas and
7819     enabling page breaks in verse control, while on a hanging verse.
7820     \ifnum\page@num=\last@page@num\else%If we have change page
7821     \IfStrEq{\led@pb@setting}{before}{%
7822         \numgdef{\abs@line@verse}{\the\absline@num-1}%
7823         \ledpbnum{\abs@line@verse}%
7824     }{}%
7825     \IfStrEq{\led@pb@setting}{after}{%
7826         \numgdef{\abs@line@verse}{\the\absline@num-1}%
7827         \lednopbnum{\abs@line@verse}%
7828     }{}%
7829     \fi%
7830 }
7831 %

```


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

```

7832 \def\Hy@raisedlink@left#1{%
7833   \ifvmode
7834     #1%
7835   \else
7836     \Hy@SaveSpaceFactor
7837     \llap{\smash{%
7838       \begingroup
7839         \let\HyperRaiseLinkLength\@tempdima
7840         \setlength\HyperRaiseLinkLength\HyperRaiseLinkDefault
7841         \HyperRaiseLinkHook
7842       \expandafter\endgroup
7843       \expandafter\raise\the\HyperRaiseLinkLength\hbox{%
7844         \Hy@RestoreSpaceFactor
7845         #1%
7846         \Hy@SaveSpaceFactor
7847       }%
7848     }}%
7849     \Hy@RestoreSpaceFactor
7850     \penalty\@M\hskip\z@\relax
7851   \fi
7852 }
7853 %

```

XXXV Compatibility with eledmac

Here, we define some commands for the `eledmac-compat` option.

```

7854 \ifeledmaccompat%
7855
7856 \newcommand{\footnormalX}[1]{\arrangementX[#1]{normal}}%
7857 \newcommand{\footparagraphX}[1]{\arrangementX[#1]{paragraph}}%
7858 \newcommand{\foottwocolX}[1]{\arrangementX[#1]{twocol}}%
7859 \newcommand{\footthreecolX}[1]{\XarrangementX[#1]{threecol}}%
7860
7861 \unless\ifnocritical@
7862   \newcommand{\footnormal}[1]{\Xarrangement[#1]{normal}}%
7863   \newcommand{\footparagraph}[1]{\Xarrangement[#1]{paragraph}}%

```

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>.

```

7864 \newcommand{\foottwocol}[1]{\Xarrangement[#1]{twocol}}%
7865 \newcommand{\footthreecol}[1]{\Xarrangement[#1]{threecol}}%
7866 \let\hsizetwocol\Xhsizetwocol
7867 \let\hsizethreecol\Xhsizethreecol
7868 \let\bhookXnote\Xbhooknote
7869 \let\boxsymlinenum\Xboxsymlinenum
7870 \let\symlinenum\Xsymlinenum
7871 \let\beforenumberinfootnote\Xbeforenumber
7872 \let\afternumberinfootnote\Xafternumber
7873 \let\beforeXsymlinenum\XbeforeXsymlinenum
7874 \let\afterXsymlinenum\XafterXsymlinenum
7875 \let\inplaceofnumber\Xinplaceofnumber
7876 \let\Xlemmaseparator\lemmaseparator
7877 \let\afterlemmaseparator\Xafterlemmaseparator
7878 \let\beforelemmaseparator\Xbeforelemmaseparator
7879 \let\inplaceoflemmaseparator\Xinplaceoflemmaseparator
7880 \let\txtbeforeXnotes\Xtxtbeforenotes
7881 \let\afterXrule\Xafterrule
7882 \let\numberonlyfirstinline\Xnumberonlyfirstinline
7883 \let\numberonlyfirstintwoline\Xnumberonlyfirstintwoline
7884 \let\nonumberinfootnote\Xnonumberinfootnote
7885 \let\pstartinfootnote\Xpstart
7886 \let\pstartinfootnoteeverytime\Xpstarteverytime
7887 \let\onlyXpstart\Xonlypstart
7888 \let\Xnonumberinfootnote\Xnonumber
7889 \let\nonbreakableafternumber\Xnonbreakableafternumber
7890 \let\maxhXnotes\Xmaxhnotes
7891 \let\beforeXnotes\Xbeforenotes
7892 \let\boxlinenum\Xboxlinenum
7893 \let\boxlinenumalign\Xboxlinenumalign
7894 \let\boxstartlinenum\Xboxstartlinenum
7895 \let\boxendlinenum\Xboxendlinenum
7896 \let\twoline\Xtwoline
7897 \let\morethantwoline\Xmorethantwoline
7898 \let\twolinebutnotmore\Xtwolinebutnotmore
7899 \let\twolineonlyinsamepage\Xtwolineonlyinsamepage
7900 \fi
7901
7902 \unless\ifnofamiliar@
7903   \let\notesXwidthliketwocolumns\noteswidthliketwocolumnsX
7904 \fi
7905 \newcommandx{\parafootsep}[2][1,usedefault]{%
7906   \Xparafootsep[#1]{#2}%
7907   \parafootsepX[#1]{#2}
7908 }%
7909
7910 \newcommandx{\afternote}[2][1,usedefault]{%
7911   \Xafternote[#1]{#2}%
7912   \afternoteX[#1]{#2}%
7913 }%

```

```

7914
7915 \unless\ifnoend@
7916   \let\XendXtwolines\Xendtwolines
7917   \let\XendXmorethantwolines\Xendmorethantwolines
7918   \let\hookXendnote\Xendhooknote
7919   \let\boxXendlinenum\Xendboxlinenum%
7920   \let\boxXendlinenumalign\Xendboxlinenumalign%
7921   \let\boxXendstartlinenum\Xendboxstartlinenum%
7922   \let\boxXendendlinenum\Xendboxendlinenum%
7923   \let\XendXlemmaseparator\Xendlemmaseparator
7924   \let\XendXbeforelemmaseparator\Xendbeforelemmaseparator
7925   \let\XendXafterlemmaseparator\Xendafterlemmaseparator
7926   \let\XendXinplaceoflemmaseparator\Xendinplaceoflemmaseparator
7927 \fi
7928
7929 \AtBeginDocument{%
7930   \ifdef\lineref{}\let\lineref\edlineref}%
7931 }%
7932
7933
7934 \fi%
7935 %

```

</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. 110 to find out why.) Aliases of the second kind may be nested without any problem.

If you really have to use `\critext` in any of the tabular or array environments, then `\edtext` must not be used in the same environment. If you use `\critext` in one of these environments then you have to issue the declaration `\usingcritext` beforehand. The declaration `\usingedtext` must be issued to revert to the default assumption that `\edtext` will be used.

Appendix A.2 Migration from ledmac to eledmac

In `eledmac`, some changes were made in the code to allow easy customization. This may cause problems for people who have already made their own. The next sections explain how to handle this.

If you have created your own series using `\addfootins` and `\addfootinsX`, you must use instead the `\newseries` command (see 6.5.1 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 (7 p. 30) or `\Xfootnote` options (6.2.2 p. 23). Otherwise please add a new ticket on Github to request a new function for doing this.³⁶

If for some reason you do not want to make the modifications to use the new functions of `eledmac`, you can continue using your own `\XXXXXfmt` command, but you must replace:

```
\renewcommand*{XXXXfmt}[3]
```

with

```
\renewcommandx*{XXXXfmt}[4][4=Z]
```

³⁶<https://github.com/maieul/ledmac/issues>

If you do not make that, you will get a spurious [X], where X is series letter.

If you used a `\protect` command inside a `\footnote` command inside a numbered section, you must change the `\protect` to `\noexpand`. Otherwise the command after the `\protect` will be discarded.

Appendix A.3 Migration to eledmac 1.5.1

The version 1.5.1 corrects a bug in `stanzaindentrepetition` (cf. 9.3 p. 43). 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 5.2.3 p. 17). If so, add a `\relax` between `\pstart`/`\pend` and the first bracket.

The version 1.12.0 also introduce a better way to handle sectional divisions inside numbered text. Please read 16.2 p. 59.

Appendix A.5 Migration to eledmac 17.1

This version changes the default setting of `\Xpstart`. Henceforth, `pstart` numbers will be printed in footnotes within the section of text where you have called `\numberpstarttrue`.

We do not see any reason to print them in the other sections. However, if you want to print the `pstart` numbers in all of the footnotes, whatever the section, without having to use `\numberpstarttrue`, you can use `\Xpstarteverytime`.

Appendix A.6 Migration to eledmac 1.21.0

Appendix A.6.1 `\Xledsetnormalparstuff` and `\ledsetnormalparstuffX`

The `\ledsetnormalparstuff` has been split into two different commands:

- `\Xledsetnormalparstuff` for critical notes;
- `\ledsetnormalparstuffX` for familiar notes.

Both commands can take an optional argument which is the series letter. If you have redefined `\ledsetnormalparstuff` or any of the commands which call them, you must change them accordingly.

Appendix A.6.2 Endnotes

In any case, delete the `.end` file before the next run.

The previous version of Eledmac had a bug: there were two spaces between the starting page number and the starting line number, but only one space between the ending page number and the ending line number.

As a matter of fact, a spurious space was added after the first `\printnpnum`. This spurious space has been deleted. However, if you want to keep the previous spurious space, you may load the package with the `oldprintnpnumspace` option.

If you have redefined `\endprint`, you must:

- Contact us and ask for the feature that required your hack, in order to avoid such a hack in the future.
- Use the new fifth argument.
- Add `\xdef\@currentseries{#4}` at the beginning of your own command.

Appendix A.7 Migration to eledmac 1.22.0

The `\ledinnote` command now takes a first optional argument, which is the label for the hyperreference. If you have redefined it, change your redefinition, and check whether you can avoid this redefinition by only redefining `\ledinnotemark`.

Appendix A.8 Migration to eledmac 1.23.0

You must delete the numbered auxiliary files before compiling with the new version of eledmac.

Appendix A.9 Migration from eledmac to reledmac

There are many changes in reledmac which require the user to make modifications.

Appendix A.9.1 Risk of ‘no room for a new’

The risk to obtain a ‘no room for a new something’ error is greater in reledmac than it is in eledmac. See 19.1.3 p. 61 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
\sidenotesep	\setsidenotsep
\startstanzahook	Optional argument of \stanza
\symplinenum	\Xsymplinenum

Appendix A.9.5 Commands the names of which have been changed

In order to help the migration from eledmac to reledmac, you may load reledmac with eledmac-compat option. However, it is advised not to, and to change the command names themselves instead. In many cases, you use only a few of them, except the \footparagraph command.

<i>Old command</i>	<i>New command</i>
\footparagraph	\Xarrangement
\footnormal	\Xarrangement
\foottwocol	\Xarrangement
\footthreecol	\Xarrangement
\footparagraphX	\arrangementX
\footnormalX	\arrangementX
\foottwocolX	\arrangementX
\footthreecolX	\arrangementX
\afterlemmaseparator	\Xafterlemmaseparator
\afternote	\Xafternote and \afternoteX
\afternumberinfootnote	\Xafternumber
\afterXrule	\Xafterrule
\afterXsymplinenum	\Xaftersymplinenum
\beforelemmaseparator	\Xbeforelemmaseparator
\beforenumberinfootnote	\Xbeforenumber
\beforeXnotes	\Xbeforenotes
\beforeXsymplinenum	\Xbeforesymplinenum

<i>Old command</i>	<i>New command</i>
<code>\bhookXnote</code>	<code>\Xbhookendnote</code>
<code>\bhookXnote</code>	<code>\Xbhooknote</code>
<code>\boxendlinenum</code>	<code>\Xboxendlinenum</code>
<code>\boxlinenum</code>	<code>\Xboxlinenum</code>
<code>\boxlinenumalign</code>	<code>\Xboxlinenumalign</code>
<code>\boxstartlinenum</code>	<code>\Xboxstartlinenum</code>
<code>\boxsymlinenum</code>	<code>\Xboxsymlinenum</code>
<code>\boxXendlinenum</code>	<code>\Xendboxlinenum</code>
<code>\boxXendlinenumalign</code>	<code>\Xendboxlinenumalign</code>
<code>\boxXendstartlinenum</code>	<code>\boxXendstartlinenum</code>
<code>\letboxXendendlinenum</code>	<code>\Xendletboxendlinenum</code>
<code>\hsizetwocol</code>	<code>\Xhsizetwocol</code>
<code>\hsizethreecol</code>	<code>\Xhsizethreecol</code>
<code>\inplaceoflemmaseparator</code>	<code>\Xinplaceoflemmaseparator</code>
<code>\inplaceofnumber</code>	<code>\Xinplaceofnumber</code>
<code>\lemmaseparator</code>	<code>\Xlemmaseparator</code>
<code>\maxhXnotes</code>	<code>\Xmaxhnotes</code>
<code>\morethantwolines</code>	<code>\Xmorethantwolines</code>
<code>\nonumberinfootnote</code>	<code>\Xnonumber</code>
<code>\notesXwidthliketwocolumns</code>	<code>\noteswidthliketwocolumnsX</code>
<code>\noXlemmaseparator</code>	<code>\Xnolemmaseparator</code>
<code>\numberonlyfirstinline</code>	<code>\Xnumberonlyfirstinline</code>
<code>\numberonlyfirstintwolines</code>	<code>\Xnumberonlyfirstintwolines</code>
<code>\nonbreakableafternumber</code>	<code>\Xnonbreakableafternumber</code>
<code>\onlyXpstart</code>	<code>\Xonlypstart</code>
<code>\parafootsep</code>	<code>\Xparafootsep</code> and <code>\parafootsepX</code>
<code>\pstartinfootnote</code>	<code>\Xpstart</code>
<code>\pstartinfootnoteeverytime</code>	<code>\Xpstarteverytime</code>
<code>\symlinenum</code>	<code>\Xsymlinenum</code>
<code>\twolines</code>	<code>\Xtwolines</code>
<code>\twolinesbutnotmore</code>	<code>\Xtwolinesbutnotmore</code>
<code>\twolinesonlyinsamepage</code>	<code>\Xtwolinesonlyinsamepage</code>
<code>\txtbeforeXnotes</code>	<code>\Xtxtbeforenotes</code>
<code>\XendXafterlemmaseparator</code>	<code>\Xendafterlemmaseparator</code>
<code>\XendXbeforelemmaseparator</code>	<code>\Xendbeforelemmaseparator</code>
<code>\XendXinplaceoflemmaseparator</code>	<code>\Xendinplaceoflemmaseparator</code>
<code>\XendXlemmaseparator</code>	<code>\Xendlemmaseparator</code>
<code>\XendXmorethantwolines</code>	<code>\Xendmorethantwolines</code>
<code>\XendXtwolines</code>	<code>\Xendtwolines</code>
<code>\Xnonumberinfootnote</code>	<code>\Xnonumber</code>
<code>\lineref</code>	<code>\edlineref</code>

Appendix A.9.6 Endnotes

With *reledmac*, there is now one auxiliary file for every endnotes set (*.Aend*, *.Bend*, *.Cend* etc.). If you have overridden `\doendnotes` (which you would not have done) you must adapt your code.

Appendix A.9.7 Z Series

The ‘Z’ series of notes has been removed. Only five series are provided now by default: A, B, C, D, E.

Appendix A.9.8 Internal commands

Users who have overridden internal commands, which is wrong, must adapt according to the following. Or better, they should not override any of such commands and use *reledmac* options instead.

- If you have modified `\Xfootfmt`, note that the fourth argument is now mandatory.
- `\unvxh` has been replaced with `\Xunvxh` and `\unvxhX` with two mandatory arguments.

Appendix A.10 Migration to *reledmac* 2.1.0

Reledmac 2.1.0 fix some bugs when using `\Xbhooknote` and `\bhooknoteX` not in order to execute code at the beginning of each notes, but to insert content of at the beginning of each notes.

People who use these commands to do it, which is not the original idea, must change the following:

1. Horizontal space is no longer automatically added after the content of the `\Xbhooknote/\bhooknoteX` argument. You must include it manually. So instead of `\Xbhooknote{content}`, use `\Xbhooknote{content }.`
2. Indent is no longer automatically added before the content of the `\Xbhooknote/\bhooknoteX` argument. If you want to keep it, add `\indent` in the argument of `\Xbhooknote/\bhooknoteX`.

Appendix A.11 Migration to *reledmac* 2.1.3

Reledmac 2.1.3 fix an historical bug, (style in *ledmac* 0.7!) which doubled the space before the rules of paragraphed familiar footnotes. Consequently, if you use paragraphed familiar footnotes, you should maybe adapt it, playing with `\beforenotesX`.

Appendix A.12 Migration to *reledmac* 2.3.0

Before *reledmac* 2.3.0, for typesetting verse, any empty line was considered a paragraph inside verses. Counting empty lines this created breaking verse, hanging verses, and also added spurious vertical spaces. Version 2.3.0 disables paragraph in stanza. If you want vertical space, use optional argument of `\stanza` or `\endverse`.

Appendix A.13 Migration to reledmac 2.4.0

It is not mandatory, but strongly recommended, to change any `\renewcommand{\endashchar}{\langle...\rangle}` to the use of `\Xlinangeseparator` or `/` and `\Xendlinangeseparator` (7.2.3 p. 32).

Appendix A.14 Migration to reledmac 2.5.0

It is strongly recommended to stop redefining `\printnpnum` and to use the hooks documented in 7.3 p. 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 `edstanz.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>	43
<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>\@iiminipage</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>CLASSarticle</code>	61
<code>CLASSbook</code>	61, 301
<code>CLASSmemoir</code>	177, 178, 231–233, 262, 301, 320, 375, 379
<code>CLASSscrbook</code>	379
<code>COMMAND*footnote</code>	63
<code>COMMAND\...@footnotemark...</code>	180

COMMAND\...d@ta	134
COMMAND\<hook	
@<series	221
COMMAND\<hookname	
<pseudoserries	223
COMMAND\<type	
footfmt	168
COMMAND\@@line	161
COMMAND\@MM	148, 376
COMMAND\@Rlineflag	264, 376
COMMAND\@Serefprefix	243
COMMAND\@Serefprefixmore	243
COMMAND\@add@	291
COMMAND\@adv	98
COMMAND\@apprefprefixmore	243
COMMAND\@apprefprefixsingle	243
COMMAND\@bsphack	234
COMMAND\@doclearpage	232, 233, 369, 379
COMMAND\@doreinfeetX	379
COMMAND\@dprintingcolumns	376
COMMAND\@edindex@hyperref	264, 265
COMMAND\@edtext@level	114
COMMAND\@esphack	234
COMMAND\@fnpos	198
COMMAND\@footnotemark	178, 369, 379
COMMAND\@footnotetext	178, 179, 369
COMMAND\@gobble	112
COMMAND\@gobblefive	218, 378
COMMAND\@gobblefour	375
COMMAND\@gobblethree	368
COMMAND\@h	163
COMMAND\@hangingsymbol	268
COMMAND\@iiiminipage	254, 256, 368, 379
COMMAND\@iiiminipage	254
COMMAND\@l	374
COMMAND\@l@tempcnta	136, 138, 144
COMMAND\@l@tempcntb	138
COMMAND\@l@reg	374
COMMAND\@lab	94, 234, 237, 240, 368
COMMAND\@ldunboxmpfoot	256
COMMAND\@led@extranofeet	321
COMMAND\@ledinnote@command	260, 261
COMMAND\@lemma	117, 118
COMMAND\@lock	89, 269
COMMAND\@lopL	369
COMMAND\@lopR	369
COMMAND\@makecol	229, 231, 232, 379
COMMAND\@mpfnpos	198
COMMAND\@nl	94, 95, 97, 99, 107, 237, 368, 369
COMMAND\@nl@reg	95, 318, 369, 374

COMMAND\@opXfeet	369
COMMAND\@opfeetX	379
COMMAND\@opxtrafeeti	379
COMMAND\@page	97, 237
COMMAND\@pend	369
COMMAND\@pendR	369
COMMAND\@ref	95, 103–105, 108, 112
COMMAND\@ref@reg	104, 369
COMMAND\@reinserts	229–232, 379
COMMAND\@secondoftwo	63
COMMAND\@sect	304
COMMAND\@series	219
COMMAND\@set	99
COMMAND\@sidenoteseq	253
COMMAND\@sw	104, 105, 120, 123, 124
COMMAND\@tag	113, 115, 118
COMMAND\@tempcnta	75
COMMAND\@tempcntb	75
COMMAND\@toksa	81
COMMAND\@toksb	81
COMMAND\@xloop	145
COMMAND\@xympar	248, 379
COMMAND\Aendnote	15, 24
COMMAND\Afootfmt	148
COMMAND\Afootgroup	148
COMMAND\Afootnote	8, 15, 22–24, 26, 115, 156, 177, 199, 212, 378
COMMAND\Afootstart	148
COMMAND\AtEveryPend	17, 18, 45, 129, 376, 377, 379
COMMAND\AtEveryPstart	17, 18, 45, 376, 377, 379, 382
COMMAND\AtEveryStanza	45, 383
COMMAND\AtEveryStopStanza	45, 383
COMMAND\Bendnote	15, 22
COMMAND\Bfootnote	8, 15, 177, 199, 212
COMMAND\Centering	39
COMMAND\Cfootnote	177
COMMAND\Columns	75, 152
COMMAND\Dfootnote	177
COMMAND\Efootnote	177
COMMAND\Gls	55
COMMAND\Hy@raisedlink	313
COMMAND\Hy@raisedlink@left	313
COMMAND\LTR	38
COMMAND\NR	304
COMMAND\Pages	75, 229, 231
COMMAND\ProcessOptionsX	67
COMMAND\RL	37
COMMAND\RaggedLeft	39
COMMAND\RaggedRight	39
COMMAND\SEonlypage	242, 381
COMMAND\Seref	49–51, 242, 244, 382

COMMAND\SErefonlypage	49–51, 324, 381
COMMAND\SErefwithpage	49–51, 242, 244, 324, 381
COMMAND\Stanza	374
COMMAND\Waklam	291
COMMAND\X@doreinfeet	230, 379
COMMAND\XR@prefix	247
COMMAND\XR@test	247
COMMAND\XR@test@mac	247
COMMAND\XR@test@mac@test	247
COMMAND\XXXXXXfmt	317
COMMAND\XXXXXfmt	317
COMMAND\Xafterlemmaseparator	36, 321
COMMAND\Xafternote	39, 320, 321
COMMAND\Xafternumber	34, 321
COMMAND\Xafterrule	40, 199, 321, 374, 377
COMMAND\Xaftersymlinenum	34, 321
COMMAND\Xarrangement	31, 62, 149, 150, 221, 321
COMMAND\Xarrangement@footparagraph	154
COMMAND\Xarrangement@normal	150
COMMAND\Xarrangement@paragraph	154
COMMAND\Xbeforeinserting	38
COMMAND\Xbeforelemmaseparator	36, 321
COMMAND\Xbeforenotes	40, 199, 321, 374, 377
COMMAND\Xbeforenumber	32, 34, 321
COMMAND\Xbeforesymlinenum	34, 321
COMMAND\Xbhookendnote	322
COMMAND\Xbhookgroup	40, 381, 382
COMMAND\Xbhooknote	38, 322, 323, 379, 380
COMMAND\Xboxendlinenum	35, 322, 378
COMMAND\Xboxlinenum	34, 35, 322
COMMAND\Xboxlinenumalign	34, 35, 322, 378
COMMAND\Xboxstartlinenum	35, 322, 378
COMMAND\Xboxsymlinenum	34, 322
COMMAND\Xcolalign	39, 377
COMMAND\Xdo@feet	229, 369, 379
COMMAND\Xend	218
COMMAND\XendXafterlemmaseparator	322
COMMAND\XendXbeforelemmaseparator	322
COMMAND\XendXinplaceoflemmaseparator	322
COMMAND\XendXlemmaseparator	322
COMMAND\XendXmorethantwolines	322
COMMAND\XendXtwolines	322
COMMAND\Xendafterenumber	34, 380
COMMAND\Xendafterlemmaseparator	36, 322
COMMAND\Xendafternote	42, 382
COMMAND\Xendafternumber	35
COMMAND\Xendafterpagenumber	35, 50
COMMAND\XendafterpagenumberSErefonlypage	324
COMMAND\Xendaftersymlinenum	34, 35, 380
COMMAND\Xendahookinplaceofnumber	35, 380

COMMAND\Xendahooklinenumber	35, 380
COMMAND\Xendbeforelemmaseparator	36, 322
COMMAND\Xendbeforelinenumber	35
COMMAND\Xendbeforenumber	34, 380
COMMAND\Xendbeforepagenumber	35, 50
COMMAND\XendbeforepagenumberSErefonlypage	50
COMMAND\Xendbeforesymlinenum	34, 35, 380
COMMAND\Xendbhookinplaceofnumber	35, 380
COMMAND\Xendbhooklinenumber	35, 380
COMMAND\Xendbhooknote	38
COMMAND\Xendboxendlinum	35, 378
COMMAND\Xendboxlinenum	35, 322, 376
COMMAND\Xendboxlinenumalign	35, 322, 378
COMMAND\Xendboxstartlinenum	35, 378
COMMAND\Xendboxsymlinenum	34, 380
COMMAND\Xendhangindent	38, 380, 382
COMMAND\Xendinplaceoflemmaseparator	24, 36, 322
COMMAND\Xendinplaceofnumber	34, 379
COMMAND\Xendinsertsep@	205
COMMAND\Xendlemmadisablefontselection	37
COMMAND\Xendlemmafont	37, 381
COMMAND\Xendlemmaseparator	25, 36, 322
COMMAND\Xendletboxendlinum	322
COMMAND\Xendlineflag	50
COMMAND\Xendlineprefixmore	35, 50
COMMAND\Xendlineprefixsingle	35, 51
COMMAND\Xendlinerangeseparator	32, 51, 324, 380
COMMAND\Xendmorethantwolines	24, 33, 51, 322, 377, 378
COMMAND\Xendnonumber	33, 379
COMMAND\Xendnote	202, 217, 218, 377
COMMAND\Xendnotefontsize	37, 321
COMMAND\Xendnotenumfont	35, 36, 321
COMMAND\Xendnumberonlyfirstinline	32, 380
COMMAND\Xendnumberonlyfirstintwolines	32, 380
COMMAND\Xendparagraph	42, 374
COMMAND\Xendsep	42
COMMAND\Xendsublinesep	33, 51
COMMAND\Xendsymlinenum	32, 380
COMMAND\Xendtwolines	24, 32, 33, 51, 322, 377, 378
COMMAND\Xendtwolinesbutnotmore	33, 51, 377, 378
COMMAND\Xendtwolinesonlyinsamepage	33, 51, 377, 378
COMMAND\Xendwrapcontent	38, 383
COMMAND\Xendwraplemma	37, 383
COMMAND\Xfootfmt	323
COMMAND\Xfootgroup	153
COMMAND\Xfootins	152
COMMAND\Xfootnote	47, 53, 113, 317, 371, 375, 377, 381
COMMAND\Xfootstarts	153
COMMAND\Xhangindent	38, 380
COMMAND\Xhsize	324, 381, 382

COMMAND\Xhsizethreecol	39, 41, 322
COMMAND\Xhsizetwocol	39, 41, 222, 322
COMMAND\Xinplaceoflemmaseparator	24, 36, 322
COMMAND\Xinplaceofnumber	34, 322, 377, 378
COMMAND\Xinsertparafootsep	158, 160
COMMAND\Xledsetnormalparstuff	319, 320, 377
COMMAND\Xlemmadisablefontselection	37
COMMAND\Xlemmafont	37, 381
COMMAND\Xlemmaseparator	36, 224, 226, 228, 322
COMMAND\Xlineflag	50
COMMAND\Xlinangeseparator	32, 50, 324, 380
COMMAND\Xmaxhnotes	41, 61, 62, 321, 322, 374, 376
COMMAND\Xmorethantwolines	23, 32, 33, 50, 322, 376
COMMAND\Xnoindent	380
COMMAND\Xnolemmaseparator	36, 228, 322
COMMAND\Xnonbreakableafternumber	34, 322, 372
COMMAND\Xnonumber	33, 322
COMMAND\Xnonumberinfootnote	322
COMMAND\Xnotefontsize	37, 321
COMMAND\Xnotefontsize@<s>	159, 162, 163
COMMAND\Xnotenumfont	36, 321
COMMAND\Xnoteswidthliketwocolumns	41, 375
COMMAND\Xnumberonlyfirstinline	31, 32, 92, 223, 224, 226, 322, 371, 376
COMMAND\Xnumberonlyfirstintwolines	32, 322, 371
COMMAND\Xonlypstart	33, 322, 371, 376
COMMAND\Xparafootsep	39, 92, 321, 322
COMMAND\Xparafootsep@series	158
COMMAND\Xparindent	324
COMMAND\Xparindent	38, 377, 380, 382
COMMAND\Xpstart	33, 319, 322, 371, 376
COMMAND\Xstarteverytime	33, 319, 322, 376
COMMAND\Xragged	40
COMMAND\Xrigidbalance	161, 324, 381
COMMAND\Xstanza	33, 45
COMMAND\Xstanzaseparator	33
COMMAND\Xsublinesep	21, 33, 50, 324
COMMAND\Xsublinesepside	21, 33
COMMAND\Xsymlinenum	32, 39, 321, 322, 378
COMMAND\Xtwolines	23, 32, 33, 50, 173, 174, 223, 322, 376
COMMAND\Xtwolinesappref	223
COMMAND\Xtwolinesbutnotmore	32, 33, 50, 322, 377
COMMAND\Xtwolinesbutnotmoreappref	223
COMMAND\Xtwolinesonlyinsamepage	32, 33, 50, 322, 377
COMMAND\Xtxtbeforenotes	40, 322
COMMAND\Xunvxh	156, 323
COMMAND\Xwidth	41, 324, 382
COMMAND\Xwrapcontent	38, 383
COMMAND\Xwraplemma	37, 38, 383
COMMAND\&	321
COMMAND\absline@num	88, 89, 135

COMMAND\accent	113
COMMAND\actionlines@list	90, 136
COMMAND\actions@list	90
COMMAND\add@inserts	90, 143
COMMAND\add@inserts@next	143, 144
COMMAND\add@penalties	135, 144
COMMAND\addcontentsline	299
COMMAND\addfootins	317, 320
COMMAND\addfootinsX	317, 320
COMMAND\advancelabel@refs	236
COMMAND\advanceline	21, 91, 98, 109, 379
COMMAND\advancepageno	229
COMMAND\affixlin@num	253
COMMAND\affixline@num	137, 140, 141, 369
COMMAND\affixpstart@num	141
COMMAND\afterXrule	321
COMMAND\afterXsymlinenum	321
COMMAND\afterenumber	34
COMMAND\aftergroup	112, 116
COMMAND\afterlemmaseparator	321
COMMAND\afternote	321
COMMAND\afternoteX	39, 320, 321
COMMAND\afternumberinfootnote	321
COMMAND\afterruleX	40, 374, 377
COMMAND\applabel	49, 237, 238, 377
COMMAND\appref	47, 49–51, 242, 244, 381, 382
COMMAND\apprefprefixmore	49, 321
COMMAND\apprefprefixsingle	49, 321
COMMAND\apprefwithpage	49–51, 242, 244, 378, 381
COMMAND\arrangementX	31, 62, 181, 221, 321
COMMAND\arrangementX@normal	186
COMMAND\article	14
COMMAND\at@every@pend	129
COMMAND\autopar	17, 126, 129, 130, 196, 370, 372, 373, 377
COMMAND\ballast	62
COMMAND\ballast@count	135, 144
COMMAND\baselineskip	31, 155, 159
COMMAND\beforeXnotes	321
COMMAND\beforeXsymlinenum	321
COMMAND\beforeelectedchapter	9, 59, 300
COMMAND\beforeinsertingX	38
COMMAND\beforelemmaseparator	321
COMMAND\beforenotesX	40, 323, 373, 374, 377
COMMAND\beforenumberinfootnote	321
COMMAND\begin	276
COMMAND\beginnumbering	16, 18, 76, 77, 79, 88, 92, 93, 106, 129, 202, 371, 374, 378, 379
COMMAND\bf	371
COMMAND\bfseries	36, 37, 371
COMMAND\bhookXnote	322
COMMAND\bhookgroupX	40, 381

COMMAND\hooknoteX	38, 323, 379, 380
COMMAND\body	269
COMMAND\bodyfootmarkA	29
COMMAND\book	14
COMMAND\boxXendlinenum	322
COMMAND\boxXendlinenumalign	322
COMMAND\boxXendstartlinenum	322
COMMAND\boxendlinenum	322
COMMAND\boxlinefootnote	170
COMMAND\boxlinenum	322
COMMAND\boxlinenumalign	322
COMMAND\boxstartlinenum	322
COMMAND\boxsymlinenum	322
COMMAND\break	31, 156
COMMAND\brokenpenalty	144
COMMAND\centering	39
COMMAND\ch@ck@l@ck	369
COMMAND\ch@cksub@l@ck	140, 369
COMMAND\chapter	58, 301, 374, 377, 379
COMMAND\chaptermark	299
COMMAND\check@pb@in@verse	312
COMMAND\colalignX	39, 377
COMMAND\collect@body	277
COMMAND\color	383
COMMAND\colorbox	63
COMMAND\columns	41
COMMAND\columnwidth	155, 375
COMMAND\command names	222, 223
COMMAND\copyright	113
COMMAND\correct@Xfootins@box	376
COMMAND\correct@footinsX@box	376
COMMAND\count	162
COMMAND\critex	370
COMMAND\critext	119, 316, 317, 320
COMMAND\csname	67, 122
COMMAND\ctab	292, 297
COMMAND\ctabtext	297
COMMAND\dcoll	286
COMMAND\def	65
COMMAND\detokenize	122
COMMAND\dimen	162
COMMAND\dimexpr	41
COMMAND\discretionary	156
COMMAND\displaywidowpenalty	144
COMMAND\do@actions	135, 137, 369
COMMAND\do@actions@fixedcode	369
COMMAND\do@actions@next	135, 136
COMMAND\do@ballast	135, 144
COMMAND\do@feetX	379
COMMAND\do@insidelinehook	372

COMMAND\do@line	90, 111, 128, 131, 133, 134, 143, 144, 269, 369, 370, 372, 374
COMMAND\do@linehook	369
COMMAND\do@lockoff	91
COMMAND\do@lockon	91
COMMAND\dodoreinxtrafeet	368
COMMAND\doendnotes	24, 205, 323, 378
COMMAND\doendnotesbysection	24, 205, 218, 378
COMMAND\doinsidelinehook	22, 375
COMMAND\dolinehook	22, 375
COMMAND\doreinxtrafeeti	379
COMMAND\doreinxtrafeetii	379
COMMAND\doxtrafeet	229, 368
COMMAND\doxtrafeeti	379
COMMAND\doxtrafeetii	379
COMMAND\dummy@ref	112
COMMAND\edaftertab	57, 292, 293
COMMAND\edatleft	57, 290
COMMAND\edatright	57, 58, 290
COMMAND\edbeforetab	57, 292
COMMAND\edfilldimen	291
COMMAND\edfont@info	118
COMMAND\edgls	55, 259
COMMAND\edgls...	382
COMMAND\edindex	53–55, 259, 260, 263, 265, 281, 372, 375, 376, 379, 380
COMMAND\edindexlab	54
COMMAND\edlabel	47–49, 51, 113, 234, 236, 237, 239, 240, 246, 259, 281, 313, 368, 371–373, 376, 381
COMMAND\edlabelE	49, 238
COMMAND\edlabelS	49, 238
COMMAND\edlabelSE	49
COMMAND\edlineref	47, 234, 322, 376, 378, 381
COMMAND\edmakelabel	48, 246
COMMAND\edpageref	47, 234, 239, 246
COMMAND\edrowfill	291
COMMAND\edtabcolsep	285
COMMAND\edtext	6, 23, 25–28, 42, 47–49, 53, 55, 63, 89, 103, 105, 108, 110–119, 121, 123–125, 237, 238, 241, 280, 282, 301, 313, 316, 317, 320, 368, 370, 372, 374–378
COMMAND\edtext@level	378
COMMAND\edvertdots	58, 290
COMMAND\edvertline	58, 290
COMMAND\elechapter	59
COMMAND\eled@sectioning@out	306
COMMAND\eledchapter	59, 320, 375, 379
COMMAND\eledchapter*	59
COMMAND\eledmac@error	368
COMMAND\eledsection	6, 15, 59, 112, 133, 300, 320, 377
COMMAND\eledsection*	59
COMMAND\eledsubsection	59, 320
COMMAND\eledsubsection*	59
COMMAND\eledsubsubsection	59, 320
COMMAND\eledsubsubsection*	59

COMMAND\eledxxx	9, 59, 300, 306, 374
COMMAND\eledxxxx	299
COMMAND\else	258, 300
COMMAND\empty	75, 139, 234
COMMAND\end	276
COMMAND\end@lemmas	112
COMMAND\endashchar	42, 167
COMMAND\endgraf	128, 158, 196
COMMAND\endlock	21, 91, 110, 273
COMMAND\endminipage	254, 256, 368, 379
COMMAND\endnotes	377, 381
COMMAND\endnumbering	16, 18, 76, 78, 79, 369, 378
COMMAND\endprint	202, 204, 218, 319
COMMAND\endstanzaextra	321
COMMAND\endsub	20, 90, 108
COMMAND\endverse	323
COMMAND\everypar	130
COMMAND\extensionchars	61, 76
COMMAND\externaldocument	51, 247
COMMAND\fix@l@cks	369
COMMAND\falseverse	320, 372, 374
COMMAND\fi	300
COMMAND\firstlinenum	19, 138, 370
COMMAND\firstsublinenum	19, 370
COMMAND\fix@page	95, 97, 369
COMMAND\flag@end	108, 117, 374
COMMAND\flag@start	108, 117, 374, 375
COMMAND\flagstanza	46
COMMAND\floatingpenalty	148, 376
COMMAND\flush@notes	145
COMMAND\fnpos	198, 373
COMMAND\footfmt	148, 150
COMMAND\footfmt...	182
COMMAND\footfootmarkA	29
COMMAND\footfudgefactor	156
COMMAND\footfudgefiddle	62, 154, 155, 368
COMMAND\footgroup	148
COMMAND\footins	152
COMMAND\footnormal	222, 321, 368
COMMAND\footnormalX	321
COMMAND\footnote	29, 62, 177–179, 318, 369
COMMAND\footnote@lang	167
COMMAND\footnoteA	15, 29
COMMAND\footnoteB	15
COMMAND\footnoteC	22
COMMAND\footnoteE	29
COMMAND\footnoteX	8, 215
COMMAND\footnoteX@reading	216
COMMAND\footnoteXmk	228
COMMAND\footnotelang@lua	147

COMMAND\footnotelang@poly	147
COMMAND\footnoteoption@	146, 380
COMMAND\footnoterule	162
COMMAND\footnotesize	37
COMMAND\footparagraph	155, 222, 321, 374
COMMAND\footparagraphX	191, 321, 374
COMMAND\footsplitskips	369, 376
COMMAND\footstart	148, 152, 162
COMMAND\footstrut	158
COMMAND\footthreecol	321
COMMAND\footthreecolX	321, 378
COMMAND\foottwocol	321
COMMAND\foottwocolX	321, 378
COMMAND\foreignlanguage	38
COMMAND\fullstop	42, 324
COMMAND\get@edindex@hyperref	264
COMMAND\get@edindex@ledinnote@command	260
COMMAND\get@fnmark	179
COMMAND\get@index@command	373
COMMAND\get@linelistfile	369
COMMAND\get@thisfootnote	185
COMMAND\getline@num	134, 136
COMMAND\gl@p	81
COMMAND\global	94
COMMAND\globaldefs	95
COMMAND\gls	55, 267
COMMAND\hangindentX	38, 377, 380
COMMAND\hangingsymbol	321, 370
COMMAND\hbox	156
COMMAND\hfill	373
COMMAND\hidenumbering	22, 103, 377
COMMAND\hline	55
COMMAND\hrulefill	291
COMMAND\hsize	31, 152, 155, 157, 162, 165, 197, 369, 375
COMMAND\hsizeX	324, 381, 382
COMMAND\hsizethreecol	322
COMMAND\hsizethreecolX	39, 41
COMMAND\hsizetwocol	322
COMMAND\hsizetwocolX	39, 41
COMMAND\hyperlinkR	263
COMMAND\hyperlinkformat	263
COMMAND\hyperlinkformatR	264
COMMAND\if@RTL	69
COMMAND\if@edtext@	375, 378
COMMAND\if@eled@sectioning	300
COMMAND\if@nobreak	129
COMMAND\if@noneed@Footnote	108
COMMAND\ifXnote@	76
COMMAND\ifbypage@	81
COMMAND\ifbypage@R	82

COMMAND\ifbypstart@	81
COMMAND\ifbypstart@R	82
COMMAND\iffirst@linenum@out@	106
COMMAND\ifindtl@innote	76
COMMAND\ifindtl@notenumber	76
COMMAND\ifinserthangingsymbol	269
COMMAND\ifinstanza	269
COMMAND\ifistwofollowinglines	174
COMMAND\ifl@d@Xmorethantwolines	172, 376
COMMAND\ifl@d@Xtwolines	172
COMMAND\ifl@d@dash	172
COMMAND\ifl@d@elin	171
COMMAND\ifl@d@esl	171
COMMAND\ifl@d@pnum	171
COMMAND\ifl@d@ssub	171
COMMAND\ifl@dend@X	217
COMMAND\ifl@dmemoir	368
COMMAND\ifl@dpaging	375
COMMAND\ifl@dpairing	75, 370
COMMAND\ifl@dprintingpages	376
COMMAND\ifl@dskipnumber	138
COMMAND\ifl@dstartendok	291
COMMAND\ifl@imakeidx	69
COMMAND\ifledRcol	75, 370
COMMAND\ifledRcol@	75, 374
COMMAND\iflemmacommand@	375
COMMAND\ifnoend@	206
COMMAND\ifnoledgroup@	258
COMMAND\ifnoteschanged@	91
COMMAND\ifnumberedpar@	126
COMMAND\ifnumbering	76, 79
COMMAND\ifnumberingR	76, 370
COMMAND\ifnumberline	117, 138
COMMAND\ifpst@rted	370
COMMAND\ifpst@rtedL	77
COMMAND\ifseriesbefore	220
COMMAND\ifsublines@	88, 100
COMMAND\iftrue	378
COMMAND\ifvmode	236
COMMAND\ifxxx	300
COMMAND\ignorespaces	115
COMMAND\imki@wrindexentry	69
COMMAND\immediate	106, 201
COMMAND\indent	17, 130, 323
COMMAND\index	266
COMMAND\indtl@wrindexentry	69
COMMAND\initnumbering@quote	297, 379
COMMAND\initnumbering@reg	369
COMMAND\initnumbering@sectcmd	379
COMMAND\inplaceoflemmaseparator	322

COMMAND\inplaceofnumber	322
COMMAND\insert	143, 148, 150, 182
COMMAND\insert@count	103, 108, 115
COMMAND\insert@countR	115
COMMAND\inserthangingsymbol	373
COMMAND\insertlines@list	90, 103, 104
COMMAND\insertparafootsepX	195
COMMAND\inserts@list	111, 126, 143, 156
COMMAND\interAfootnotelinepenalty	370
COMMAND\interfootnotelinepenalty	370
COMMAND\interlinepenalty	148
COMMAND\interparanoteglue	320
COMMAND\justifying	39
COMMAND\l@advance@parledegroupp@beforenormalnotes	379
COMMAND\l@d@@wrindexhyp	375
COMMAND\l@d@add	120
COMMAND\l@d@end	202, 217
COMMAND\l@d@nums	115, 117, 119, 120, 171
COMMAND\l@d@section	202
COMMAND\l@d@set	99, 109
COMMAND\l@dampcount	282
COMMAND\l@dbfnote	179, 369
COMMAND\l@dcheckstartend	291
COMMAND\l@dchset@num	99
COMMAND\l@dcolcount	282, 283
COMMAND\l@dcollect@@body	276
COMMAND\l@dcollect@body	276
COMMAND\l@dcsnote	374
COMMAND\l@dcsnotetext	134, 251
COMMAND\l@dcsnotetext@l	134, 251
COMMAND\l@dcsnotetext@r	134, 251
COMMAND\l@ddodoreinextrafeet	230, 368
COMMAND\l@ddoxtrafeet	230, 368
COMMAND\l@emptyd@ta	370
COMMAND\l@dend@close	201
COMMAND\l@dend@open	201
COMMAND\l@dend@stuff	202
COMMAND\l@denvbody	276
COMMAND\l@dfeetbeginmini	368
COMMAND\l@dfeetendmini	368
COMMAND\l@dgetline@margin	370
COMMAND\l@dgetlock@disp	370
COMMAND\l@dgetref@num	240, 241
COMMAND\l@dgetsidenote@margin	248, 370
COMMAND\l@dgobbeloptarg	375
COMMAND\l@dgobblearg	375
COMMAND\l@dgobbleoptarg	281
COMMAND\l@dlabel@parse	240, 241
COMMAND\l@dld@ta	137, 140
COMMAND\l@dldp@rbox	252

COMMAND\l@dlsn@te	370
COMMAND\l@dlsnote	374
COMMAND\l@dmake@labels	236, 247
COMMAND\l@dmake@labelsR	247
COMMAND\l@dnumpsstartsL	77, 370
COMMAND\l@dp@rsefootspec	172
COMMAND\l@dpush@begins	276
COMMAND\l@drd@ta	137, 140
COMMAND\l@dref@undefined	240
COMMAND\l@drsn@te	370
COMMAND\l@drsnote	374
COMMAND\l@dtabaddcols	291
COMMAND\l@dtabnoexpands	368
COMMAND\l@dumboxmpfoot	379
COMMAND\l@dunboxmpfoot	370
COMMAND\l@dzeropenalties	370, 375
COMMAND\l@pb	311
COMMAND\l@prev@nopb	311
COMMAND\l@prev@pb	311
COMMAND\l@reg	318
COMMAND\label	18, 49, 51, 54, 234, 235, 241
COMMAND\label@refs	234
COMMAND\labelstarttrue	18, 371
COMMAND\labelref@list	234, 237
COMMAND\language	156
COMMAND\last@page@num	369
COMMAND\lastbox	130
COMMAND\lastskip	108
COMMAND\leavevmode	17, 130
COMMAND\led@check@nopb	311
COMMAND\led@check@pb	311
COMMAND\led@nopb	311, 312
COMMAND\led@nopbnum	311
COMMAND\led@pb	311, 312
COMMAND\led@pb@macro	311
COMMAND\led@pbnum	311
COMMAND\led@reinit@index@fornote	266
COMMAND\led@set@index@fornote	266
COMMAND\ledRflag	263
COMMAND\ledchapter	320, 372
COMMAND\ledfootinsdim	321
COMMAND\ledinnernote	51, 249, 374
COMMAND\ledinnote	262, 319, 378
COMMAND\ledinnotemark	53, 319, 377
COMMAND\ledleftnote	52, 249
COMMAND\ledlinenum	87, 370
COMMAND\ledllfill	134
COMMAND\ledlsnotefontsetup	383
COMMAND\ledlsnotesep	52
COMMAND\ledlsnotewidth	52

COMMAND\lednopb	60, 311
COMMAND\lednopbinverse	312
COMMAND\lednopbinversetrue	45, 60
COMMAND\lednopbnum	311
COMMAND\ledouternote	51, 249, 374
COMMAND\ledpb	60, 311, 320
COMMAND\ledpbnum	311
COMMAND\ledpbsetting	60, 311, 380
COMMAND\ledrightnote	52, 249
COMMAND\ledrsnotefontsetup	383
COMMAND\ledrsnotesep	52
COMMAND\ledrsnotewidth	52
COMMAND\ledsection	320
COMMAND\ledsectnomark	299
COMMAND\ledsectnotoc	299
COMMAND\ledsetnormalparstuff	319, 320, 377
COMMAND\ledsetnormalparstuff@common	196
COMMAND\ledsetnormalparstuffX	319, 320, 377
COMMAND\ledsidenote	51, 52, 249, 251
COMMAND\ledsubsection	320
COMMAND\ledsubsubsection	320
COMMAND\ledxxx	374
COMMAND\left	57
COMMAND\leftctab	292
COMMAND\leftheadline	87
COMMAND\leftlinenum	20, 87, 368, 370
COMMAND\leftltab	292
COMMAND\leftnoteupfalse	52
COMMAND\leftpstartnum	142
COMMAND\leftrtab	292
COMMAND\leftsidenote	251
COMMAND\leftskip	152, 155, 156
COMMAND\lemma	3, 23, 25–28, 111, 114–116, 118, 119, 121, 316, 370, 371, 378, 379, 381
COMMAND\lemmaseparator	322
COMMAND\let	26, 42, 273, 368
COMMAND\letboxXendendlinenum	322
COMMAND\line	161, 163
COMMAND\line@list	89, 104, 117, 118
COMMAND\line@list@stuff	77, 92, 93, 106, 368, 370
COMMAND\line@list@version	95
COMMAND\line@margin	83, 139, 248
COMMAND\line@num	88–90, 138, 368
COMMAND\line@set	119, 120
COMMAND\lineation	20, 82
COMMAND\linebreak	31
COMMAND\linenum	23, 25, 26, 48, 49, 111, 119, 239, 241, 247, 316, 382
COMMAND\linenum@out	105, 106, 234, 237
COMMAND\linenumberlist	19, 75, 139, 368
COMMAND\linenumberstyle	21, 86, 368
COMMAND\linenumincrement	19, 370

COMMAND\linenummargin	20, 83, 248
COMMAND\linenumr@p	86, 368, 370
COMMAND\linenumrep	86, 370
COMMAND\linenumsep	20, 52, 87, 249
COMMAND\linerangesep@	228
COMMAND\lineref	234, 239, 246, 322, 376
COMMAND\list@clear	80
COMMAND\list@clearing@reg	370
COMMAND\list@create	80
COMMAND\lock@disp	85
COMMAND\lock@off	102
COMMAND\lock@on	101
COMMAND\lockdisp	21, 85
COMMAND\loop	145, 269
COMMAND\ltab	292, 293, 297
COMMAND\ltabtext	297
COMMAND\m@mmf@prepare	178
COMMAND\makeatletter	133
COMMAND\makehboxofhboxes	157, 158
COMMAND\makeindex	53, 263
COMMAND\makelabel	246
COMMAND\managestanza@modulo	270
COMMAND\marginpar	51, 62, 247, 248, 369
COMMAND\marginparwidth	52, 249
COMMAND\markboth	133
COMMAND\mathchardef	270
COMMAND\maxhXnotes	322
COMMAND\maxhnotesX	41, 62, 321, 373, 374, 376–378
COMMAND\maxlinesinpar@list	92
COMMAND\measurebody	294
COMMAND\measuretbody	294
COMMAND\memorybreak	19
COMMAND\morenoexpands	63, 111, 113
COMMAND\morethantwolines	322
COMMAND\mpfnpos	198, 373
COMMAND\mpnormalfootgroup	369
COMMAND\mpnormalvfootnote	369
COMMAND\multfootsep	29, 177
COMMAND\multiplefootnotemarker	177
COMMAND\musixtex	374
COMMAND\n@num	370, 377
COMMAND\n@num@ref	377
COMMAND\new@line	107, 369
COMMAND\newcommand	26, 65, 177, 236
COMMAND\newcommandx	26
COMMAND\newhookarg@specific	228
COMMAND\newhookcommand@series	222, 224, 377
COMMAND\newhookcommand@series@reload	224
COMMAND\newhookcommand@toggle@reload	223, 375
COMMAND\newhooktoggle@series	223, 377

COMMAND\newhooktoggle@specific	227
COMMAND\newif	377
COMMAND\newline	31
COMMAND\newlinechar	217
COMMAND\newseries	29, 317, 320, 321
COMMAND\newseries@	209, 210, 221
COMMAND\newverse	45, 46, 320, 374
COMMAND\next	269
COMMAND\next@action	93
COMMAND\next@actionline	93
COMMAND\next@insert	143
COMMAND\nl@regR	95
COMMAND\no@expands	63, 118, 368
COMMAND\noXlemmaseparator	322
COMMAND\nobreak	171
COMMAND\nocritical	210
COMMAND\noeledsec	59, 320
COMMAND\noendnotes	320
COMMAND\noexpand	318
COMMAND\nofamiliar	225
COMMAND\noindent	17, 18, 130, 380
COMMAND\noindentX	380
COMMAND\nomk@	228
COMMAND\nonbreakableafternumber	322
COMMAND\nonnumberinfootnote	322
COMMAND\normal@footnotemarkX	181
COMMAND\normal@page@break	310
COMMAND\normal@pars	196
COMMAND\normalbfnoteX	370
COMMAND\normalbodyfootmarkX	181
COMMAND\normalfootfmt	42, 151, 158, 167, 202, 382
COMMAND\normalfootfmtX	182, 183
COMMAND\normalfootfootmarkX	183
COMMAND\normalfootgroup	153
COMMAND\normalfootgroupX	183
COMMAND\normalfootnoterule	149
COMMAND\normalfootstart	152, 155
COMMAND\normalfootstartX	183
COMMAND\normalvfootnote	150
COMMAND\normalvfootnoteX	182
COMMAND\notbool	300
COMMAND\notfontsetup	321
COMMAND\notfontsizeX	37, 321
COMMAND\notenumfont	321
COMMAND\notenumfontX	37, 321
COMMAND\notesXwidthliketwocolumns	322
COMMAND\noteswidthliketwocolumnsX	41, 322, 375, 377
COMMAND\num@lines	126, 144
COMMAND\numberlinefalse	19
COMMAND\numberlinetrue	19

COMMAND\numberonlyfirstinline	220, 322
COMMAND\numberonlyfirstintwolines	322
COMMAND\numberpstartfalse	18
COMMAND\numberpstarttrue	18, 33, 319, 371, 379
COMMAND\numberstanza	33
COMMAND\numberstanzafalse	45
COMMAND\numberstanzatrue	45
COMMAND\numlabfont	20, 42, 87
COMMAND\one@line	126
COMMAND\onehalfspacing	380
COMMAND\onlyXpstart	322
COMMAND\page@action	90, 99
COMMAND\page@start	90, 370
COMMAND\pagecontents	90
COMMAND\pagelinesep	53
COMMAND\pageno	229
COMMAND\pageparbreak	320
COMMAND\pageref	49, 239
COMMAND\par	25, 31, 129, 130, 196
COMMAND\par@line	126, 144
COMMAND\para@footgroup	155
COMMAND\para@footgroupX	194
COMMAND\para@footsetup	155, 368
COMMAND\para@footsetupX	192, 368, 375
COMMAND\para@vfootnote	158
COMMAND\para@vfootnoteX	193
COMMAND\parafootfmt	157, 158, 382
COMMAND\parafootfmtX	194
COMMAND\parafootftm	160
COMMAND\parafootftmX	195
COMMAND\parafootftmsep	321
COMMAND\parafootsep	322, 373, 378
COMMAND\parafootsepX	39, 92, 321, 322
COMMAND\parafootstart	155
COMMAND\parafootstartX	192
COMMAND\paravfootnote	156
COMMAND\parfillskip	158
COMMAND\parindent	380
COMMAND\parindentX	38, 324, 380, 382
COMMAND\parshape	62
COMMAND\parskip	130
COMMAND\pausenumbering	18, 79, 93, 94, 131, 373, 375, 382
COMMAND\penalty	158
COMMAND\pend	2, 6, 16–21, 59, 60, 109, 111, 114, 120, 126–131, 141–143, 318, 373, 374
COMMAND\preXnotes	40, 200, 377
COMMAND\preXnotes@	152, 200, 371
COMMAND\prenotesX	40, 200, 377
COMMAND\prepare@preXnotes	199
COMMAND\prev@nopb	311
COMMAND\prev@pb	311

COMMAND\prevlineX	92
COMMAND\prevpageX@num	92
COMMAND\print@Xfootnoterule	377
COMMAND\print@Xnotes	229, 231
COMMAND\print@Xnotes@forpages	376
COMMAND\print@eledsection	133
COMMAND\print@footnoteXrule	377
COMMAND\print@leftmargin@eledsection	300
COMMAND\print@lemma	168
COMMAND\print@line	131
COMMAND\print@notesX@forpages	376
COMMAND\print@rightmargin@eledsection	300
COMMAND\printendlines	206, 244, 324, 368, 370
COMMAND\printlinefootnote	168, 170, 376
COMMAND\printlinefootnotearea	170, 171, 376
COMMAND\printlinefootnotenumbers	168
COMMAND\printlines	151, 167, 171, 172, 206, 244, 324, 368, 370, 376, 381
COMMAND\printnpnum	319, 324
COMMAND\printpstart	167
COMMAND\protect	113, 318
COMMAND\providecommand	177, 368
COMMAND\pstart	2, 6, 16–21, 58–60, 99, 109, 114, 120, 126, 128–130, 133, 143, 318, 370–372, 374, 375, 377–379, 382
COMMAND\pstartinfootnote	322
COMMAND\pstartinfootnoteeverytime	322
COMMAND\pstartnum	142
COMMAND\pstartref	47, 234, 240, 373
COMMAND\pstarts	371
COMMAND\raggedX	40
COMMAND\raggedleft	39
COMMAND\raggedright	39
COMMAND\raw@text	126
COMMAND\rbracket	36, 42
COMMAND\read@linelist	92, 94
COMMAND\ref	49, 51, 54
COMMAND\reformatted@	244
COMMAND\reformattedwithpage	244
COMMAND\relax	18, 99, 135, 143, 273, 281, 318
COMMAND\renewcommand	62, 321, 324
COMMAND\resetprevline@	92
COMMAND\resetprevpage@	92
COMMAND\resumenumbering	18, 76, 79, 93, 94, 131, 370, 374, 375, 382
COMMAND\right	57
COMMAND\rightctab	292
COMMAND\rightlinenum	20, 87, 368, 370
COMMAND\rightltab	293
COMMAND\rightnoteupfalse	52
COMMAND\righttab	293
COMMAND\rightsidenote	251
COMMAND\righskip	152, 155, 156, 158

COMMAND\rightstartnum	142
COMMAND\rigidbalance	160–163, 324, 381
COMMAND\rigidbalanceX	161, 324, 381
COMMAND\robustify	32
COMMAND\roman	286, 381
COMMAND\rtab	292, 293, 297
COMMAND\rtabtext	294, 297
COMMAND\sameword	26–28, 120–122, 124, 376, 378, 380
COMMAND\sameword@inedtext	121
COMMAND\saweword	121
COMMAND\scriptsize	87
COMMAND\section	58, 370
COMMAND\section@num	76
COMMAND\sectionmark	299
COMMAND\select@lemmafont	42, 146
COMMAND\series	209, 210
COMMAND\series@	210
COMMAND\seriesatbegin	30, 219, 377
COMMAND\seriesatend	30, 220, 377
COMMAND\set@line	117
COMMAND\set@line@action	91, 100
COMMAND\setSErefonlypageprefixmore	50, 244, 382
COMMAND\setSErefonlypageprefixsingle	50, 244, 382
COMMAND\setSErefprefixmore	50
COMMAND\setSErefprefixsingle	50
COMMAND\setapprefprefixmore	50, 321
COMMAND\setapprefprefixsingle	50, 321, 381
COMMAND\setcommand@series	222
COMMAND\sethangingsymbol	44, 268, 321, 380
COMMAND\sethanginsymbol	43
COMMAND\setistwofollowinglines	174
COMMAND\setl@dlprbox	252
COMMAND\setline	21, 91, 95, 99, 109, 113, 128, 379
COMMAND\setlinenum	21, 95, 99, 109, 368
COMMAND\setprintendlines	206, 208, 370
COMMAND\setprintlines	172, 174, 206, 370
COMMAND\setsidenotesep	52
COMMAND\setsidenotsep	321
COMMAND\setstanzaindent	271
COMMAND\setstanzaindents	44, 270, 318
COMMAND\setstanzapenalties	270
COMMAND\setstanzavalues	270
COMMAND\settoggle@series	221, 371, 375
COMMAND\showlemma	112, 368, 369
COMMAND\showwordrank	28, 122
COMMAND\sidenote@margin	369
COMMAND\sidenotemargin	51, 369, 374
COMMAND\sidenotesep	321
COMMAND\sidepstartnumtrue	18
COMMAND\skip	152

COMMAND\skipnumbering	22, 102, 110, 370, 377, 378
COMMAND\skipnumbering@reg	377
COMMAND\small	37
COMMAND\special	12
COMMAND\splitmaxdepth	148, 162
COMMAND\splitoff	161
COMMAND\splittopskip	148, 162, 163
COMMAND\stanza	21, 22, 45, 46, 273, 321, 323, 380
COMMAND\stanza@hang	273
COMMAND\stanza@line	273
COMMAND\stanzaindent	44, 271, 376
COMMAND\stanzaindent*	44
COMMAND\stanzaindentbase	270
COMMAND\stanzanumwrapper	46
COMMAND\startlock	21, 91, 110, 273
COMMAND\startstanzahook	321
COMMAND\startsub	20, 90, 108
COMMAND\strip@pt	155
COMMAND\strutbox	162
COMMAND\sub@action	90, 100
COMMAND\sub@lock	89
COMMAND\sub@off	97, 237
COMMAND\sub@on	97, 237
COMMAND\subline@num	88–90
COMMAND\sublinenum@rep	368
COMMAND\sublinenumberstyle	22, 86, 368
COMMAND\sublinenumincrement	19
COMMAND\sublinenumr@p	86, 368, 370
COMMAND\sublinenumrep	86, 370
COMMAND\sublineref	47, 234, 239
COMMAND\subsectionmark	299
COMMAND\sw@inthisedtext	114
COMMAND\sw@list@inedtext	118, 125
COMMAND\symlinenum	322
COMMAND\symplinum	321
COMMAND\sza@penalty	273
COMMAND>tag	376
COMMAND\text	316
COMMAND\text<language>	38
COMMAND\textcolor	63
COMMAND\textheight	62
COMMAND\the	368
COMMAND\thefootnoteA	29
COMMAND\thefootnoteX	372
COMMAND\thelabidx	265
COMMAND\thepage	95
COMMAND\thepstart	18
COMMAND\thepstartL	371
COMMAND\thepstartR	371
COMMAND\thestanza	45

COMMAND\this@line@list@version	106
COMMAND\thisfootnote	185
COMMAND\threecolfootfmt	163, 382
COMMAND\threecolfootfmtX	190
COMMAND\threecolfootgroup	162
COMMAND\threecolfootgroupX	191
COMMAND\threecolfootsetup	162
COMMAND\threecolfootsetupX	190
COMMAND\threecolvfootnote	162
COMMAND\threecolvfootnoteX	190
COMMAND\twocolfootfmt	382
COMMAND\twocolfootfmtX	188
COMMAND\twocolfootgroupX	188
COMMAND\twocolfootsetupX	187
COMMAND\twocolvfootnoteX	188
COMMAND\twolines	221, 322
COMMAND\twolines@A	221
COMMAND\twolines@B	221
COMMAND\twolines@C	221
COMMAND\twolinesbutnotmore	322
COMMAND\twolinesonlyinsamepage	322
COMMAND\txtbeforeXnotes	322
COMMAND\unhbox	156
COMMAND\unpenalty	157, 158
COMMAND\unskip	158
COMMAND\unvxh	158, 323
COMMAND\unvxhX	323
COMMAND\upbracefill	291
COMMAND\usingcritext	317, 320
COMMAND\usingedtext	317, 320
COMMAND\VAfootnote	148
COMMAND\variant	26
COMMAND\vbox	128, 130, 156, 160, 161, 198
COMMAND\vfootnote	148, 152, 156, 162
COMMAND\vl@dbfnote	179, 369
COMMAND\vnumfootnoteX	370
COMMAND\vsizer	41, 62
COMMAND\vsplit	144
COMMAND\waklam	291
COMMAND\waklamec	291
COMMAND\wapunktel	291
COMMAND\wastricht	291
COMMAND\widthX	41, 324, 382
COMMAND\wrap@edcrossref	239, 375
COMMAND\wrapcontentX	38, 383
COMMAND\wrapped@bodyfootmarkX	196
COMMAND\wrapped@footfootmarkX	195
COMMAND\x...	48
COMMAND\xdef	81, 273
COMMAND\xflagref	48, 240, 324, 381

COMMAND\xleft@appenditem	81, 112
COMMAND\xlineref	48, 324, 381
COMMAND\xpageref	48
COMMAND\xpstartref	48, 373
COMMAND\xr	51
COMMAND\xright@appenditem	81
COMMAND\xsublineref	48
COMMAND\xxref	48, 241, 247, 373, 376
COMMAND\zz@@@	368
ENVIRONMENTastanza	381
ENVIRONMENTedarrayc	297
ENVIRONMENTedarrayl	297
ENVIRONMENTedarrayr	297
ENVIRONMENTedtabularc	297
ENVIRONMENTedtabularl	297
ENVIRONMENTedtabularr	297
ENVIRONMENTledgroup	68, 257, 324, 381
ENVIRONMENTledgroupsize	257
PACKAGE(r)(e)ledmac	30
PACKAGEEledmac	11, 65, 90, 262, 319, 320, 376, 377
PACKAGEEledpar	377
PACKAGEEtoolbox	67
PACKAGEParallel	325
PACKAGEReledmac	323, 324
PACKAGEamsgen	275
PACKAGEamsmath	275
PACKAGEbabel	38, 64, 286, 381
PACKAGEbiblatex	61
PACKAGEbidi	37, 38, 69, 380
PACKAGEccaption	75
PACKAGEcolor	63
PACKAGEedmac	1, 6, 10–13, 65, 171, 177, 234, 270, 316, 325, 368
PACKAGEedstanza	1, 13, 268
PACKAGEeledmac	1, 10, 13, 15, 54, 120, 177, 259, 262, 279, 301, 313, 317, 319–321, 372, 374, 376
PACKAGEeledpar	75, 148, 299, 325, 370, 374–376
PACKAGEetex	380
PACKAGEetoolbox	80, 120, 210, 221, 229, 251, 300, 310, 311
PACKAGEfloatrow	61, 324
PACKAGEfootmisc	29, 63, 177, 325
PACKAGEgeometry	14
PACKAGEglossaries	54, 55, 267, 381
PACKAGEhandout	375
PACKAGEhyperlink	216
PACKAGEhyperref	48, 114, 195, 196, 235, 264, 304, 313, 373–375, 382
PACKAGEifluatex	67
PACKAGEifxetex	67
PACKAGEimakeidx	53, 62, 68, 69, 259, 262, 320, 372–374, 376
PACKAGEindextols	266
PACKAGEindextool	320
PACKAGEindextools	53, 62, 68, 69, 76, 259, 262, 266, 320, 376, 381

PACKAGEinputenc	122
PACKAGEledarab	64
PACKAGEledmac	1, 10, 13, 64, 80, 262, 316, 317, 320, 323
PACKAGEledpar	64
PACKAGEMemoir	68, 262, 320, 325, 375
PACKAGEMorewrites	62
PACKAGEMusixtex	374
PACKAGEperpage	381
PACKAGEpolyglossia	36, 64, 115, 147, 167, 381
PACKAGERagged2e	39, 67
PACKAGEreledmac	1, 2, 10, 11, 13–16, 18, 19, 22, 24–27, 29, 32, 35, 38–40, 42, 44, 46, 48, 49, 51–55, 59–63, 65, 66, 82, 84, 90, 91, 94, 95, 97, 105, 106, 113, 143, 149, 152, 156, 177, 202, 210, 214, 221, 229, 239, 242, 247, 262, 279, 299, 300, 312, 313, 320, 321, 323, 324, 379, 382
PACKAGEreledpar	1, 4, 6, 8, 15, 18, 41, 48, 50, 60, 61, 64, 66, 75, 82, 92, 97, 115, 150, 152, 197, 198, 210, 215, 228, 229, 231, 258, 268, 380, 381
PACKAGESuffix	67
PACKAGETabmac	1, 13, 325
PACKAGEuninormalize	27
PACKAGExargs	26, 67
PACKAGExkeyval	66, 228
PACKAGExr	4, 51, 247, 382
PACKAGExref	247
PACKAGExstring	67, 264

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, 21
\advancepageno	1
\Aendnote	24
\affixline@num	1
\affixpstart@num	1
\affixside@note	1
\Afootnote	23
\afternoteX	39
\afterruleX	40
\ampersand	1, 46
\applabel	1, 49
\appref	1, 49
\apprefwithpage	1, 49
\arrangementX	1, 31
\arrangementX@normal	1

<code>\arrangementX@threecol</code>	1
<code>\arrangementX@twocol</code>	1
<code>\at@every@pend</code>	1
<code>\AtEveryPend</code>	1, 18
<code>\AtEveryPstart</code>	1, 18
<code>\AtEveryStanza</code>	1
<code>\AtEveryStopStanza</code>	1
<code>\autopar</code>	1, 17

B

<code>\ballast</code>	62
<code>\ballast@count</code>	1
Beeton, Barbara Ann Neuhaus Friend	17
<code>\beforeeledchapter</code>	1
<code>\beforeinsertingX</code>	38
<code>\beforenotesX</code>	40
<code>\beginnumbering</code>	1, 16
<code>\Bendnote</code>	24
<code>\Bfootnote</code>	23
<code>\bhookgroupX</code>	40
<code>\bhooknoteX</code>	38
<code>\bodyfootmarkA</code>	29
<code>\boxfootnotenumbers</code>	1
Bredon, Simon	12
Breger, Herbert	13, 279
Brey, Gerhard	12
Busard, Hubert L. L.	12
<code>\bypage@false</code>	1
<code>\bypage@true</code>	1
<code>\bypstart@false</code>	1
<code>\bypstart@true</code>	1

C

<code>\c@addcolcount</code>	1
<code>\c@ballast</code>	1
<code>\c@firstlinenum</code>	1
<code>\c@firstsublinenum</code>	1
<code>\c@labidx</code>	1
<code>\c@linenumincrement</code>	1
<code>\c@sublinenumincrement</code>	1
<code>\Cendnote</code>	24
<code>\Cfootnote</code>	23
<code>\ch@ck@l@ck</code>	1
<code>\ch@cksub@l@ck</code>	1
<code>\chapter</code>	1
<code>\check@pb@in@verse</code>	1
Chester, Robert of	12
Claassens, Geert H. M.	12
<code>\colalignX</code>	39
Copernicus, Nicolaus	12

<code>\critext</code>	316
<code>\ctab</code>	1
<code>\ctabtext</code>	1

D

Dekker, Dirk-Jan	63
<code>\Dendnote</code>	24
<code>\Dfootnote</code>	23
<code>\disable@familiarnotes</code>	1
<code>\disable@notes</code>	1
<code>\disable@sidenotes</code>	1
<code>\disablel@dtabfeet</code>	1
<code>\do@actions</code>	1
<code>\do@actions@fixedcode</code>	1
<code>\do@actions@next</code>	1
<code>\do@ballast</code>	1
<code>\do@feetX</code>	1
<code>\do@insidelinehook</code>	1
<code>\do@line</code>	1
<code>\do@linehook</code>	1
<code>\do@lockoff</code>	1
<code>\do@lockoffL</code>	1
<code>\do@lockon</code>	1
<code>\do@lockonL</code>	1
<code>\doedindexlabel</code>	1
<code>\doendnotes</code>	1, 24
<code>\doendnotesbysection</code>	1, 24
<code>\doinsidelinehook</code>	1, 22
<code>\dolinehook</code>	1, 22
<code>\dosplits</code>	1
Downes, Michael	62, 156, 158
<code>\doxtrafeet</code>	1
<code>\dummy@edtext</code>	1
<code>\dummy@edtext@showlemma</code>	1
<code>\dummy@ref</code>	1

E

<code>\edaftertab</code>	1, 57, 292
<code>edarrayc</code> (environment)	55
<code>edarrayl</code> (environment)	55
<code>edarrayr</code> (environment)	55
<code>\edatleft</code>	1, 57
<code>\edatright</code>	1, 57
<code>\edbeforetab</code>	1, 57, 292
<code>\edfilldimen</code>	1
<code>\edfont@info</code>	1
<code>\EDINDEX</code>	1
<code>\edindex</code>	1, 52
<code>\edindexlab</code>	1, 54
<code>\EDLABEL</code>	1

<code>\edlabel</code>	1, 47
<code>\edlabelE</code>	1, 49
<code>\edlabelS</code>	1, 49
<code>\edlabelSE</code>	1, 49
<code>\edlineref</code>	1, 47
<code>\edmakelabel</code>	1, 48
<code>\edpageref</code>	1, 47
<code>\edrowfill</code>	1, 56
<code>\EDTAB</code>	1
<code>\edtabcolsep</code>	1, 56
<code>\EDTABINDENT</code>	1
<code>\edtabindent</code>	1
<code>\EDTABtext</code>	1
<code>edtabularc</code> (environment)	55
<code>edtabularl</code> (environment)	55
<code>edtabularr</code> (environment)	55
<code>\EDTEXT</code>	1
<code>\edtext</code>	1, 23
<code>\edvertdots</code>	1, 58
<code>\edvertline</code>	1, 58
<code>\Endnote</code>	24
<code>\Efootnote</code>	23
<code>\eled@chapter</code>	1
<code>\eled@section</code>	1
<code>\eled@sectioning@out</code>	1
<code>\eled@subsection</code>	1
<code>\eled@subsubsection</code>	1
<code>\eledchapter</code>	1
<code>\eledchapter*</code>	1
<code>\eledsection</code>	1
<code>\eledsection*</code>	1
<code>\eledsubsection</code>	1
<code>\eledsubsection*</code>	1
<code>\eledsubsubsection</code>	1
<code>\eledsubsubsection*</code>	1
<code>\enablel@dtabfeet</code>	1
<code>\end@lemmas</code>	1
<code>\endashchar</code>	1
<code>\endline@num</code>	1
<code>\endlock</code>	1, 21
<code>\endminipage</code>	1
<code>\endnumbering</code>	1, 16
<code>\endpage@num</code>	1
<code>\endprint</code>	1
<code>\endquotation</code>	1
<code>\endquote</code>	1
<code>\endsub</code>	1, 20
<code>\endsubline@num</code>	1
environments:	
<code>edarrayc</code>	55

edarrayl	55
edarrayr	55
edtabularc	55
edtabularl	55
edtabularr	55
ledgroup	46
ledgroupsize	46
minipage	46
Euclid	12
\extensionchars	1, 61

F

\f@x@l@cks	1
Fairbairns, Robin	29
\first@linenum@out@false	1
\first@linenum@out@true	1
\firstlinenum	1, 19
\firstseriesX@	1
\firstsublinenum	1, 19
\firstXseries@	1
\fix@page	1
\flag@end	1
\flag@start	1
\flagstanza	1, 46
\flush@notes	1
\fnpos	1, 30
Folkerts, Menso	12
\footfootmarkA	29
\footfudgefiddle	1, 62
\footnote	1
\footnoteA	29
\footnoteB	29
\footnoteC	29
\footnoteD	29
\footnoteE	29
\footnotelang@lua	1
\footnotelang@poly	1
\footnoteoptions@	1
\footplitskips	1
\fullstop	1

G

Gädeke, Nora	13
\get@edindex@hyperref	1
\get@edindex@ledinnote@command	1
\get@fnmark	1
\get@fnmarkX	1
\get@index@command	1
\get@linelistfile	1
\get@sw@txt	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>	38
<code>\hidenumbering</code>	1, 22
<code>\Hilfsbox</code>	1
<code>\hilfsbox</code>	1
<code>\hilfscount</code>	1
<code>\HILFSskip</code>	1
<code>\Hilfsskip</code>	1
<code>\hilfsskip</code>	1
<code>\hsizethreecolX</code>	39
<code>\hsizetwocolX</code>	39
<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@dhidenumbers</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@startendok</code>	1
<code>\ifl@imakeidx</code>	1
<code>\ifl@indextools</code>	1
<code>\ifledfinal</code>	1, 61
<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	13
-----------------	----

K

Kabelschacht, Alois 145

L

\l@advance@parledgroup@beforenormalnotes 1
\l@d@add 1
\l@d@nums 1
\l@d@section 1
\l@d@set 1
\l@d@Xend 1
\l@dampcount 1
\l@dbfnote 1
\l@dcheckcols 1
\l@dcheckstartend 1
\l@dchset@num 1
\l@dcolcount 1
\l@dcollect@body 1
\l@dcollect@body 1
\l@dcolwidth 1
\l@dcsnote 1
\l@dcsnotetext 1
\l@dcsnotetext@l 1
\l@dcsnotetext@r 1
\l@ddodoreinextrafeet 1
\l@dedbeginmini 1
\l@dedendmini 1
\l@demptyd@ta 1
\l@dend@close 1
\l@dend@open 1
\l@dend@stuff 1
\l@dend@Xfalse 1
\l@dend@Xtrue 1
\l@denbody 1
\l@dfambeginmini 1
\l@dfamendmini 1
\l@dfeetbeginmini 1
\l@dfeetendmini 1
\l@dgetline@margin 1
\l@dgetlock@disp 1
\l@dgetref@num 1
\l@dgetsidenote@margin 1
\l@dobblearg 1
\l@dlabel@parse 1
\l@dld@ta 1
\l@dldp@rbox 1
\l@dlsn@te 1
\l@dlsnote 1
\l@dmake@labels 1
\l@dmodfordtext 1
\l@dnullfills 1

<code>\ldnumpstartsL</code>	1
<code>\ldoldold@footnotetext</code>	1
<code>\ldp@rsefootspec</code>	1
<code>\ldparsedendline</code>	1
<code>\ldparsedendpage</code>	1
<code>\ldparsedendsub</code>	1
<code>\ldparsedstartline</code>	1
<code>\ldparsedstartpage</code>	1
<code>\ldparsedstartsub</code>	1
<code>\ldpush@begins</code>	1
<code>\ldrd@ta</code>	1
<code>\ldref@undefined</code>	1
<code>\ldrestorefills</code>	1
<code>\ldrestoreforedtext</code>	1
<code>\ldrp@rbox</code>	1
<code>\ldrsn@te</code>	1
<code>\ldrsnote</code>	1
<code>\ldsetmaxcolwidth</code>	1
<code>\ldskipnumberfalse</code>	1
<code>\ldskipnumbertrue</code>	1
<code>\ldtabaddcols</code>	1
<code>\ldtabnoexpands</code>	1
<code>\ldunboxmpfoot</code>	1
<code>\ldunhbox@line</code>	1
<code>\ldzeropenalties</code>	1
<code>\label</code>	49
<code>\labelpstartfalse</code>	1
<code>\labelpstarttrue</code>	1, 18
<code>\labelref@list</code>	1
<code>\labelrefsparseline</code>	1
<code>\labelrefsparsesubline</code>	1
<code>\last@page@num</code>	1
Lavagnino, John	11
<code>\led@check@nopb</code>	1
<code>\led@check@pb</code>	1
<code>\led@err@AutoparNotNumbered</code>	1
<code>\led@err@Edtextoutsidepstart</code>	1
<code>\led@err@EdtextWithoutFootnote</code>	1
<code>\led@err@FootnoteWithoutEdtext</code>	1
<code>\led@err@HighEndColumn</code>	1
<code>\led@err@LineationInNumbered</code>	1
<code>\led@err@LowStartColumn</code>	1
<code>\led@err@ManyLeftnotes</code>	1
<code>\led@err@ManyRightnotes</code>	1
<code>\led@err@ManySidenotes</code>	1
<code>\led@err@NumberingNotStarted</code>	1
<code>\led@err@NumberingShouldHaveStarted</code>	1
<code>\led@err@NumberingStarted</code>	1
<code>\led@err@NumberingWithoutPstart</code>	1
<code>\led@err@PendNoPstart</code>	1

<code>\led@err@PendNotNumbered</code>	1
<code>\led@err@PstartInPstart</code>	1
<code>\led@err@PstartNotNumbered</code>	1
<code>\led@err@ReverseColumns</code>	1
<code>\led@err@TooManyColumns</code>	1
<code>\led@err@UnequalColumns</code>	1
<code>\led@error@fail@patch@@doclearpage</code>	1
<code>\led@error@fail@patch@@iiiminipage</code>	1
<code>\led@error@fail@patch@@makecol</code>	1
<code>\led@error@fail@patch@@reinserts</code>	1
<code>\led@error@fail@patch@endminipage</code>	1
<code>\led@error@ImakeidxAfterEledmac</code>	1
<code>\led@error@IndextoolsAfterEledmac</code>	1
<code>\led@mess@NotesChanged</code>	1
<code>\led@mess@SectionContinued</code>	1
<code>\led@nopb</code>	1
<code>\led@nopbnum</code>	1
<code>\led@pb</code>	1
<code>\led@pb@setting</code>	1
<code>\led@pbnum</code>	1
<code>\led@reinit@index@fornote</code>	1
<code>\led@set@index@fornote</code>	1
<code>\led@toksa</code>	1
<code>\led@toksb</code>	1
<code>\led@warn@AppLabelOutEdtext</code>	1
<code>\led@warn@BadAction</code>	1
<code>\led@warn@BadAdvancelineLine</code>	1
<code>\led@warn@BadAdvancelineSubline</code>	1
<code>\led@warn@BadLineation</code>	1
<code>\led@warn@BadLinenummargin</code>	1
<code>\led@warn@BadLockdisp</code>	1
<code>\led@warn@BadSetline</code>	1
<code>\led@warn@BadSetlinenum</code>	1
<code>\led@warn@BadSidenotemargin</code>	1
<code>\led@warn@BadSublockdisp</code>	1
<code>\led@warn@DuplicateLabel</code>	1
<code>\led@warn@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>	46
<code>ledgroupsize (environment)</code>	46
<code>\ledinnernote</code>	1, 51
<code>\ledinnote</code>	1
<code>\ledinnotehyperpage</code>	1
<code>\ledinnotemark</code>	1

<code>\ledleftnote</code>	1, 51
<code>\ledlinenum</code>	1
<code>\ledllfill</code>	1
<code>\ledlsnotefontsetup</code>	1, 52
<code>\ledlsnotesep</code>	1, 52
<code>\ledlsnotewidth</code>	1, 52
<code>\lednopb</code>	1, 60
<code>\lednopbinversetrue</code>	60
<code>\lednopbnum</code>	1
<code>\ledouternote</code>	51
<code>\ledouterote</code>	1
<code>\ledpb</code>	1, 60
<code>\ledpbnum</code>	1
<code>\ledpbsetting</code>	1, 60
<code>\ledrightnote</code>	1, 51
<code>\ledrlfill</code>	1
<code>\ledrsnotefontsetup</code>	1, 52
<code>\ledrsnotesep</code>	1, 52
<code>\ledrsnotewidth</code>	1, 52
<code>\ledsectnomark</code>	1
<code>\ledsectnotoc</code>	1
<code>\ledsetnormalparstuff@common</code>	1
<code>\ledsetnormalparstuffX</code>	1
<code>\ledsidenote</code>	1, 51
<code>\leftctab</code>	1
<code>\leftlinenum</code>	1, 20
<code>\leftltab</code>	1
<code>\leftnoteupfalse</code>	52
<code>\leftpstartnum</code>	1
<code>\leftrtab</code>	1
Leibniz	13
<code>\lemma</code>	1, 25
<code>\letsforverteilen</code>	1
<code>\line@list</code>	1
<code>\line@list@stuff</code>	1
<code>\line@list@version</code>	1
<code>\line@margin</code>	1
<code>\line@num</code>	1
<code>\line@set</code>	1
<code>\lineation</code>	1, 20
<code>\linenum</code>	1, 25
<code>\linenum@out</code>	1
<code>\linenumberlist</code>	1, 19
<code>\linenumberstyle</code>	1, 21
<code>\linenumincrement</code>	1, 19
<code>\linenummargin</code>	1, 20
<code>\linenumr@p</code>	1
<code>\linenumrep</code>	1
<code>\linenumsep</code>	1, 20
<code>\linerrangesep@</code>	1

\list@clear	1
\list@clearing@reg	1
\list@create	1
\lock@disp	1
\lock@off	1
\lock@on	1
\lockdisp	1, 21
Lorch, Richard	12
\ltab	1
\ltabtext	1
Luecking, Dan	67

M

\m@mmf@check	1
\m@mmf@prepare	1
\M@sect	1
\makehboxofhboxes	1
\manage stanza@modulo	1
\maxhnotesX	41
Mayer, Gyula	13
\measurebody	1
\measurecell	1
\measuremrow	1
\measuretbody	1
\measuretcell	1
\measuretrow	1
Middleton, Thomas	13, 88
minipage (environment)	46
Mittelbach, Frank	12
\morenoexpands	1, 63
\mpfnpos	1, 30
\mpnormalfootgroup	1
\mpnormalfootgroupX	1
\mpnormalvfootnote	1
\mpnormalvfootnoteX	1
\mppara@footgroupX	1
\mppara@vfootnoteX	1
\mpparafootgroup	1
\mpparavfootnote	1
\mpthreecolfootgroup	1
\mpthreecolfootgroupX	1
\mpthreecolfootsetup	1
\mpthreecolfootsetupX	1
\mptwocolfootgroup	1
\mptwocolfootgroupX	1
\mptwocolfootsetup	1
\mptwocolfootsetupX	1
\multfootsep	1, 29
\multiplefootnotemarker	1

N

<code>\n@num</code>	1
<code>\n@num@stanza</code>	1
<code>\new@line</code>	1
<code>\newhookarg@specific</code>	1
<code>\newhookcommand@series</code>	1
<code>\newhookcommand@series@reload</code>	1
<code>\newhooktoggle@series</code>	1
<code>\newhooktoggle@series@reload</code>	1
<code>\newhooktoggle@specific</code>	1
<code>\newseries@</code>	1
<code>\newverse</code>	1
<code>\NEXT</code>	1
<code>\no@expands</code>	1
<code>\noeledsec</code>	59
<code>\nomk@</code>	1
<code>\normal@footnotemarkX</code>	1
<code>\normal@page@break</code>	1
<code>\normal@pars</code>	1
<code>\normalbfnoteX</code>	1
<code>\normalbodyfootmarkX</code>	1
<code>\normalfootfmt</code>	1
<code>\normalfootfmtX</code>	1
<code>\normalfootfootmarkX</code>	1
<code>\normalfootgroup</code>	1
<code>\normalfootgroupX</code>	1
<code>\normalfootnoterule</code>	1
<code>\normalfootnoteruleX</code>	1
<code>\normalfootstart</code>	1
<code>\normalfootstartX</code>	1
<code>\normalvfootnote</code>	1
<code>\normalvfootnoteX</code>	1
<code>\notefontsizeX</code>	37
<code>\notenumfontX</code>	37
<code>\noteschanged@false</code>	1
<code>\noteschanged@true</code>	1
<code>\noteswidthliketwocolumnsX</code>	41
<code>\nulledindex</code>	1
<code>\nullsetzen</code>	1
<code>\num@lines</code>	1
<code>\numberedpar@false</code>	1
<code>\numberedpar@true</code>	1
<code>\numberingfalse</code>	1
<code>\numberingtrue</code>	1
<code>\numberlinefalse</code>	19
<code>\numberlinetrue</code>	19
<code>\numberpstartfalse</code>	1, 18
<code>\numberpstarttrue</code>	1, 18
<code>\numberstanzafalse</code>	45
<code>\numberstanzatrue</code>	45

\numlabfont 1, 42

O

\old@hsize 1
 \one@line 1
 optioncontinuousnumberingwithcolumns 382
 optioninnnote 381
 optioninnote 381
 optionlinrangesep 228
 optionnocritical 381
 optionnoend 381
 optionnotenumber 381

P

\page@action 1
 \page@num 1
 \pagelinesep 1, 53
 \pageno 1
 \pageref 49
 \par@line 1
 \para@footgroupX 1
 \para@footsetup 1
 \para@footsetupX 1
 \para@vfootnoteX 1
 \parafootfmt 1
 \parafootfmtX 1
 \parafootgroup 1
 \parafootsepX 39
 \parafootstart 1
 \parafootstartX 1
 \paravfootnote 1
 \parindentX 38
 \pausenumbering 1, 18
 \pend 1, 16
 Plato of Tivoli 12
 \postbodyfootmark 1
 \prebodyfootmark 1
 \prenotesX 40
 \prepare@edindex@fornote 1
 \prepare@prenotesX 1
 \prepare@preXnotes 1
 \prev@nopb 1
 \prev@pb 1
 \prevpage@num 1
 \preXnotes 1, 40
 \preXnotes@ 1
 \print@eledsection 1
 \print@footnoteXrule 1
 \print@leftmargin@eledsection 1
 \print@lemma 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>	47

Q

<code>\quotation</code>	1
<code>\quote</code>	1

R

<code>\raggedX</code>	40
<code>\raw@text</code>	1
<code>\rbracket</code>	1
<code>\read@linelist</code>	1
<code>\ref</code>	49
<code>\Relax</code>	1
<code>\reledmac@error</code>	1
<code>\reledmac@warning</code>	1
<code>\removehboxes</code>	1
<code>\resetprevline@</code>	1, 92
<code>\resetprevpage@</code>	1
<code>\resetprevpage@num</code>	92
<code>\restore@familiarnotes</code>	1
<code>\restore@notes</code>	1
<code>\restore@sidenotes</code>	1
<code>\resumenumbering</code>	1, 18
<code>\rightctab</code>	1
<code>\rightlinenum</code>	1, 20
<code>\rightlftab</code>	1
<code>\rightnoteupfalse</code>	52
<code>\rightrtab</code>	1
<code>\rightstartnum</code>	1

\rigidbalance	1
\rigidbalanceX	1
\rtab	1
\rtabtext	1

S

Sacrobosco	13
\sameword	1, 26
\sameword@inedtext	1
Schöpf, Rainer	12
\section@num	1
\select@lemfont	1
\select@lemfont	1, 42
\Seref	1, 49
\Serefonlypage	49
\Serefwithpage	1, 49
\series	1
\seriesatbegin	1, 30
\seriesatend	1, 30
\set@line	1
\set@line@action	1
\setapprefprefixmore	49
\setapprefprefixsingle	49
\setcommand@series	1
\sethangingsymbol	1, 44
\setistwofollowinglines	1
\setl@dlp@rbox	1
\setl@drpr@box	1
\setline	1, 21
\setlinenum	1, 21
\setmcellcenter	1
\setmcellleft	1
\setmcellright	1
\setmrowcenter	1
\setmrowleft	1
\setmrowright	1
\setnoteswidthliketwocolumnsX@	1
\setnotesXpositionliketwocolumns@	1
\setprintendlines	1
\setprintlines	1
\setSerefonlypageprefixmore	50
\setSerefonlypageprefixsingle	50
\setSerefprefixmore	50
\setSerefprefixsingle	50
\setsidenotesep	52
\setstanzaindents	1, 43
\setstanzapenalties	1, 44
\setstanzavalues	1
\settcellcenter	1
\settcellleft	1

<code>\settcclright</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, 61
<code>\showwordrank</code>	1, 28
<code>\sidenote@margin</code>	1
<code>\sidenotemargin</code>	1, 51
<code>\sidepstartnumtrue</code>	18
<code>\skip@lockoff</code>	1
<code>\skipnumbering</code>	1, 22
<code>\splitoff</code>	1
<code>\spreadmath</code>	1, 56
<code>\spreadtext</code>	1, 56
<code>\stanza</code>	1, 43
<code>\stanza@count</code>	1
<code>\stanza@hang</code>	1
<code>\stanza@line</code>	1
<code>\stanzaindent</code>	1, 44
<code>\stanzaindent*</code>	1, 44
<code>\stanzaindentbase</code>	1, 43
<code>\stanzanumwrapper</code>	1, 46
<code>\startlock</code>	1, 21
<code>\startsub</code>	1, 20
<code>\stepl@dcclcount</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, 47
<code>\sublines@false</code>	1
<code>\sublines@true</code>	1
<code>\sublock@disp</code>	1
<code>\sublockdisp</code>	1
Sullivan, Wayne	12, 13, 62, 75, 79, 156, 157, 234, 268
<code>\sza@penalty</code>	1

T

<code>\tabHilfbox</code>	1
<code>\tabhilfbox</code>	1
<code>\theadcolcount</code>	1

<code>\theedtext</code>	1
<code>\theendpageline</code>	1
<code>\thefootnoteA</code>	29
<code>Theodosius</code>	13
<code>\thepageline</code>	1
<code>\thepstart</code>	1, 18
<code>\thestanza</code>	1, 45
<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	
<code>\unvxhX</code>	1
V	
<code>Vamana</code>	13
<code>\variab</code>	1
<code>\vbfnoteX</code>	1
<code>\vl@dbfnote</code>	1
<code>\vl@dcnote</code>	1
<code>\vl@dlsnote</code>	1
<code>\vl@drsnote</code>	1
<code>\vnumfootnoteX</code>	1
W	
<code>Whitney, Ron</code>	12
<code>\widthX</code>	41
<code>\wrap@edcrossref</code>	1
<code>\wrapcontentX</code>	38
<code>\wrapped@bodyfootmarkX</code>	1
<code>\wrapped@footfootmarkX</code>	1
<code>Wujastyk, Dominik</code>	11
X	
<code>\X@doreinfeet</code>	1

\Xafterlemmaseparator	36
\Xafternote	39
\Xafternumber	34
\Xafterrule	40
\Xaftersymlinenum	34
\Xarrangement	1, 31
\Xarrangement@normal	1
\Xarrangement@paragraph	1
\Xarrangement@threecol	1
\Xarrangement@twocol	1
\Xbeforeinserting	38
\Xbeforelemmaseparator	36
\Xbeforenotes	40
\Xbeforenumber	32, 34
\Xbeforesymlinenum	34
\Xhookgroup	40
\Xhooknote	38
\Xboxlinenum	34
\Xboxlinenumalign	34
\Xboxsymlinenum	34
\Xcolalign	39
\Xdo@feet	1
\xedindex	1
\xedlabel	1
\xedtext	1
\Xendafterenumber	34
\Xendafterlemmaseparator	36
\Xendafternote	42
\Xendafterpagenumber	35
\Xendaftersymlinenum	34
\Xendahookinplaceofnumber	35
\Xendahooklinenum	35
\Xendbeforelemmaseparator	36
\Xendbeforenumber	34
\Xendbeforepagenumber	35
\Xendbeforesymlinenum	34
\Xendbhookinplaceofnumber	35
\Xendbhooklinenum	35
\Xendbhooknote	38
\Xendboxendlinenumalign	35
\Xendboxlinenum	35
\Xendboxlinenumalign	35
\Xendboxstartlinenumalign	35
\Xendboxsymlinenum	34
\Xendhangindent	38
\Xendinplaceofflemmaseparator	36
\Xendinplaceofnumber	34
\Xendlemmadisablefontselection	37
\Xendlemmafont	37
\Xendlemmaseparator	36

\Xendlineprefixmore	35
\Xendlineprefixsingle	35
\Xendlinerangeseparator	32
\Xendmorethantwolines	33
\Xendnonumber	33
\Xendnotefontsize	37
\Xendnotenumfont	36
\Xendnumberonlyfirstinline	32
\Xendnumberonlyfirstintwolines	32
\Xendparagraph	42
\Xendsep	42
\Xendsublinesep	33
\Xendsymlinenum	32
\Xendtwolines	33
\Xendtwolinesbutnotmore	33
\Xendwrapcontent	38
\xflagref	1
\Xhangindent	38
\Xhsizethreecol	39
\Xhsizetwocol	39
\Xinplaceoflemmaseparator	36
\Xinplaceofnumber	34
\Xinsertparafootsep	1
\Xledsetnormalparstuff	1
\xleft@appenditem	1
\Xlemmadisablefontselection	37
\Xlemmafont	37
\Xlemmaseparator	36
\Xlinerrangeseparator	32
\xlineref	1, 47
\Xmaxhnotes	41
\Xmorethantwolines	32
\Xnolemmaseparator	1, 36
\Xnonbreakableafternumber	34
\Xnonumber	33
\Xnotefontsize	37
\Xnotenumfont	36
\Xnoteswidthliketwocolumns	41
\Xnumberonlyfirstinline	31
\Xnumberonlyfirstintwolines	31
\Xonlypstart	33
\xpageref	1, 47
\Xparafootsep	39
\Xparindent	38
\Xpstart	33
\Xpstarteverytime	33
\xpstartref	1, 47
\XR@test	1
\XR@test@mac	1
\XR@test@mac@test	1

<code>\Xragged</code>	40
<code>\xright@appenditem</code>	<u>1</u>
<code>\Xrigidbalance</code>	<u>1</u>
<code>\Xstanza</code>	33
<code>\Xstanzaseparator</code>	33
<code>\xsublineref</code>	<u>1</u> , 47
<code>\Xsublinesep</code>	33
<code>\Xsymlinenum</code>	32
<code>\Xtwolines</code>	32
<code>\Xtwolinesonlyinsamepage</code>	32
<code>\Xtxtbeforenotes</code>	40
<code>\Xunvxh</code>	<u>1</u>
<code>\Xwidth</code>	41
<code>\Xwrapcontent</code>	38
<code>\Xwrapendlemma</code>	37
<code>\Xwraplemma</code>	37
<code>\xxref</code>	<u>1</u> , 48

Z

<code>\zz@@@</code>	<u>1</u>
---------------------------	----------

Change History

v0.1.0.	
General: First public release	1
v0.2.0.	
\ifl@dmemoir: Added \ifl@dmemoir for memoir class having been used	68
\morenoexpands: Added \l@dtabnoexpands to \no@expands	113
\reledmac@error: Added \eledmac@error and replaced error messages	69
General: Added tabmac code, and extended indexing	1
v0.2.1.	
\@lab: Removed page setting from \@lab	237
\doxtrafeet: Renamed \doxtrafeet to \l@ddoxtrafeet	229
\edlabel: Tweaked \edlabel to get correct page numbers	234
\l@ddodoreinxtrafeet: Renamed \dodoreinxtrafeet to \l@ddodoreinxtrafeet	230
\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	113
\zz@@@: Minor change to \zz@@@	234
General: Added text about normal labeling	49
Bug fixes and match with mempatch v1.8	1
Major changes to insert code when memoir is loaded	232
v0.2.2.	
\footfudgefiddle: Added \footfudgefiddle	154
\line@list@stuff: Added initial write of page number in \line@list@stuff	106
\para@footsetup: Added \footfudgefiddle to \para@footsetup	155
\para@footsetupX: Added \footfudgefiddle to \para@footsetupX	192
General: Improved paragraph footnotes	1
New Dekker example	1
Used \providecommand for \@gobblethree to avoid clash with the amsfonts package	74
v0.3.0.	
\@lab: Replaced \the\line@num by \linenumr@p\line@num in \@lab, and similar for sub-lines	237
\@nl@reg: Added a bunch of code to \@nl for handling \setlinenum	95
\ledlinenum: Added \linenumr@p and \sublinenum@rep to \leftlinenum and \rightlinenum	87
\linenumberlist: Added \linenumberlist mechanism	75
\printendlines: Added \linenumr@p and \sublinenumr@p to \printendlines	208
\printlines: Added \linenumr@p and \sublinenumr@p to \printlines	176
\sublinenumr@p: Added \linenumberstyle and \sublinenumberstyle	86
General: Includes edstanza and more	1
v0.3.1.	
General: Not released. Added remarks about the parallel package	1
v0.4.0.	
\@iiiminipage: Modified kernel \@iiiminipage and \endminipage to cater for critical footnotes	256
\Xarrangement@normal: Added minpage footnote setup to \footnormal	150
\edtext: Added \showlemma to \edtext	114
\l@dfeetendmini: Added \l@dfeetbeginmini, \l@dfeetendmini and all their supporting code	254

\mpnormalfootgroup: Added \mpnormalfootgroup	153
\mpnormalvfootnote: Added \mpnormalvfootnote	151
\showlemma: Added \showlemma	75
General: Added final/draft options	66
Added ledgroup environment	257
Added ledgroupsize environment	257
Added minipage, etc., support	1
v0.4.1.	
\Xdo@feet: Changed \Xdo@feet code for easier extensions	229
\edindex: Let eledmac take advantage of memoir's indexing	262
\print@Xnotes: Added \opXfeet	229
General: Added code for changing \doclearpage	232
Not released. Minor editorial improvements and code tweaks	1
Only change \@footnotetext and \@footnotemark if memoir not used	178
v0.5.0.	
\@footnotetext: Enabled regular \footnote in numbered text	178
\@xympar: Eliminated \marginpar disturbance	248
General: Added left and right side notes	248
Added sidenotes, familiar footnotes in numbered text	1
v0.5.1.	
\affixline@num: Changed \affixline@num to cater for sidenotes	138
\l@edgetsidenote@margin: Added \sidenotemargin and \sidenote@margin	248
General: Added moveable side note	248
Fixed right line numbers killed in v0.5	1
Only change \hsize in ledgroupsize environment otherwise page number can be in wrong place	257
v0.6.0.	
\@lopR: Added \@pend, \@pendR, \@lopL and \@lopR in anticipation of parallel processing	97
\@nl@reg: Added \fix@page to \@nl	95
Extended \@nl to include the page number	95
\fix@page: Added \last@page@num and \fix@page	97
\get@thisfootnote: Changed \l@dbfnote and \vl@dbfnote as originals could give incorrect markers in the footnotes	179
\new@line: Extended \new@line to output page numbers	107
General: Fixed long paragraphs looping	1
Fixed minor typos	1
Prepared for eledpar package	1
v0.7.0.	
\@nl@reg: Added \@nl@reg	95
\@ref@reg: Added \@ref@reg	104
\affixline@num: Added skipnumering to \affixline@num	138
\do@actions@fixedcode: Added \do@actions@fixedcode	137
\do@actions@next: Added number skipping to \do@actions	135
\do@insidelinehook: Added \do@linehook for use in \do@line	133
\endnumbering: Changed \endnumbering for eledpar	78
\fix@l@cks: Added \ch@cksub@l@ck, \ch@ck@l@ck and \fix@l@cks	140
\footplitskips: Added \footplitskips for use in many footnote styles	148
\get@linelistfile: Added \get@linelistfile	94
\initnumbering@reg: Added \initnumbering@reg	77

\l@advance@parledgroup@beforenormalnotes: Added \l@dunboxmpfoot containing some common code	256
\l@dcsnotetext@r: Added \l@emptyd@ta	134
\l@dgetline@margin: Added \l@dgetline@margin	83
\l@dgetlock@disp: Added \l@dgetlock@disp	85
\l@dgetsidenote@margin: Added \l@dgetsidenote@margin	248
\l@dnumpstartsL: Added \l@dnumpstartsL, \ifl@dpairing and \ifpst@rted for/from eledpar	75
\l@drsn@te: Added \l@dlsn@te and \l@drsn@te for use in \do@line	134
\l@dzeropenalties: Added \l@dzeropenalties	129
\ledlinenum: Added \ledlinenum for use by \leftlinenum and \rightlinenum	87
\line@list@stuff: Deleted \page@start from \line@list@stuff	106
\list@clearing@reg: Added \list@clearing@reg	94
\n@num: Added \n@num	102
\normalbfnoteX: Removed extraneous space from \normalbfnoteX	184
\resumenumbering: Changed \resumenumbering for eledpar	79
\setprintendlines: Added \setprintendlines for use by \printendlines	206
\setprintlines: Added \setprintlines for use by \printlines	172
\skipnumbering: Added \skipnumbering and supports	110
\sublinenumincrement: Added \firstlinenum, \linenumincrement, \firstsublinenum and \linenumincrement	84
\sublinenumr@p: Using \linenumrep instead of \linenumr@p	86
Using \sublinenumrep instead of \sublinenumr@p	86
\vnumfootnoteX: Removed extraneous space from \vnumfootnoteX	186
General: eledmac having been available for 2 years, deleted the commented out original edmac texts	1
Maieul Rouquette new maintainer	1
Made macros of all messages	69
Replaced all \interAfootnotelinepenalty, etc., by just \interfootnotelinepenalty	1
Tidying up for eledpar and ledarab packages	1
v0.8.0.	
General: Bug on endnotes fixed: in a // text, all endnotes will print and be placed at the ends of columns ()	1
v0.8.1.	
General: Bug on \edtext ; \critex ; \lemma fixed: we can now us non-switching commands	1
v0.9.0.	
General: No more ledpatch. All patches are now in the main file.	1
v0.9.1.	
General: Fix some bugs linked to integrating ledpatch on the main file.	1
v0.10.0.	
General: Corrections to \section and other titles in numbered sections	1
v0.11.0.	
General: Makes it possible to add a symbol on each verse's hanging, as in French typography. Redefines the command \hangingsymbol to define the character.	1
v0.12.0.	
\l@dnumpstartsL: Added \ifledRcol and \ifnumberingR for/from eledpar	75
General: For compatibility with eledpar, possibility to use \autopar on the right side.	1
Possibility to number \pstart.	18

Possibility to number the pstart with the commands <code>\numberpstarttrue</code>	1
v0.12.1.	
General: Don't number <code>\pstarts</code> of stanza.	1
The numbering of <code>\pstarts</code> restarts on each <code>\beginnumbering</code>	1
v0.13.0.	
<code>\managestanza@modulo</code> : New <code>stanzaindentsrepetition</code> counter to repeat stanza indents every n verses.	270
General: New <code>stanzaindentsrepetition</code> counter to repeat stanza indents every n verses. . .	43
New <code>stanzaindentsrepetition</code> counter: to repeat stanza indents every n verses.	1
v0.13.1.	
General: <code>\thepstartL</code> and <code>\thepstartR</code> use now <code>\bfseries</code> and not <code>\bf</code> , which is deprecated and makes conflicts with memoir class.	1
v0.14.0.	
<code>\edlabel</code> : Tweaked <code>\edlabel</code> to get correct line number if the command is first element of a paragraph.	234
General: Tweaked <code>\edlabel</code> to get correct line number if the command is first element of a paragraph.	1
v0.15.0.	
<code>\affixline@num</code> : Line numbering can be disabled.	138
<code>\ifinserthangingsymbol</code> : New management of <code>hangingsymbol</code> insertion, preventing undesirable insertions.	269
<code>\printlines</code> : Line numbering can be reset at each <code>pstart</code>	175
General: Line numbering can be reset at each <code>pstart</code>	81
Possibility to print <code>\pstart</code> number inside.	18
v0.17.0.	
<code>\ifinserthangingsymbol</code> : New new management of <code>hangingsymbol</code> insertion, preventing undesirable insertions.	269
v1.0.0.	
<code>\morenoexpands</code> : Change to be compatible with new features	113
General: <code>\lemma</code> can contain commands.	25
Debug in lineation command	20
New generic commands to customize footnote display.	30, 220
Options <code>nonum</code> and <code>nosep</code> in <code>\Xfootnote</code>	23
Options of <code>\Xfootnotes</code>	146
Possibility to have commands in sidenotes.	51
Some compatibility break with <code>eledmac</code> . Change of name: <code>eledmac</code>	1
v1.0.1.	
General: Correction on <code>\Xnumberonlyfirstinline</code> with lineation by <code>pstart</code> or by page. . .	31
v1.1.0.	
<code>\preXnotes</code> : New skip <code>\preXnotes@</code>	200
<code>\settoggle@series</code> : <code>\settoggle@series</code> switch the global value of the toggle, not only the local value.	221
General: Add <code>\labelpstarttrue</code>	18
Add <code>\Xnumberonlyfirstintwolines</code>	31
Add <code>\Xpstart</code> and <code>\Xonlypstart</code>	33
New hook to add arbitrary code at the beginning of the notes	38
New options for block of notes.	40
New package option: <code>parapparatus</code>	1
New tools to change order of series	219
Sectioning commands.	58

v1.2.0.	
\endquote:	Compatibility of \ledchapter with the <i>memoir</i> class. 297
\preXnotes:	Debug in familiar footnotes (bug introduced by v1.1). 200
v1.3.0.	
\endquote:	<i>Quotation</i> and quote environment inside numbered sections. 297
v1.4.0.	
\edtext:	Compatibility of \edtext with the right-to-left direction (with Polyglossia). 114
\ledsetnormalparstuffX:	Direction of footnotes with polyglossia. 196
\newseries@:	Remembers the language of the lemma, in order to create a correct direction for the footnote separator. 212
\rbraacket:	Switch the right bracket to a left bracket when the lemma is RTL (needs polyglossia or LuaTeX). 167
General:	Compatibility with LuaTeX of RTL notes. 1
v1.4.1.	
\affixside@note:	Remove spurious spaces. 253
\endquote:	New option <i>noquotation</i> 297
\get@thisfootnote:	Compatibility of standard footnotes with eledmac when these footnotes contain any commands. 179
\labelrefsparsesubline:	Fix bug with \edlabel. 236
v1.4.2.	
General:	Debug with some special classes. 1
v1.4.3.	
General:	Add \Xnonbreakableafternumber. 34
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. 297
v1.4.8.	
General:	Corrects a bug with parallel texts introduced by 1.1. 1
v1.4.9.	
\normalbfnoteX:	Allow to redefine \thefootnoteX with alph when some packages are loaded. 184
v1.5.0.	
\do@insidelinehook:	Added \do@insidelinehook for use in \do@line 133
\edindex:	Compatibility with imakeidx package, and possibility to use multiple index with \edindex. 262
General:	Correct indexing when the call is made in critical notes. 259
v1.5.1.	
\managestanza@modulo:	Correct stanzaindentsrepetition counter 270
\normalvfootnoteX:	Fix bug with normal familiar footnotes when mixing RTL and LTR text. 182
v1.6.0.	
\newverse:	Add \falseverse macro. 273
v1.6.1.	
\AtEveryPstart:	Spurious space in \pstart. 126

\ifinserthangingsymbol: Hang verse is now not automatically flush right.	269
\l@dunhbox@line: Move the call to \inserthangingsymbol to allow use \hfill inside.	131
\pend: Spurious space in \pend.	127
General: Corrects a false hanging verse when a verse is exactly the length of a line.	1
v1.7.0.	
General: New features for managing page breaks.	60
v1.8.0.	
\endquote: Correction of sectioning commands in parallel texts.	297
\get@index@command: Debug \get@index@command and compatibility with hyperref package.	261
\newhookcommand@series@reload: Debug \beforenotesX and \maxhnotesX which did not work.	224
\prevpage@num: Correct \parafootsep when using with ledgroup.	160
General: Compatibility with parledgroup option ofeledpar package.	1
If imakeidx and hyperref are loaded, adds hyperref in the index.	259
v1.8.1.	
General: Debug endnotes when more than one series is used (change the position where tools for endnotes are defined).	201
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, witheledpar.	259
v1.9.0.	
\doxtrafeet: Add \fnpos to choice the order of footnotes.	229
\l@dfeetendmini: Add \mpfnpos to choice the order of footnotes in minipage / ledgroup.	254
v1.10.0.	
\endquote: Correction of sectioning commands in parallel texts.	297
General: Add \pstartref and \xpstartref to refer to a pstart number (extension of \edlabel).	1
v1.10.1.	
General: Compatibility with cleveref.	1
v1.10.2.	
General: Compatibility of stanza with v1.8a of babel-greek.	1
v1.10.3.	
General: Debug of cross-referencing.	1
v1.10.4.	
General: Debug of critical notes in edtabular environment.	1
v1.10.5.	
General: Debug of \pausenumbering.	1
Debug of \xxref.	1
v1.10.6.	
General: Debug of interaction between \autopar and \pausenumbering.	1
v1.11.0.	
General: Add hooks to disable the font selection for lemma in footnote.	37
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.	95
\AtEveryPstart: New optional argument for \pstart, to execute code before it. . .	126
\edindex: Use correctly default index when imakeidx is loaded.	262
\endquote: \ledxxx sectioning commands are deprecated and replaced by \eledxxx commands.	297
\initnumbering@reg: \beginnumbering is defined only on eledmac, not on eledpar. . .	77
\l@dcnote: \l@dlnote, \l@drnote and \l@dcnote defined only one time, in eledmac, including needs for eledpar case.	250
\l@dgetsidenote@margin: \sidenotemargin is now directly defined in eledmac to be able to manage eledpar.	248
\l@dnumpstartsL: Add \ifledRcol@ for eledpar	75
\l@dunhbox@line: \do@line is split in more little commands.	131
\newhookcommand@series@reload: Debug \beforenotesX and \maxhnotesX which did not work when called after \footparagraphX.	224
Debug \Xbeforenotes and \Xmaxhnotes which did not work when called after \footparagraph.	224
\pend: New optional argument for \pend, to execute code after it.	127
\stanza: &can have an optional argument: content to be printed after.	273
\Stanza can have an optional argument: content to be printed before.	273
Add \newverse macro, \falseverse deprecated.	273
General: Add \ledinnernote and \ledouternote commands.	51
Add \Xendparagraph and related settings.	42
Add hyperlink to crossref (needs hyperref package).	47
Compatibility with musixtex.	1
Debug eledmac sectioning command after using \resumenumbering.	1
Ensure that imakeidx is loaded <i>before</i> eledmac	259
New hooks: \Xafterrule and \afterruleX	40
New options for ragged-paragraph notes	40
New sectioning commands.	58
Optional arguments for \pstart and \pend.	17
v1.12.1.	
\wrap@edcrossref: Fix spurious spaces.	239
v1.12.2.	
\l@dunhbox@line: Fix a bug with critical notes at the tops of pages (added by v12.0.0)	131
v1.12.3.	
\flag@end: \flag@start and \flag@end are now defined only one time for eledmac and eledpar	108
\flag@start send a error message when a \edtext is done without insert (note)	108
\reledmac@error: Replaced error messages	69
General: Add macros for new messages since v0.7	69
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.	51
Debug left and right notes (bugs added by 1.12.0)	1
Underline lemma in \eledxxx when using draft mode.	1
v1.12.4.	
General: Debug spurious page breaks before \chapter (bug added in 1.12.0)	1

v1.12.5.	
\@edindex@hyperref: Debug \edindex when hyperref is not loaded	264
\@ssect: Debug \eledchapter in parallel with memoir	301
\doinsidelinehook: Added \dolinehook and \doinsidelinehook	133
\endnumbering: Allow to mix parallel columns and normal text when using	
\pausenumbering	78
\l@dgobblearg: \l@dgobblearg becomes \l@gobbeloptarg	281
\l@drestoreforedtext: Debug optional arguments of \Xfootnote in tabular context	281
\resumenumbering: Debug \resumenumbering	79
v1.12.7.	
\wrap@edcrossref: \wrap@edcrossref is now robust	239
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	108
v1.13.0.	
\newhooktoggle@series@reload: Add \newhookcommand@toggle@reload	223
\para@footsetupX: In \para@footsetupX, use \columnwidth instead of \hsize	192
\settoggle@series: \settoggle@series can take an optional arguments to reload	
series setup.	221
General: Add \Xnoteswidthliketwocolumns and \noteswidthliketwocolumnsX	41
Added widthliketwocolumns option	66
v1.13.1.	
\thepstart: Add \l@dzeropenalties in \pstart	126
General: Coming back of page and line breaking penalties's management, deleted by	
error in v0.17.	1
Debug quotation environment inside of a \pstart preceded by a sectioning command.	1
v1.13.2.	
\l@dnumpstartsL: Add \ifl@dpadding for \eledpar	75
General: Fix bug with normal footnotes, added by v1.13.0.	1
v1.13.3.	
General: Fix extra spaces with paragraphed footnotes, added by v1.13.0.	1
v1.13.4.	
General: Fix bug with index when memoir class is used without hyperref	1
v1.14.0.	
\edindex: Let \eledmac take advantage of \makeidx even when memoir class is used	262
General: Debug spurious characters before endnotes.	201
Delete previous override of \l@d@@wrindexhyp at the beginning of a document	
when hyperref is not loaded.	266
Move gobbling command	74
Provide \@gobblefour	74
v1.14.1.	
\@ssect: Debug sectioning commands when using both <code>handout</code> and <code>hyperref</code>	
package.	304
v1.14.2.	
\@ssect: Debug \edtext after starred sectioning commands when using memoir class.	301
v1.15.0.	
\@edtext@level: New boolean \if@edtext@.	114
\arrangementX@threecol: Correct bug with paragraphed familiar footnotes setting.	191
\endsub: Restore subline feature (disabled by mistake in v1.8.0).	108
\if@lemmacommand@: New boolean \iflemmacommand@.	119

General: Fix bug with footnotes layout when using some options of the geometry package (bug add by v1.13.0).	1
New commands <code>\AtEveryPstart</code> and <code>\AtEveryPend</code>	18
New tools to prevent ambiguous references in lemma	26
v1.15.1.	
<code>\line@list@stuff</code> : Revert modification of 1.5.2 which makes bug with numbering. Leave vertical mode to solve spurious space before minipage.	106
v1.16.0.	
General: <code>\edtext</code> is now defined only in <code>eledmac</code> , not in <code>eledpar</code> . Debug wrong numbering when using <code>\sameword + eledpar + \tag</code> command.	114
Compatibility of standard footnotes with some biblatex styles.	1
New <code>\stanzaindent</code> command.	1
v1.16.1.	
<code>\xlineref</code> : <code>\lineref</code> is not defined if defined by some other package, like <code>lineno</code> . Eledmac provides <code>\edlineref</code> instead.	239
v1.17.0.	
<code>\edtext</code> : Error message when calling <code>\edtext</code> outside of a numbered paragraph. . . .	114
v1.18.0.	
<code>\@edindex@hyperref</code> : Fix spurious space with <code>\edindex</code> when using <code>imakeidx/indextools + hyperref</code>	264
<code>\edlabel</code> : <code>\edlabel</code> is now defined only one time for both <code>eledmac</code> and <code>eledpar</code> . . .	234
<code>\l@d@section</code> : Option <code>parapparatus</code> works for endnotes.	202
<code>\l@dnumpstartsL</code> : Add <code>\ifl@dprintingpages</code> and <code>\@dprintingcolumns</code> for <code>eledpar</code>	75
<code>\print@line</code> : Compatibility with Lua \TeX RTL languages.	131
<code>\printlinefootnote</code> : Code refactoring in <code>\printlinefootnote</code> : the printing of the numbers are factorized in <code>\printlinefootnotearea</code>	168
<code>\printpstart</code> : Debug <code>\Xpstart</code> with parallel pages and columns (<code>eledpar</code>)	167
General: Add <code>\Xpstarteverytime</code>	33
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.	33
v1.19.0.	
<code>\footssplitskips</code> : <code>\footssplitskips</code> doesn't set <code>\floatingpenalty</code> to <code>\@MM</code> when processing parallel pages.	148
<code>\xxref</code> : <code>\xxref</code> works also with right side numbers, when <code>\@Rlineflag</code> is not empty. .	241
General: <code>\Xmaxhnotes</code> and <code>\maxhnotesX</code> work now for both two-columns and three-columns setting.	1
Compatibility with <code>eledpar</code> v1.13.0.	1
v1.19.1.	
General: Call <code>\correct@footinsX@box</code> and <code>\correct@Xfootins@box</code> directly in <code>\print@notesX@forpages</code> and <code>\print@Xnotes@forpages</code> , that is in <code>eledpar</code> . . .	1
v1.20.0.	
<code>\printlines</code> : Added <code>\ifl@d@Xmorethantwolines</code> and <code>\ifl@d@Xmorethantwolines</code> to <code>\printlines</code>	176
<code>\stanza</code> : <code>&</code> and <code>&</code> can be preceded by spaces.	273
<code>\xxref</code> : Debug <code>\xxref</code> when not loading <code>eledpar</code> (fix bug added in 1.19.0).	241
General: Add <code>\Xendboxlinenum</code>	35
Add <code>\Xtwolines</code> and <code>\Xmorethantwolines</code> hooks	32
Add series option.	1

Correct <code>\Xinplaceofnumber</code> hook.	1
Explicit error message when calling <code>\Xfootnote</code> outside of <code>\edtext</code>	1
Fix bug with line number typesetting direction when using <code>\eledsection</code> and similar commands for RTL texts with <code>Lua\TeX</code>	1
Fix issues with RTL text in notes when using <code>Lua\TeX</code>	1
Options fulllines in <code>\Xfootnote</code>	23
The <code>\newifs</code> are not followed by boolean values set to false, because it is the <code>\TeX</code> default setting.	1
v1.21.0.	
<code>\@edindex@hyperref</code> : Look at the hyperindex option of hyperref before inserting hyperref	264
<code>\l@d@section</code> : <code>\endnotes</code> take five arguments.	202
<code>\ledinnotemark</code> : Add <code>\ledinnotemark</code>	262
<code>\ledsetnormalparstuffX</code> : <code>\ledsetnormalparstuff</code> is deprecated and becomes <code>\ledsetnormalparstuffX</code> and <code>\Xledsetnormalparstuff</code>	196
<code>\n@num</code> : <code>\n@num@ref</code> deleted	102
<code>\n@num</code> defined only one time for both <code>Eledmac</code> and <code>Eledpar</code>	102
<code>\newhookcommand@series</code> : <code>\newhookcommand@series</code> can take an optional argument.	223
<code>\newhooktoggle@series</code> : <code>\newhooktoggle@series</code> can take an optional argument.	223
<code>\print@footnoteXrule</code> : Code refactoring: the spaces after the footnote rules are directly managed in <code>\print@Xfootnoterule</code> and <code>\print@footnoteXrule</code>	198
<code>\seriesatend</code> : Fix spurious space in <code>\seriesatend</code>	220
<code>\skipnumbering</code> : <code>\skipnumbering</code> defined only one time for both <code>Eledmac</code> and <code>Eledpar</code>	110
Correct <code>\skipnumbering</code> for stanza.	110
Delete <code>\skipnumbering@reg</code>	110
General: <code>\AtEveryPstart</code> and <code>\AtEveryPend</code> are now compatible with <code>\autopar</code> . . .	1
<code>\Xafterrule</code> and <code>\afterruleX</code> features no longer create problems of overflowing at the bottom of the page.	1
<code>\chapter</code> inside optional argument of <code>\pstart</code> works when typesetting parallel pages	1
<code>\preXnotes</code> and <code>\prenotesX</code> features no longer create problems of overflowing at the bottom of the page.	1
<code>\seriesatbegin</code> and <code>\seriesatbegin</code> more efficient	219
Add <code>\applabel</code> and related	49
Add <code>\beforenotesX</code> and <code>\Xbeforenotes</code> features for notes set in two and three column.	1
Add <code>\hidenumbering</code>	22
Add <code>\Xcolalign</code> and <code>\colalignX</code> hooks	39
Add <code>\Xendtwolines</code> , <code>\Xendmorethantwolines</code> , <code>\Xendtwolinesbutnotmore</code> and <code>\Xendtwolinesonlyinsamepage</code>	33
Add <code>\Xparindent</code> and <code>\hangindentX</code>	38
Add <code>\Xtwolinesbutnotmore</code> and <code>\Xtwolinesonlyinsamepage</code>	1
Add <code>nocritical</code> , <code>noend</code> , <code>nofamiliar</code> and <code>noledgroup</code> options.	1
Add <code>noeledsec</code> package option	1
Debug <code>\beforenotesX</code> <code>\maxhnotesX</code> <code>\noteswidthliketwocolumnsX</code> and <code>\afterruleX</code> with footnotes set in two and three columns.	1
Fix bug when a <code>\Xfootnote</code> follows a <code>\Xendnote</code> in the second argument of <code>\edtext</code> (bug added in <code>eledmac</code> 1.0.0).	1

Fix bug with <code>\maxnotesX</code> when using <code>\foottwocolX</code> or <code>\footthreecolX</code>	1
Fix bug with space between columns with notes in two columns (bug added in v1.13.0).	1
Fix spurious space after first page number in <code>\doendnotes</code> . <code>oldprintnpnumspace</code> option allows to come back to previous setting	1
<code>parapparatus</code> option works now with familiar footnotes.	1
Provide <code>\@gobblefive</code>	74
v1.22.0.	
<code>\ledinnote</code> : <code>\ledinnote</code> takes a first optional argument, which is the label for hyperlinks.	262
General: Add <code>\doendnotesbysection</code> command.	24
Add option for lemma separator inside endnotes	36
Adds hyperlink for references to notes in indices.	1
Fix conflict between <code>noend</code> package option and <code>edtabularx</code> environments	1
Provides support for <code>xindy</code>	1
Standardize endnotes handbook.	24
When using <code>hyperref</code> package, internal links in index or with <code>\edlineref</code> are now targeted to the top and not longer to the bottom of the lines they refer to.	1
v1.22.1.	
<code>\prevpage@num</code> : Correct double symbol when using both <code>\parafootsep</code> and <code>\Xsymlinenum</code>	160
General: Fix bug (added on v1.22.0) with <code>\Xinplaceofnumber</code> hook.	1
v1.23.0.	
<code>\@edtext@level</code> : The boolean <code>\if@edtext@</code> becomes the counter <code>\edtext@level</code>	114
<code>\SErefwithpage</code> : Debug <code>\Xendtwolines</code> , <code>\Xendmorethantwolines</code> , <code>\Xendtwolinesbutnotmore</code> and <code>\Xendtwolinesonlyinsamepage</code> when using <code>\apprefwithpage</code>	244
<code>\lemma</code> : Fix spurious space after <code>\lemma</code> command	118
<code>\newseries@</code> : Prevent spurious spaces when <code>\Afootnote</code> and similar commands are followed by spaces (bug added on 1.0.0).	212
<code>\sameword</code> : In order to allow use of <code>\sameword</code> with <code>inputenc</code> , we detokenize its mandatory argument before using it in control sequence names.	122
General: Add <code>\Xboxlinenumalign</code> and <code>\Xendboxlinenumalign</code>	34
Add <code>\Xboxstartlinenum</code> , <code>\Xendboxstartlinenum</code> , <code>\Xboxendlinenum</code> , <code>\Xendboxendlinenum</code>	35
Allow use of <code>\sameword</code> with <code>inputenc</code> managing of UTF-8.	1
Compatibility between <code>nofamiliar/nocriticals</code> option and <code>minipage/ledgroup</code>	1
Error message when using <code>\beginnumbering... \endnumbering</code> without <code>\pstart</code>	1
Fix bug with <code>\sameword</code> when the lemma overlaps multiple line.	26
Fix bug with <code>\sameword</code> when the same lemma is used for multiple notes or for nested <code>\edtexts</code>	26
Fix bug with <code>\skipnumbering</code> called immediately after a <code>\pstart</code>	1
Fix error of <code>\iftrue</code> not closed.	1
Fix spurious space with <code>\skipnumbering</code> (bug added on v1.21.0).	1
New tools to ensure the line-list file uses the right version of commands when upgrading the <code>eledmac</code> version.	1
Optional argument of <code>\sameword</code> can be a comma-separated list of <code>\edtext</code> depth.	26
v1.23.1.	
General: Fix bug with <code>\lemma</code> command in the right side.	1
v1.23.2.	
General: Compatibility with 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 ‘ <code>nocritical</code> ’ option.	1
v1.24.2.	
General: Fix incompatibility between ‘ <code>nofamiliar</code> ’ option and ‘ <code>memoir</code> ’ package.	1
v1.24.3.	
General: Restore marginal numbers and notes with sectioning command (bug introduced in v1.21.0)	1
v1.24.4.	
General: Fix spurious space with <code>\edindex</code> when using <code>xindy+hyperref</code> option.	1
v1.24.5.	
General: Fix bug of indent, when a added in 1.1.0, when a <code>\beginnumbering</code> immediately follow a sectioning command.	1
v2.0.0.	
<code>\@iiiminipage</code> : Patch <code>\@iiiminipage</code> instead of redefining it.	256
<code>\@xympar</code> : Patching <code>\@xympar</code> instead of redefining it	248
<code>\endminipage</code> : Patch <code>\endminipage</code> instead of redefining it.	256
<code>\initnumbering@quote</code> : <code>\initnumbering@sectcmd</code> becomes <code>\initnumbering@quote</code>	297
<code>\l@advance@parledgroup@beforenormalnotes</code> : Some conde of <code>\l@dumboxmpfoot</code> moved to <code>\l@advance@parledegroupp@beforenormalnotes</code>	256
<code>\newseries@</code> : One endnotes file by series.	217
General: <code>\@makecol</code> , <code>\@reinserts</code> and <code>\@doclearpage</code> are patched instead of begin redefined	232
<code>\doxtrafeeti</code> becomes <code>\do@feetX</code> ; <code>\doxtrafeetii</code> becomes <code>\Xdo@feet</code> ; <code>\@opxtrafeeti</code> becomes <code>\@opfeetX</code> ; <code>\doreinxtrafeetii</code> becomes <code>\X@doreinfeet</code> ; <code>\doreinxtrafeeti</code> becomes <code>\@doreinfeetX</code>	232
Add <code>\Xendinplaceofnumber</code> hook.	1
Add <code>\Xendnonumber</code> hook.	1
Add <code>nonum</code> option for endnotes.	1
Fix bug when printing only one series of endnotes, but wanted to keep endnotes for other series.	1
In order to have a more consistent name’s convention, many names has been changed.	1
Many \TeX ’s output macros are now patched and not override.	1
Package’s name becomes <code>reledmac</code>	1
Patch <code>\@footnotemark</code> instead of redefine it	178
Suppress indexing command specific to <code>memoir</code>	262
v2.0.1.	
General: Fix bug in <code>eledmac-compat</code> option	1
Fix incompatibility between optional argument of <code>\pstart</code> and <code>\numberpstarttrue</code>	1
v2.1.0.	
General: Fix bug with <code>\advanceline</code> at the beginning of a <code>\pstart</code>	1
Fix bug with <code>\chapter</code> in optional argument of <code>\pstart</code> in parallel typesetting with <code>scrbook</code>	1
Fix bug with <code>\eledchapter</code> in parallel typesetting with <code>scrbook</code>	1
Fix bug with <code>\setline</code> at the beginning of a <code>\pstart</code>	1
Fix spacing bug with <code>\Xhooknote</code> and <code>\hooknoteX</code> when using them to insert text and not to execute code.	1

New tools to number stanzas	1
v2.1.1.	
General: Fix bug with <code>\ledpbsetting{before}</code>	1
v2.1.2.	
General: Fix bug with lineation by <code>pstart</code> and <code>tabular</code> environments (added in 2.1.0).	1
v2.1.3.	
<code>\ledsetnormalparstuffX</code> : Replaced <code>\noindent</code> with <code>\parindent</code> set to 0pt.	196
General: <code>\Xhangindent</code> and <code>\hangindentX</code> work now with all the paragraphs in the note.	1
<code>\Xnoindent</code> and <code>\noindentX</code> work now again (broken in 2.0.0).	1
Change some internal code in order to provide compatibility with \TeX release of october 2015	1
Fix bug which inserted double space before paragraphed familiar notes.	1
Fix bug with <code>\edindex</code> when using not-Latin characters without UTF-8 engines	1
v2.2.0.	
General: Fix bug with combination of <code>\onehalfspacing</code> and two columns and three columns notes typeset.	1
Fix bug with some setting command and optimization option.	1
Fix spurious space with paragraphed critical notes when using $\text{Lua}\TeX$	1
Increase line list version number to ensure compatibility with new options of <code>reledpar</code> package.	1
New setting tools for endnotes: <code>\Xendnumberonlyfirstinline</code> , <code>\Xendnumberonlyfirstintwolines</code> , <code>\Xendsymmlinenumber</code> , <code>\Xendbeforenumber</code> , <code>\Xendafternumber</code> , <code>\Xendbeforemsymmlinenumber</code> , <code>\Xendaftersymmlinenumber</code> , <code>\Xendboxsymmlinenumber</code> , <code>\Xendhangindent</code> , <code>\Xendbhooklinenumber</code> , <code>\Xendahooklinenumber</code> , <code>\Xendbhookinplaceofnumber</code> , <code>\Xendahookinplaceofnumber</code>	1
v2.2.1.	
General: Compatibility with TeXformat 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.	
<code>\footnoteoptions@</code> : First argument of <code>\footnoteoption@</code> is now mandatory, not optional.	146
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>	32

Option <code>linrangesep</code> for critical footnotes and endnotes.	32
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 \LaTeX <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 \LaTeX <code>\roman</code> command, like Greek.	1
v2.5.0.	
<code>\SErefwithpage</code> : Debug <code>\setapprefprefixsingle</code>	243
<code>\edlabel</code> : Fix bug when calling <code>\edlabel</code> in a footnotes of the rightside	234
<code>\l@d@section</code> : <code>\endnotes</code> take six arguments.	202
<code>\printlines</code> : <code>\printlines</code> takes an eighth argument: the line flag	175
<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.	239
General: <code>\apprefwithpage</code> and <code>\appref</code> print double quotation mark when the label was not defined.	1
<code>\apprefwithpage</code> and <code>\appref</code> work with right side crossref.	1
<code>\apprefwithpage</code> works also when <code>noend</code> option is enabled.	1
<code>\appref</code> and <code>\apprefwithpage</code> can take <code>linrangesep</code> optional argument.	1
<code>\edlabel</code> works now in <code>Xfootnote</code>	1
<code>\lemma</code> can be used even when the <code>nocritical</code> is enabled.	1
Compatibility with new hook and tools of <code>reledpar</code> 2.6.0.	1
Fix spurious vertical space in <code>astanza</code> environment (<code>reledpar</code>)	1
Log now states ‘There were undefined references’ when using wrong references in <code>\edlineref</code> or <code>edpageref</code>	1
New hooks to customize page and line number appearance in endnotes.	1
New hooks: <code>\Xhookgroup</code> and <code>\bhookgroupX</code>	1
New tools to easily make cross-reference to a passage defined by a start and an end line	49
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>	161
<code>\l@d@section</code> : <code>\endnotes</code> take seven arguments.	202
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>innnote</code> and <code>notenumber</code> options of <code>indextools</code> package, when indexing outside of a <code>ledgroup</code>	1
New commands to make glossaries connected to page and linenumber with the <code>glossaries</code> package	1
New hooks: <code>\Xhsize</code> and <code>\hsizeX</code>	41
New hooks: <code>\Xlemmafont</code> and <code>\Xendlemmafont</code>	37

New setting commands: <code>\setSErefonlypageprefixsingle</code> and <code>\setSErefonlypageprefixmore</code>	1
Warning for duplicate and undefined labels are parsable by latexmk	1
Warning for duplicate labels does not send any more a false line and page number	1
When using <code>hyperref</code> package, add link in familiar footnotes between the footnote marks in the text and the footnote marks in the footnote	1
When using <code>hyperref</code> package, add links for <code>\SEref</code> and <code>related</code> , <code>\appref</code> and <code>related</code> .	1
When using <code>hyperref</code> package, add links from critical footnotes and critical endnotes to the line of text they refers	1
v2.7.1.	
General: Debug <code>\Xhookgroup</code> hooks executed on columnar footnotes (moved to a larger group, to take effect).	1
v2.7.2.	
General: <code>\Xhsize</code> and <code>\hsizeX</code> become <code>\Xwidth</code> and <code>\widthX</code>	41
Fix problem of hyphenation when using <code>hyperref</code> package (added in v2.7.0).	1
v2.8.0.	
<code>\l@d@section</code> : <code>\Xendhangindent</code> and <code>\Xendafternote</code> can take values which are relative to the font size of the endnote.	202
General: <code>reledmac</code> cross-referencing can take advantage of <code>xr</code> package.	1
More <code>\edgls...</code> commands.	1
No indentation for paragraphed notes in <code>ledgroup</code> . Can be changed with <code>\Xparindent</code> and <code>\parindentX</code> .	1
v2.8.1.	
General: Warnings for undefined labels are really parsable by latexmk	1
v2.8.2.	
General: Fix bug concerning indent in a paragraph immediately following a sectioning command (bug NOT fixed on <code>reledpar</code>)	1
Fix bug with <code>\AtEveryPstart</code> added in version 2.0.0.	1
Fix bug with vertical space after the between-sectioning command as optional argument of a <code>\pstart</code> and <code>\pstart</code> content	1
v2.9.0.	
General: Allow continuing line numbering between normal text and parallel text, using <code>\pausenumbering</code> and <code>\resumenumbering</code> and the <code>continuousnumberingwithcolumns</code> option.	1
Fix bug when using <code>\linenum{page}</code> and <code>\pausenumbering... \resumenumbering</code> .	1
Fix bug with three- and two-column footnote setting (added in v2.4.0).	1
Fix spurious space inside three-column familiar footnote.	1
Write correct metadata in numbered files when using <code>\pausenumbering... \resumenumbering</code> .	1
v2.9.1.	
General: Fix bug when notes start with “plus” or “minus”.	1
v2.9.2.	
General: Fix bug with <code>hyperref</code> package when a lemma starts with “plus” or “minus” (bug introduced in v. 2.7.0).	1
v2.9.3.	
General: Fix bug with line number position and reset added by v. 2.9.0	1
v2.10.0.	
<code>\print@lemma</code> : Code refactoring between <code>\parafootfmt</code> , <code>\twocolfootfmt</code> , <code>\threecolfootfmt</code> and <code>\normalfootfmt</code> .	168

General: Add <code>\AtEveryStanza</code> and <code>\AtEveryStopStanza</code>	1
Fix bug with <code>\ledlsnotefontsetup</code> and <code>\ledrsnotefontsetup</code> when using <code>\color</code> command inside.	1
More accurate error messages.	1
New hooks: <code>\Xwrapcontent</code> , <code>\Xendwrapcontent</code> and <code>\wrapcontentX</code>	38
New hooks: <code>\Xwraplemma</code> and <code>\Xendwraplemma</code>	37